University of Reading

Investigating the impact of image content on the energy efficiency of hardware-accelerated digital spatial filters

Article

Accepted Version

It is advisable to refer to the publisher's version if you intend to cite from the work. See Guidance on citing.

To link to this article DOI: http://dx.doi.org/10.1145/3341819

Publisher: Association for Computing Machinery

CentAUR

Central Archive at the University of Reading

Reading's research outputs online

Investigating the Impact of Image Content On the Energy Efficiency of Hardware Accelerated Digital Spatial Filters

Rajkumar K. Raval, University of Reading, *UK*
Atta Badii, University of Reading, *UK*

Abstract

Battery operated low-power portable computing devices are becoming an inseparable part of human daily life. One of the major goals is to achieve the longest battery life in such a device. Additionally, the need for performance in processing multimedia content is ever increasing. Processing image and video content consume more power than other applications. A widely used approach to improving energy efficiency is to implement the computationally intensive functions as digital hardware accelerators. Spatial filtering is one of the most commonly used methods of digital image processing. As per the Fourier theory, an image can be considered as a two-dimensional signal that is composed of spatially extended two-dimensional sinusoidal patterns called gratings. Spatial frequency theory states that sinusoidal gratings can be characterised by its spatial frequency, phase, amplitude and orientation. This paper presents results from our investigation into assessing the impact of these characteristics of a digital image on the energy efficiency of hardware accelerated spatial filters employed to process the same image. Two greyscale images each of size 128x128 pixels comprising of two-dimensional sinusoidal gratings at maximum spatial frequency of 64 cycles per image orientated at 0 and 90 degrees respectively, were processed in a hardware implemented Gaussian smoothing filter. The energy efficiency of the filter was compared with the baseline energy efficiency of processing a featureless plain black image. The results show that energy efficiency of the filter drops to 12.5% when the gratings are orientated at 0 degrees whilst rises to 72.38% at 90 degrees.

Introduction and Motivation

For the past thirty years, Moore's law together with Dennard scaling have driven the era of modern computing providing exponential increases in performance. Moore's law [25] states that the number of transistors in an integrated circuit doubles every two years approximately whereas Dennard scaling [8] claims that even though transistors get smaller, their power density remains constant. Another related law, Koomey's law [20] states that performance per watt would double every 1.57 years. However, the scale of integrated circuits density achievable has exceeded the

XX XX

levels within which Dennard scaling and Koomey's law were applicable, and the computational capabilities of multi-cores are still rising, but with much less enhancement in energy efficiency. The International Technology Roadmap for Semiconductors (ITRS) reported that following Moore's law, the transistor density continues to double every two years however the energy efficiency of transistors is increasing only by 1.4x. This short fall in the energy efficiency indicates the end of the Dennardian scaling era where progress was measured with improvements in transistor count and speed, and the beginning of a new era where advances are measured by improvements in transistor energy efficiency [2]. All of this has resulted in another technological constraint known as the utilisation wall which limits the portion of the chip that can be used at the full performance within the power budget at the same time [11]. This limits the number of transistors that can be active at a given time due to the power constraint. Therefore, some parts of the chip i.e. transistors have to remain inactive or underperforming to allow the chip to function within the power budget. This presents the current major technological issue of dark silicon.

It is important to mention the three key prevailing technological bottlenecks for high performance computational efficiency gains. These are the memory bottleneck, the Instructional Level Parallelism (ILP) bottleneck and the power bottleneck. The memory bottleneck relates to the recognised technological constraint that memory speed does not increase as fast as computing speed and as a result it is difficult to hide memory latency. ILP quantifies the number of instructions that can be executed in a single clock cycle. However, Amdahl's law [1] states that the maximum speedup of a program is limited by the serial portion of the code. This presents the ILP bottleneck. The utilisation wall and dark silicon together present the power bottleneck. Therefore, it becomes necessary to explore all the avenues of reducing power consumption and improving energy efficiency of such a digital system. Energy efficiency optimisation has become an essential objective in the design of modern embedded systems. The main motivation of this paper is to address the third bottleneck, the power bottleneck.

Portable mobile devices such as tablets, mobile phones, IoT devices, wearable computing devices etc. to list few are becoming part of daily human life [23]. Many such devices with a screen or a camera include some form of digital image processing circuit. These devices mainly run on a battery and therefore battery life-time is a critical factor for their continued functioning. It has now been established that multimedia applications that involve processing image and video content, dominate the power consumption in any battery-operated computing device.

Digital images are essentially a collection of pixels. These pixels are samples of intensity values which are represented in the form of binary numbers. The variation in the content of digital images can be considered to be the variation in the values of the constituent pixels and vice versa. These pixel values are typically represented using binary numbers comprising of 1s and 0s. When the image is processed in a hardware accelerated image processing block, which is fundamentally a digital logic circuit, these binary numbers directly contribute to the switching of the digital logic circuit. It is now well known that the amount of switching is one of the major contributing factors in the dynamic power consumption of digital logic circuit. Therefore, the binary pixel values must have some direct impact on the power consumption and thus the energy efficiency of the circuit.

Let us examine the structure of an image closely by taking an example of a greyscale image of size 128x128 pixels as shown in Figure 1. A greyscale image is comprised of pixels and these pixel values range from 0 to 255 if the pixel width is, typically, 8 bits. Let the shade of the image be just plain grey i.e. all the pixel values have one value. Let us for the sake of this example, take the value of the pixel, i.e. shade, to be 170 in decimal value. If this number is represented in hexadecimal, it is 0xAA and in binary it is 101010101. Table 1 shows a generic binary representation of the grey image.

Fig. 1. Grey Image with all the pixels of value 0xAA

Table 1. Greyscale image having all pixels with value 0xAA

Pixel Index	0	1	…	127
0	10101010	10101010	…	10101010
1	10101010	10101010	…	10101010
…	…	…	…	…
127	10101010	10101010	…	10101010

Please note the pixel values, moving from one pixel to the next and within the pixel one bit to the next bit, the bit value transitions from 1 to 0 and 0 to 1 are purely due to the binary value representation of the pixel data. On a screen, this image would appear to feature less and content less to human eyes as shown in Figure 1. This means even if the visual content in an image is not changing spatially, just because of the way pixel values are represented there exists switching due to the binary values of the pixels. Therefore, if the content changes, the characteristics of the switching will change even more. In a video of a newsreader delivering a news bulletin, one can say that the content is nearly static and changing very slowly in comparison with a sports video [22]. However, as explained, there is always inherent continuous switching activity due to the way pixel values are represented in binary number system and this cannot be avoided.

This warranted some initial empirical evidence to motivate us to further carry out a detailed investigation. Therefore, a two-dimensional Gaussian filter with a 5x5 kernel with an input clock frequency of 500 MHz was implemented for Xilinx Virtex-6 [3] Field Programmable Gate Array (FPGA) using the Xilinx System Generator [38], [24] Electronic Design Automation (EDA) tool. Images of Lena, a chequerboard, a plain white and a plain black of size 640x480 pixels were processed in the filter. Dynamic power consumption to process each of the images was estimated using the simulation based power estimation design flow in the Xilinx System Generator. The results as detailed in Table 2 intrigued us and pointed to a new more focused direction into investigating the impact of image content on the energy efficiency of the filter.

Table 2. The Initial Result

Image Content	Power, W	Dynamic Power, W	Static Power, W	Dynamic Energy consumed per image (mJ)
Black	3.04	0.11	2.93	0.76
White	3.04	0.11	2.93	0.76
Chequers	3.06	0.13	2.93	0.86
Lena	3.07	0.14	2.93	0.98

The research question that begs an answer is, does the content in an arbitrary image have an impact on the energy efficiency of a hardware accelerated image processing function employed to filter the same image? If so, then what is the impact, how can it be quantified? In order answer this question objectively, other related questions are required to be answered first. This paper attempts to answer the following subordinate research questions using supporting evidence provided in the existing literature:

- *How can the content of an image be quantified such that the relationship between the content and energy efficiency can be investigated?*
- *What operations can be commonly performed to process a digital image?*
- *Why is there a need to accelerate digital image processing operations in hardware (digital integrated circuits)?*
- *How can the power consumption of a digital integrated circuit be calculated?*
- *Can the content processed by a digital integrated circuit have an impact on the power consumption of the circuit?*
- *How are the commonly used image processing functions implemented in hardware?*

Background and Literature Review

The research in this paper combines research from several fields. This meant that, against the general expectations of this section, most of the available literature was not suitable for critically reviewing against/for the innovative research work presented in this paper. Nonetheless, the review presented in this chapter is research questions driven and seeks to provide background information and support for the main and subordinate research questions, deductions, experimental design, results and findings presented in this paper, as follows:

How can the content of an image be quantified such that the relationship between the content and energy efficiency can be investigated?

Spatial frequency theory [27] defines an image as an accumulation of many primitive spatial "atoms" whereby these primitives are spatially extended patterns called sinusoidal gratings. Sinusoidal gratings are two-dimensional patterns whose luminance varies according to the sine wave over one spatial dimension and is constant over the perpendicular dimension. The primitive sinusoidal gratings can be characterised using four parameters: spatial frequency, phase, amplitude and orientation. Applying the Fourier analysis method to a two-dimensional image, produces a sum of a set of sinusoidal gratings that vary in spatial frequency, phase, amplitude and orientation. The summation of all of these gratings at the proper amplitudes and phases would produce the original image. Fourier analysis can be used to decompose complex images into primitive components [9,21,27,29,37].

What operations can be commonly performed to process a digital image?

Digital image processing operations are typically classified into three categories: Point based Operations, Local Neighbourhood Operations and Global Operations. The Local Neighbourhood Operations exploit and work on the spatial characteristics around a pixel therefore these types of

operations are also called spatial filters. The focus of this research was on the image processing operations that work on the spatial characteristics of an image. In a spatial filter, the input is typically convolved with a filter mask or a kernel to generate the output as shown in Figure 2. The kernel contains weights or coefficients for producing the desired filter response. Spatial filters are widely used in image processing and pre-processing stages of image processing pipelines.
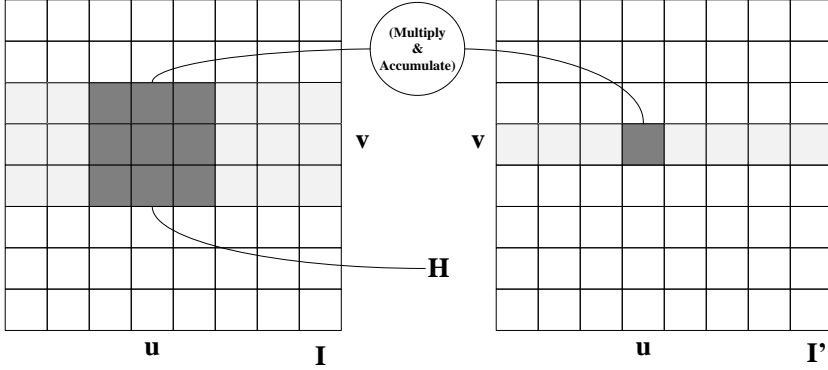

Fig. 2. Sliding window based spatial filtering of an image using Convolution [42]

The Convolution of image I by a kernel H is given by:

$$I'(u,v) = \sum_{u=-k}^{k} \sum_{v=-k}^{k} I(u-i, v-j) \cdot H(i,j) \tag{1}$$

This is denoted by: $I' = I * H$ (2)

Here H is the impulse response function. This is because the kernel function, H, convolved with an impulse signal, $\delta(i,j)$ (an image that is 0 everywhere except at the origin) reproduces itself, H * δ = H.

Why is there a need to accelerate digital image processing operations in hardware (digital integrated circuits)?

There is a significant rising trend in low power and ultra-low power battery operated portable mobile computing devices. Some such devices include mobile phones, tablet computers, Wireless Sensor Network (WSN) Nodes, Internet of Things (IoT) sensors, e-health systems, security systems, home automation and environmental monitoring systems etc. Mobile devices run on a battery and are therefore extremely constrained by battery-imposed energy budget. The density of lithium-ion batteries has shown improvement of only 10% a year therefore, battery technology has not scaled responsive to Moore's law due to a fundamental physics limitation [17].

Computer vision and image processing applications are becoming popular in mobile battery powered devices ranging from every-day smart phones to Unmanned Aerial Vehicles (UAVs) [36]. These algorithms and applications were originally designed for high-performance desktop computers however are now required to be deployed onto much less powerful and energy efficient mobile computing platforms. Designers are expected to increase throughput per Watt in order to meet the performance and energy efficiency requirements. For example, a typical digital camera capturing VGA resolution (640X480) video at a rate of 30 frames requires processing of 27 million pixels per second [18]. This is due to real-time computing requirements and limited data transfer capabilities. This imposes an implied requirement to carry out image processing required by these applications on the edge i.e. locally in the computing device.

Performance is becoming a major issue as the traditional single and multi-core scaling techniques employed in the design of mobile CPU are failing to keep up with the demands of the mobile technology [17]. The single-core thermal design point (TDP) of the mobile CPU's has

saturated at around 1.5W which is similar to the 100W power ceiling common to desktop CPUs. Moreover, the energy efficiency improvement of mobile CPU has plateaued as the performance improvements do not make up for the additional power consumption. Additionally, dark silicon is becoming a major problem due to increasing transistor densities and TDP. Customised hardware accelerators appear to be the way forward in terms of sustaining power, performance and energy improvements for future computing. Modern mobile SoCs comprise of a number of custom hardware accelerators and this number will continue to rise in future. There is 3.5 times rise in fixed-function accelerators across the six most recent Apple SoCs. The ITRS predicts thousands of different on-chip accelerators by 2022. In order to increase performance and reduce energy costs, application specific processors should be used to exploit the structure of algorithms [5].

At the sub-symbolic level, the mathematical operations involved in processing images (i.e. convolution operations consisting of Multiplication and Addition, MAC) need to be repeated on the image data many times. Accordingly, it remains difficult to achieve real-time performance in software-based implementations of image processing while maintaining constraints on the energy consumption of battery operated mobile devices. It can be seen that these types of processing applications could certainly benefit from hardware enabled parallelisation. In this research, the FPGAs have been chosen as a hardware platform to deploy and perform the experiments however the approach can be generalised to be applied to other hardware acceleration platforms such as ASIC.

How can the power consumption of a digital integrated circuit be calculated?

Most modern silicon chips are manufactured using Complementary Metal Oxide Semiconductor (CMOS) technology. The main advantage of CMOS is its low power consumption. The power consumption in a CMOS circuit can be given by the following equation:

$$P_{Total} = P_D + P_S \tag{3}$$

Where P_{Total} is the total power consumption, P_D is the dynamic component and P_S is the static component of the power consumption.

Dynamic power consumption P_D of a CMOS Integrated Circuit (IC) has two extra components namely the switching P_{SW} and the short-circuit power consumption P_{SC}. A typical example of current flowing through a CMOS NOT gate (Inverter) when its output is switching from 0 to 1 and from 1 to 0 is shown in Figure 3. The $I_{0->1}$ is absorbed into the output capacitance CL during the output transition from 0->1 and the current $I_{1->0}$ flows from the output capacitance to the ground during the output transition from 1->0 for discharging the output capacitance. Dynamic power consumption contributes to the overall power consumption significantly when the circuit is switching at high frequency due to charging and discharging of a capacitive output load [34].
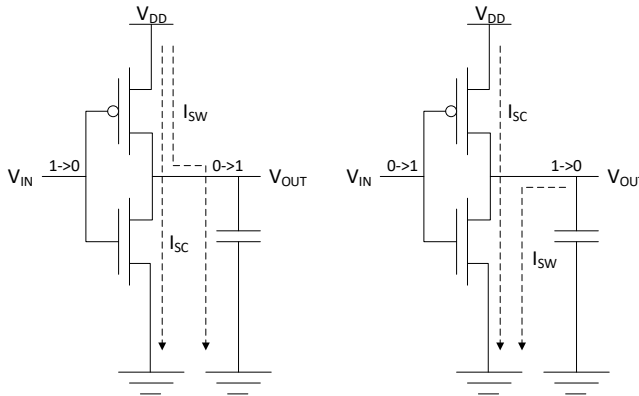

Fig. 3. Dynamic currents flowing through a CMOS inverter when output switches from 0 to 1 (left) and 1 to 0 (right) [34]

$$P_D = P_{SW} + P_{SC} \tag{4}$$

$$P_{SW} = \alpha_t C_L V_{dd}^2 f_{clk} \qquad\qquad\qquad\qquad\qquad\qquad\qquad (5)$$

$$P_{SC} = \alpha_t V_{DD} I_{CC.max} t_{SC} f \qquad\qquad\qquad\qquad\qquad\qquad (6)$$

Here, in Equation 4, P_D is the total dynamic power which is the total of P_{SW} the power consumption due to switching of the transistor and P_{SC} which is due to the momentary short circuit between V_{DD} and ground. This occurs when one transistor is turning ON while the other is switching OFF, at the time there exist a direct momentary path between V_{DD} and ground [31]. Equation 5 shows the switching component of the power consumption where C_L is the load capacitance, f_{clk} is the clock frequency, V_{dd} is the input voltage and α_t is the node switching activity factor. Equation 6 provides a simplified formula that models the average short circuit power for a CMOS gate [4,31]. In equation 6, $I_{CC.max}$ is the peak current which depends on the saturation current of the devices therefore on the transistors dimensions, t_{SC} is related to the signal rise time and fall time [31]. Dynamic power can be reduced significantly using techniques addressing the voltage and frequency parameters of Equation 5 by the way of down-scaling the supply voltage and frequency as and when required [26]. However, in many situations scaling clock frequency or voltage while changing relative speed of the components of the design in order to support the scaling can cause system malfunctions. For example, the conventional architectures based on time-multiplexing in DSP circuits and microprocessors do not allow down-scaling of voltage [26]. In such cases alternative solutions must be explored. One such method is to reduce the effective capacitance of the digital design. The effective capacitance C_{Eff} is defined as the product of the average switching activity (α_t, the average number of transitions per clock cycle) and the total circuit capacitive load.

Can the content processed by a digital integrated circuit have an impact on the power consumption of the circuit?

Much of the research on estimating and optimizing power consumption of embedded systems does not take into account the α_t, as shown in equations 5 and 6, the node switching activity factor as a potential candidate for power optimization. This can be because α_t depends on input data and in any embedded signal processing system, generally, input data is not known a-priori. However, in the case of digital image processing, the input data is the input image and when the images and videos are processed offline the input data is known a priori. Even in the case of a surveillance camera, when it is capturing live images of a scene, the image of the background and foreground remains static if there is no activity. This enables the image content to be known a-priori. The knowledge of the data allows accurate estimation of the resultant switching activity within a hardware processing pipeline and as a result enables accurate estimation of power consumption. In a Power Analysis Attacks (PAAs) scenario, the secret key (data) of a cryptographic core can be retrieved by measuring CMOS power consumption [4].

If the digitally stored data can be identified reliably by measuring power consumption, the converse must also be possible where power consumption can be accurately estimated from the data particularly in the case of digital image processing. It was demonstrated that by analysing consumers' household's electricity usage profile at a higher sample rate [13] it was possible to identify which channel the TV set in the household was displaying. If content could be detected from power consumption, surely, power consumption could be estimated from the content. This motivated us further to carry out detailed investigation in the area of our research.

Five algorithms (1) motion estimation, (2) Discrete Cosine Transform (DCT) (3) Three-dimensional graphics rendering (4) Lempel-Ziv lossless compression (5) Viterbi decoding were examined to be adapted dynamically based on variations in the input signal statistics with a view to reducing power consumption and improve performance [19]. The authors of [6] provided power-performance trade-offs for a dynamically parameterised MPEG-4 motion estimation algorithm. They reported that selecting the correct parameters based on the operating environment reduced the average power consumption by 40% for 2% loss in compression. A data driven clock gating technique to switch off portions of their low-power and low-complexity VLSI architecture

implementation of two-dimensional Discrete Cosine Transform DCT and Inverse Discrete Cosine Transform (IDCT) was presented by [12]. The system monitored input data for being zero, Null Row Check (NRC) and containing sign extended most significant bits, Sign Extension Check (SEC) in order to turn off portions of the implemented circuit. The authors have stated that for typical H.263/MPEG video coding applications their approach provided 36% and 26% power reduction in IDCT and DCT modes respectively. All of this research work does not explain the relationship between image content characterised by spatial frequencies and power and energy consumption of the implemented circuit. Moreover, they do not take into account the inherent switching present in an image or a frame of a video due to the way pixel values are represented in binary values.

Hadizadeh et al [15] proposed a method for producing energy-efficient images for energy-adaptive displays such as OLED displays while preserving its perceptual quality to their original images. The authors exploited the property of OLED displays whereby the energy consumption of pixels is directly proportional to the luminance of the pixels. The authors used a Just-Noticeable-Difference (JND) threshold to reduce the luminance of the pixels in an image. The authors were able to empirically demonstrate that their proposed method reduced energy consumption by about 14.1% while preserving the perceptual quality of the displayed images. This research clearly demonstrates that image content can have an image on the energy efficiency of the hardware used to display the image and serves as a supporting evidence to the findings presented in this paper.

How are the commonly used image processing functions implemented in hardware?

Spatial Image filtering is carried out by performing convolution between a two-dimensional kernel and the image. The algorithms in image processing work in a very similar manner to a two-dimensional convolution operation of an image. The process calculating the output pixel in a convolution involves a rectangular window of the input image pixels and a few constant coefficients fetched typically in a row-major order. This window is then slid and traversed on the whole input image to produce the pixels values for the output image. Therefore, convolution is also known to be working in a sliding window manner. This sliding window is also called a stencil [5]. As shown in Figure 4, the hardware implementation of convolution kernel contains a window function, a line buffer and a stencil register [5]. The window function accepts the pixel values supplied by the stencil register and processes each of the values with the corresponding coefficient values and calculates the output pixel. The line buffer stores the rows of pixel values that are required to be re-used between successive row traversals. The stencil register is provided with a refreshed column of pixel values for each overlapping window of input image.
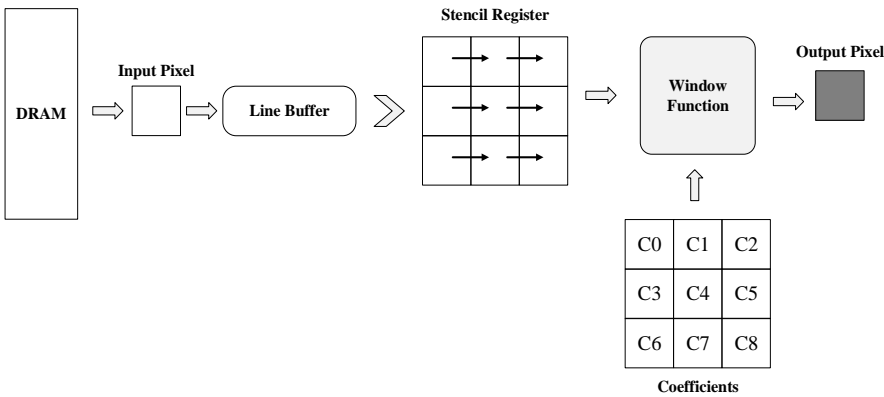

Fig. 4. Stencil kernel architecture for Convolution [35]

As shown in Figure 5, more complex image processing operations can be implemented by cascading the kernels. These kernels work in the same way as a convolution kernel. Therefore, such

convolution family applications can be implemented by reusing hardware components from a single kernel application and interconnecting them [5].
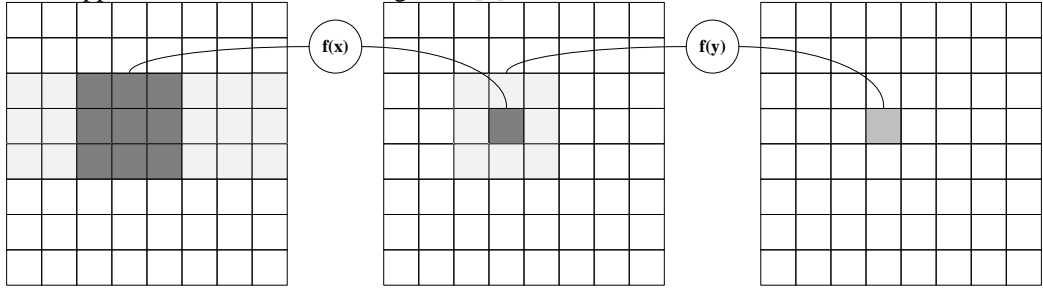

Fig. 5. Cascading kernels in image processing applications [35]

Most image processing applications can be constructed from a set of "stencil" kernels [5]. Many applications in the domains of computer vision, image and signal processing and computational photography could be mapped onto a virtual machine model of the stencil kernel. Stencil kernels typically involve computing the pixel in an output image from a fixed-size sliding window of pixels in its corresponding input image. Stencil kernels are essentially spatial filters which mainly use a convolution operator for processing [5].

The sliding window technique is one of the most widely used techniques in image processing algorithms [10]. Hardware implementation typically comprises of image rows buffered on the chip to benefit from the locality of the data and avoid unnecessary off-chip pixel transfers. The Sliding Window Operations (SWOs) are typically deployed on FPGA based prototyping boards as hardware accelerators for image processing applications [16].

The authors of [14] proposed a configurable image convolution architecture where the input pixel resolution, the image size, the convolution window size, coefficients, and the type of memory used can be explored to identify design trade-offs in obtaining energy efficiency. The authors carried out design space exploration with these parameters and constructed an energy model to estimate the energy consumption. The authors used a number of operations per Joule as their metric for energy efficiency. The authors claimed to have achieved energy efficiency of up to 32.98 Gops/Joule and sustained peak energy efficiency up to 34.38%. Even though the authors carried out the design space exploration with energy efficiency as their main objective, they did not take into account the impact of the switching variability in the input data on the energy consumption.

Experimental Design

The following sections detail the experimental design.

Dependent Variables

Energy efficiency was selected as the main dependent variable however power and energy consumption were the other related the dependent variables of interest.

Independent Variables

Since the impact of the content of an image was investigated, the parameters that characterise the image content at the fundamental level were selected as the independent variables. The aim was to capture a statistically significant sample from the population while considering practicalities of implementation and simulation time. The selected independent variables are shown in Table 3.

Table 3. Independent Variables

Independent Variable	Value
Spatial Frequency	Spatial frequency of sinusoidal gratings present in a synthetic dataset of images of Sinusoidal Gratings. The spatial frequency of gratings ranged from 0 to the maximum number of gratings that can be accommodated in an image of a given size based on the Nyquist-Shannon theorem.
Orientation	Orientation of sinusoidal gratings present in the dataset of images of sinusoidal gratings. The orientation of gratings ranged from 0 to 180 degrees. This is because gratings rotates around the centre and covers the remaining 180 degrees thus covering the entire 360 degrees.
Phase	Images of Sinusoidal Gratings where the phase of gratings ranged from 0 to 360 degrees.
Contrast	Images of Sinusoidal Gratings where the contrast of images was varied as given by the Michelson Contrast. The maximum value was 1.
Image Size	From the literature, it was found that typically, square images were used in image processing research and their dimensions range from 16x16 pixels to 1024x1024 pixels.
Spatial Filter Operation	Image processing operations that are sliding window with a two-dimensional kernel based spatial filter architecture were selected for the experimentation. These operations were implemented using Xilinx System Generator. A library of such operations was created.

Prototyping Platform

The Xilinx ISE and System Generator tool version 14.7 [38] with Matlab-Simulink with the image processing tool box version 2012a has been used to implement and prototype the entire library of spatial filters for FPGA implementation. The System Generator extends the Matlab-Simulink environment to enable hardware design, providing high-level abstractions that can be automatically compiled into an FPGA. The System Generator also carries out full timing simulation based power estimation using the Xilinx Power Estimation Tool XPower Analyser (XPA) [40]. The particular design flow offered by the System Generator which is known as the Timing and Power Analysis flow is used in the experiments. The output at the end of this flow shows both timing and power analysis. This tool takes into account the exact logic and routing resources used and the actual activity from design simulation. All of the implemented spatial filters were also functionally validated on the Xilinx ML605 [39] by prototyping them in the HW/FPGA.

Library of HW Implemented Spatial Filters

In order to explore the impact of energy consumption on various spatial filter operations, a library of hardware implemented spatial filter operations was developed. These filters included line buffers, Difference of Gaussian (DoG) Operation, SIFT Detector, Gaussian 3x3, Gaussian 5x5,

Gaussian 7x7, Gaussian 9x9, Gaussian Separable 5x5, Laplacian 3x3, Mean Filter 5x5, Median
Filter 3x3, Morphological Filter 5x5 and Sobel Filter 3x3. These spatial filters were implemented
based on the most commonly used hardware architecture, the two-dimensional kernel sliding
window architecture. The input image and the kernel coefficients were not stored in memories in
order to isolate the energy efficiency of the implemented FPGA logic. The image was streamed into
and provided by the Matlab-Simulink environment to the hardware block, whereas the kernel
coefficients were hardcoded into the logic. The convolution block required multiplying the input
sample with a coefficient and then adding the result with a result from the next pixel. Therefore,
the implementation required a number of multiply and accumulate blocks consisting of a multiplier
and an adder blocks. Since a Matlab-Simulink based Xilinx System Generator tool is used for the
design entry, each of the implemented spatial filters was saved and stored as a Simulink Model file
with an "mdl" file extension. Gaussian Smoothing spatial filter with a kernel size of 5x5 was
selected as the template spatial filter on which most experiments were carried out.

Software

The following software programs were implemented to generate, extract and process the
necessary input data for the experimentation.

- Synthetic Image Data Set Generator Tool: A program that synthesised images with
  Sinusoidal Gratings while varying spatial frequency, orientation and image size upon user
  configuration.
- Spatial Filter Configuration and model creator Tool: A program that automatically
  configured the existing hardware implemented spatial filter to adapt and support varying
  image sizes and clock frequencies
- Extraction & Tabulation Tool: A program to automatically extract necessary information
  from the timing and power reports generated by the EDA tools and tabulate it in a CSV
  format.
- Co-ordinating Tool: A program coordinating the entire experiment automatically.

Generating Synthetic Images With Sinusoidal Patterns

A dataset of synthetic images was generated using a Matlab script. A black circular mask was
applied to every image with sinusoidal grating. This ensured that the length of the gratings remained
uniform across all the different orientations as shown in Figure 6. The images that were generated
had the Michelson contrast set to 1 which meant the range of black and white pixels of most of the
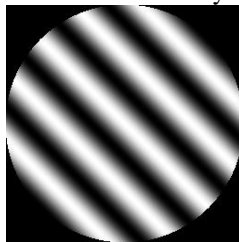gratings is 256 with equal width of black and white half cycles, i.e. from 0x00 to 0xFF.

Fig. 6. Output from the second attempt Matlab Script

Results

In this section, the results from the experimental exploration experiments are presented. A software called JMP [32,33] was used extensively for plotting graphs and data analysis.

Metrics

In the experimentation, at first, power consumption given by watt (W) was used for initial validation. However, in the final validation, energy efficiency of the spatial filter was investigated. The image processing workload can be characterised by the image size and the kernel size. The workload in terms of image and kernel size was kept constant in the experiments where the image content was varied. Energy efficiency was considered to be the number of operations per unit dynamic energy consumed. For an image processing operation such as convolution, where an image size is NxN and kernel size KxK, the energy efficiency can be given by N^2 K^2 divided by dynamic energy consumed by the spatial filter [14]. The metric Giga Operations per Joule metric was used for the energy efficiency analysis. Metric for Spatial Frequency for an image comprising only of two-dimensional sinusoidal gratings was number of cycles per image. The maximum number of sinusoidal gratings that can be fit in an image is half of the width of image given in pixels. The orientation of the sinusoidal gratings in an image was measured in degrees.

Experimental Assumptions

It has been established that a spatial filter follows a common anatomical structure in its hardware implementation. Therefore, the default template architecture for all the experiments was the 5x5 Gaussian Filter implemented in the System Generator. The default clock frequency of the experimentation was set to 100MHz and image size was set to 128x128 pixels however image sizes of 256x256 and 512x512 pixels were also used. The test images were synthesised images of sinusoidal gratings of varying phase, orientation, spatial frequencies and contrast. The orientation of the sinusoidal gratings was calculated from vertical to clockwise direction in all the experiments.

Initial Validation

First, how the variation in the power consumption is affected by the varying the independent variables. The Coefficient of Variation (CV) in the power consumption results was compared amongst the various independent variables. The coefficient of variation or relative standard deviation (RSD) is the ratio of the standard deviation to the mean (average). This statistic shows the measure of spread which describes the amount of variability relative to the mean. Since the statistic is a unitless ratio, it can be used to objectively compare the spread of data sets that have different units or different means, and that is exactly what was done. If the CV of a set of results was found to be statistically significantly less than the others, the variable was omitted from the experiments. The threshold for comparison for the CV was set to be statistically significant to 2%.

Table 4 shows the summary of CVs for all the independent variables. It is quite clear from the table that the variability in the data for independent variables Contrast and Phase is significantly lower (0.08% and 1.24% respectively) than all the other variables. This can only happen if the effect of the Contrast and Phase on the dependent variable was negligible. Therefore, in the experiments, the independent variables Contrast and Phase were omitted.

Table 4. Summary CV for all independent variables

Independent Variable	CV %
Contrast	0.0778143831
Phase	1.2377676786
Orientation	18.498540082
Spatial Frequency	20.647365605
Image Size	12.098000093

Spatial Filter Operation	82.501542221

Moreover, the contrast was normalized to make image calculations independent of the contrast. Since the contrast is given by the Michelson contrast, the contrast normalisation was performed using the contrast stretching method to cover maximum range of an 8-bit pixel value which ranges from 0 to 255. This was carried out by stretching the range of intensity values to make full use of possible values [30].

Experimental Exploration

The main aim here was to investigate the relationship of the image content in the form of the spatial frequencies and the orientations of those spatial frequencies present in an image, with the energy consumption of the hardware implemented spatial filter that was applied to process the same image.

Spatial Frequency

First, the key results showing the impact of spatial frequencies on energy consumption are presented. The spatial frequency was varied with orientation while keeping the image size to 128x128, kernel size to 5x5, clock frequency to 100 MHz and the filtering operation to the template Gaussian Filter. Some of the sample images are shown below. These images were processed in the implemented filter using the simulation based power estimation flow in the System Generator tool. Fig. 7 and Fig. 8 are example images of sinusoidal gratings used in the experimentation.
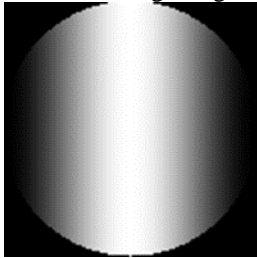

Fig. 7. Sinusoidal grating image with spatial frequency one cycle per image and orientation 0 degrees
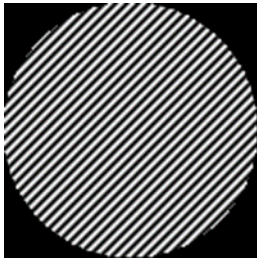

Fig. 8. Sinusoidal grating image with spatial frequency thirty-two cycles per image and orientation 45 degrees

Table 5 shows selected results (Spatial frequencies 0, 1, 2, 4, 8, 16, 32 and 64) from the experiment. The time taken to process one image of 128x128 pixels was 180499 Nano Seconds. The energy efficiency of a plain black image was considered as the base line for the analysis of the results. Here the orientation was fixed to 0 and 90 degrees in order to assess the impact of the variation in spatial frequency. It can be seen from the table that the energy efficiency drops to

12.05% for the maximum spatial frequency 64 cycles per image for a 128x128 image. However, for an orientation of 90 degrees, the energy efficiency is at 72.38% when the spatial frequency of the image is at the maximum.

Table 5. Dynamic energy consumption vs spatial frequency, 0 & 90-degree orientation and image size 128x128

Spatial Frequency, Cycles/Image	Total Power, mW	Dynamic Power, mW	Static Power, mW	Dynamic Energy, uJ	Energy Efficiency, Giga Ops per Joule	% Energy Efficiency
Black	3145.92	11.95	3133.97	2.16	189.90	100
White	3145.92	11.95	3133.97	2.16	189.90	100
1	3184.06	48.95	3135.11	8.84	46.36	24.41
2	3189.94	54.66	3135.29	9.87	41.52	21.86
4	3195.93	60.46	3135.47	10.91	37.53	19.76
8	3205.57	69.81	3135.75	12.60	32.50	17.12
16	3214.91	78.88	3136.04	14.24	28.77	15.15
32	3227.25	90.85	3136.41	16.40	24.98	13.15
64	3235.78	99.12	3136.66	17.89	22.89	12.05
Orientation set to 90 Degrees						
1	3150.2	16.11	3134.09	2.91	140.86	74.18
2	3150.59	16.48	3134.11	2.97	137.70	72.51
4	3150.59	16.48	3134.11	2.97	137.70	72.51
8	3150.84	16.73	3134.11	3.02	135.64	71.43
16	3151.02	16.9	3134.12	3.05	134.27	70.71
32	3151.31	17.18	3134.13	3.10	132.09	69.56
64	3150.62	16.51	3134.11	2.98	137.45	72.38

The energy efficiency in terms of Giga operations per joule versus spatial frequency overlaid with orientation is plotted in Figure 9. The graph shows that the energy efficiency decreases with the increase in spatial frequency. The energy efficiency is at a maximum when orientation is 90 degrees whilst it is at a minimum when orientation is 0 degrees.
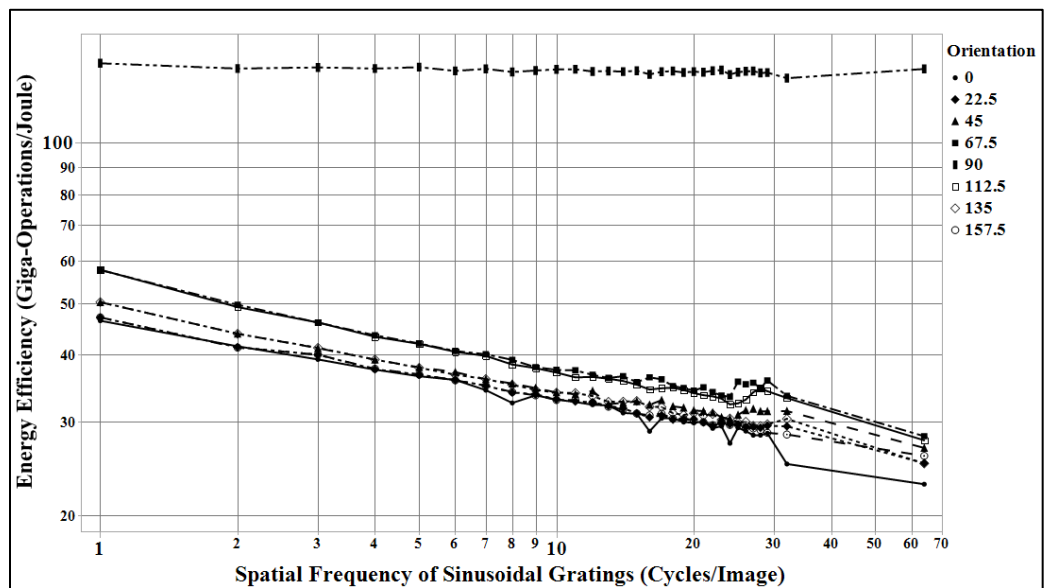
Fig. 9. Energy efficiency vs. spatial frequency, overlaid by orientations, image size 128x128

Image Size 256x256

Table 6 shows selected results (Spatial frequency 0, 1, 2, 4, 8, 16, 32, 64 and 128) from the experiment where the spatial frequency and the orientation were varied while the spatial filter was the template Gaussian filter with kernel size 5x5, image size 256x256 and clock frequency 100 MHz. The time taken to process one image of 256x256 pixels was 721171 Nano Seconds. The energy consumption and energy efficiency of the black image was considered as the base line for the analysis of the results. Here, the orientation was fixed to 0 and 90 degrees in order to assess the impact of the variation in spatial frequency. The energy efficiency was considered in terms of Giga operations per joule. It can be seen from the table that the energy efficiency drops to 14.46% for the maximum spatial frequency 128 cycles per image for 256x256 image. However, for an orientation of 90 degrees, the energy efficiency is at 86.73% when the spatial frequency of the image is at the maximum.

Table 6. Dynamic energy consumption vs spatial frequency, 0 & 90 degrees orientations and image size 256x256

Spatial Frequency, Cycles/Image	Total Power, mW	Dynamic Power, mW	Static Power, mW	Dynamic Energy, uJ	Energy Efficiency, Giga Ops per Joule	% Energy Efficiency
Black	3150.44	16.34	3134.1	11.78	139.04	100
1	3183.35	48.26	3135.09	34.80	47.07	33.86
2	3190.66	55.35	3135.31	39.92	41.04	29.52
4	3197	61.5	3135.5	44.35	36.94	26.57
8	3203.15	67.47	3135.68	48.66	33.67	24.22

16	3213.89	77.88	3136	56.16	29.17	20.98
32	3223.7	87.4	3136.3	63.03	25.99	18.70
64	3236.25	99.57	3136.68	71.81	22.82	16.41
128	3250.05	112.96	3137.09	81.46	20.11	14.46
Orientation set to 90 Degrees						
1	3152.84	18.66	3134.17	13.46	121.7503	87.57
2	3153.06	18.88	3134.18	13.62	120.3316	86.55
4	3153.03	18.85	3134.18	13.59	120.5231	86.68
8	3153.12	18.94	3134.18	13.66	119.9504	86.27
16	3153.17	18.99	3134.18	13.70	119.6346	86.05
32	3153.22	19.04	3134.19	13.73	119.3204	85.82
64	3153.4	19.21	3134.19	13.85	118.2645	85.06
128	3153.02	18.84	3134.18	13.59	120.5871	86.73

The energy efficiency considered in terms of Giga operations per joule versus spatial frequency overlaid with orientation is plotted in Figure 10. The graph shows that the energy efficiency decreases with the increase in spatial frequency. The energy efficiency is at maximum when orientation is 90 degrees whilst it is at minimum when orientation is 0 degrees.
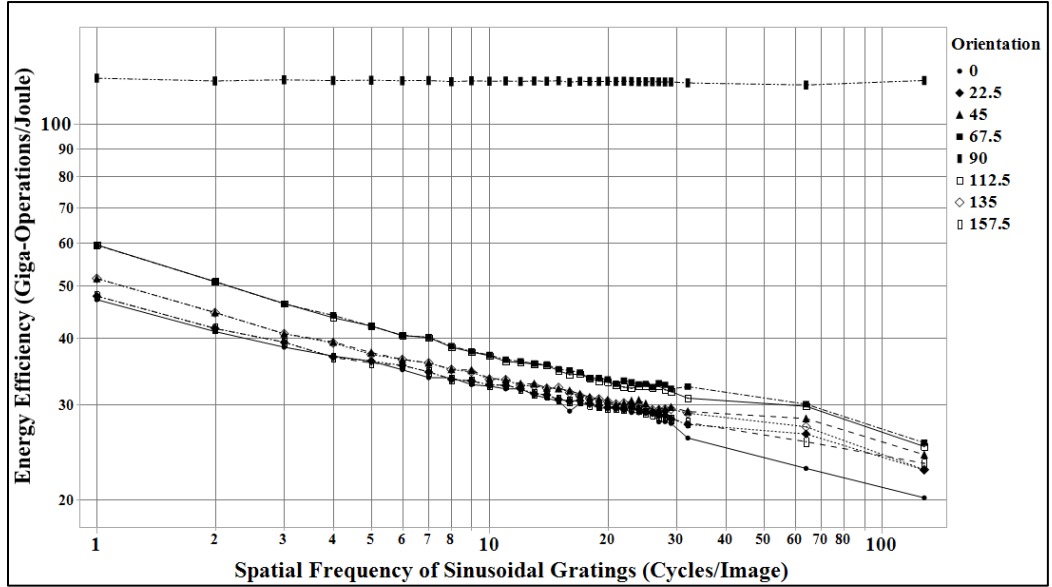

Fig. 10. Energy efficiency vs. spatial frequency, overlaid by orientations, image size 256x256

Image Size 512x512

Table 7 shows selected results (Spatial frequency 0, 1, 2, 4, 8, 16, 32, 64, 128 and 256) from the experiment where the spatial frequency and the orientation were varied while the spatial filter was the template Gaussian filter with kernel size 5x5, image size 512x512 and clock frequency 100 MHz. The time taken to process one image of 512x512 pixels was 2883859 Nano Seconds. The energy consumption and energy efficiency of a black image was considered as the base line for the analysis of the results. Here, the orientation was fixed to 0 and 90 degrees in order to assess the impact of the variation in spatial frequency. The energy efficiency was considered in terms of Giga

operations per joule. It can be seen from the table that the energy efficiency drops to 15.14% for the maximum spatial frequency 256 cycles per image for a 512x512 image. However, for orientation 90 degrees, the energy efficiency is at 93.76% when the spatial frequency of the image is at the maximum.

Table 7. Dynamic energy consumption vs spatial frequency, 0 & 90 degrees orientations and image size 512x512

Spatial Frequency, Cycles/Image	Total Power, mW	Dynamic Power, mW	Static Power, mW	Dynamic Energy, uJ	Energy Efficiency, Giga Ops per Joule	% Energy Efficiency
Black	3154.98	20.74	3134.24	59.81	109.57	100
1	3183.3	48.21	3135.09	139.03	47.14	43.02
2	3192.2	56.85	3135.35	163.94	39.97	36.48
4	3200.57	64.97	3135.6	187.36	34.98	31.92
8	3208.04	72.21	3135.83	208.24	31.47	28.72
16	3214.48	78.46	3136.02	226.27	28.96	26.43
32	3227.62	91.21	3136.42	263.04	24.91	22.74
64	3236.86	100.17	3136.69	288.88	22.69	20.70
128	3250.25	113.15	3137.1	326.31	20.08	18.33
256	3274.76	136.93	3137.83	394.89	16.59	15.14
Orientation set to 90 Degrees						
1	3156.6	22.32	3134.29	64.37	101.81	92.92
2	3156.71	22.42	3134.29	64.65	101.36	92.51
4	3156.41	22.12	3134.28	63.79	102.73	93.76
8	3156.41	22.13	3134.28	63.82	102.69	93.72
16	3156.44	22.16	3134.28	63.91	102.55	93.59
32	3156.46	22.18	3134.28	63.96	102.46	93.51
64	3156.52	22.24	3134.28	64.14	102.18	93.25
128	3156.9	22.6	3134.3	65.17	100.55	91.77
256	3156.4	22.12	3134.28	63.79	102.74	93.76

The energy efficiency considered in terms of Giga operations per joule versus spatial frequency overlaid with orientation is plotted in Figure 11. The graph shows that the energy efficiency decreases with an increase in spatial frequency. The energy efficiency is at a maximum when the orientation is 90 degrees whilst it is at minimum when the orientation is 0 degrees.
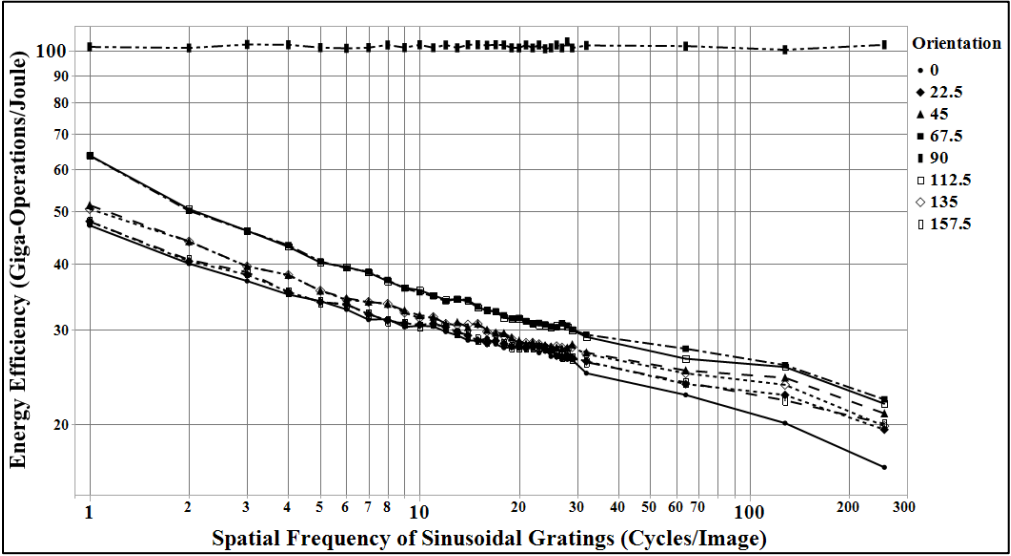
Fig. 11. Energy efficiency vs. spatial frequency, overlaid by orientation, image size 512x512

All Image Sizes

In order to isolate the impact of image size on the dependent variable, images of varying sizes were synthesised. The spatial frequency of the sinusoidal grating in the image was set to 1 cycle per image, the clock frequency to 100MHz, the phase to 90 degrees and the orientation to 0 degrees. Images with varying sizes of 16x16, 32x32, 64x64, 128x128, 256x256, 512x512 and 1024x1024 pixels were synthesised. Since the width of the line buffers in the spatial filter changes with the width of an image, dedicated template spatial filters with line buffers of different sizes to accommodate each of these different image sizes were developed and implemented. These images were processed in the implemented filters and power consumption was estimated in the System Generator tool. Figure 12 shows the graph plot between dynamic power consumption and image size in a linear-log scale.
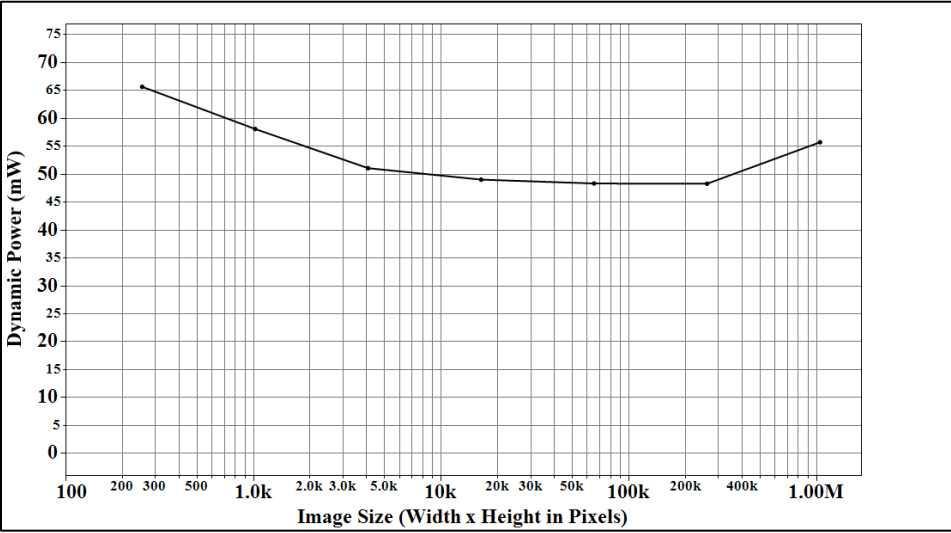

Fig. 12. Graph of dynamic power consumption (mW) vs. size of the image in pixels

It is important to note in the graph that, at lower sizes of images such as 16x16, 32x32 and 64x64, more dynamic power has been shown to have been consumed than with some of the larger sizes. This is a counter intuitive result because, generally, the increase in the image size increases the amount of logic to store and process the image which should result into increased dynamic power consumption. When this was investigated in detail in the power analysis reports, this was found to be largely due to the power consumed by the primary inputs and outputs (IO). The IO power is a component of the total dynamic power consumption. The dynamic power consumption and IO power consumption for each image size is presented in Table 8. In the case of the image of 16x16 pixels, the IO power is almost 70% of the dynamic power:

Table 8. Dynamic power and IO power for varying image sizes

Image Size	Dynamic Power mW	IO Power mW	IO Power, % of dynamic power
16x16	65.57	45.21	69.0
32x32	58.02	38.02	65.5
64x64	51.0	30.36	59.5
128x128	48.95	24.18	49.4
256x256	48.26	17.80	36.9
512x512	48.21	12.04	25.0
1024x1024	55.64	6.93	12.5

These images were processed with varying orientations in the template 5x5 Gaussian filter and power consumption was estimated in the System Generator tool. Figure 13 is a graph of dynamic IO power vs image size in a linear-log scale. As the image size is increased the IO power is decreasing. This can only be possible if the number of IO switching in a unit time is more for the smaller image than it is for the larger image. This is explained in detail in the analysis section.
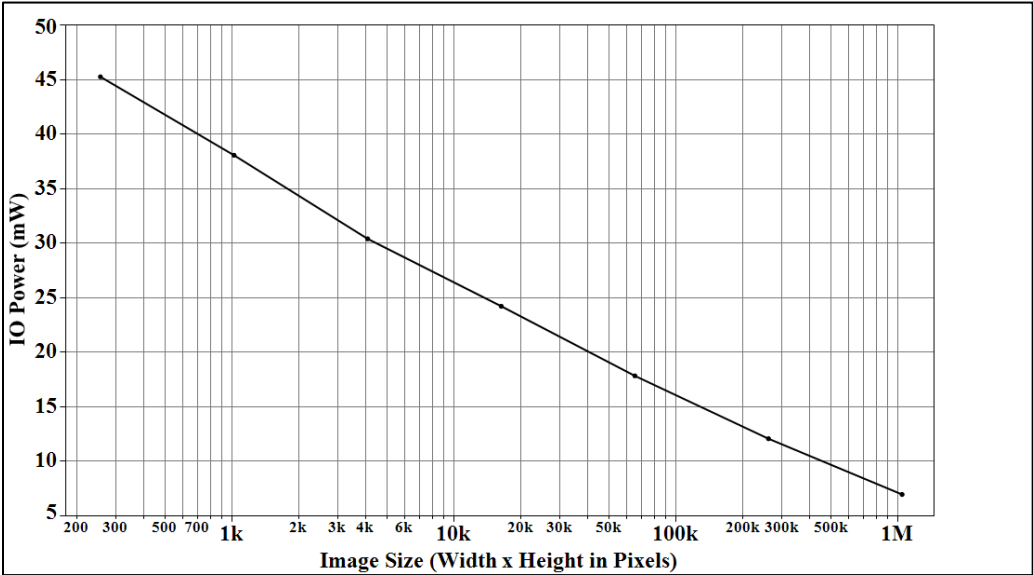

Fig. 13. Dynamic IO power consumption vs size of the image

However, when the graph of image size against the energy consumed to process the image was plotted, the graph follows the intuition whereby the consumed energy increases with the image size as shown in Figure 14. The graph is plotted using a log-log scale on X and Y axes.
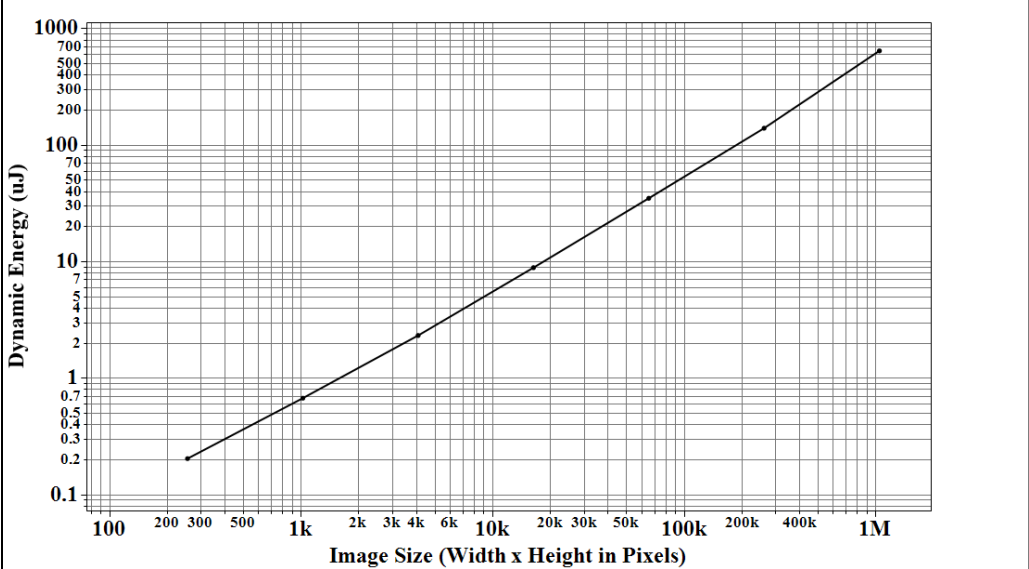

Fig. 14. Dynamic energy consumption vs size of the image

Figure 15 is a graph which plots dynamic energy against varying image sizes (16x16, 32x32, 64x64, 128x128, 256x256, 512x512 and 1024x1024 pixels) while varying spatial frequencies and orientation, using a log-log scale. Here the range of the spatial frequencies are from 1 to 4 because the minimum image size that is explored is 16x16. The energy consumption increases with the increase in image size.
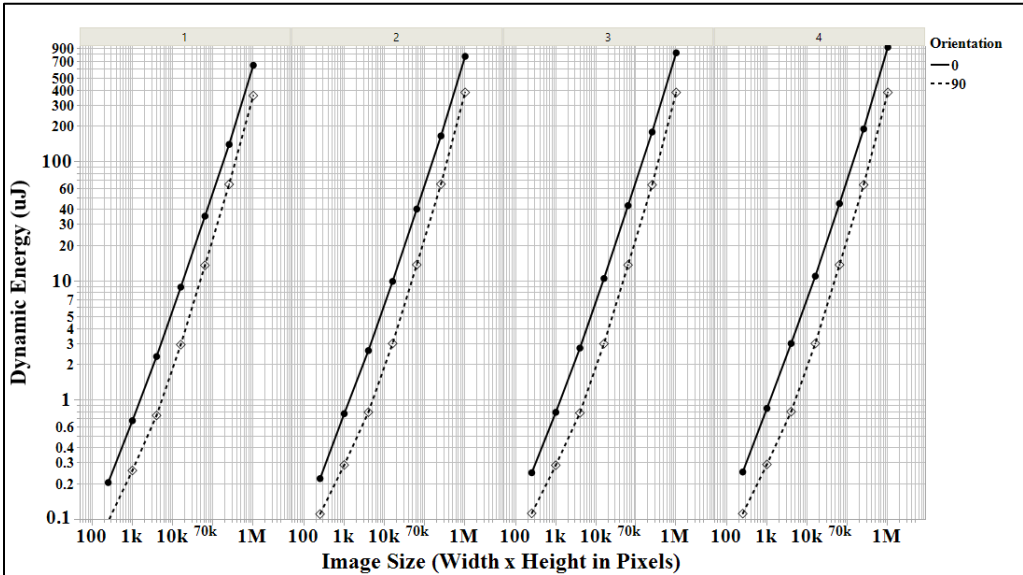

Fig. 15. Dynamic energy consumption vs image size while varying spatial frequency and orientation

Figure 16 shows the graph of energy efficiency in Giga operations per Joule versus image size overlaid with spatial frequency and orientation set to 0 degrees. This graph is plotted using a linear scale on its Y axis and log scale on its X axis to better represent the data. As can be seen the energy efficiency increases for smaller images however decreases for larger images as the image size and spatial frequency increases.
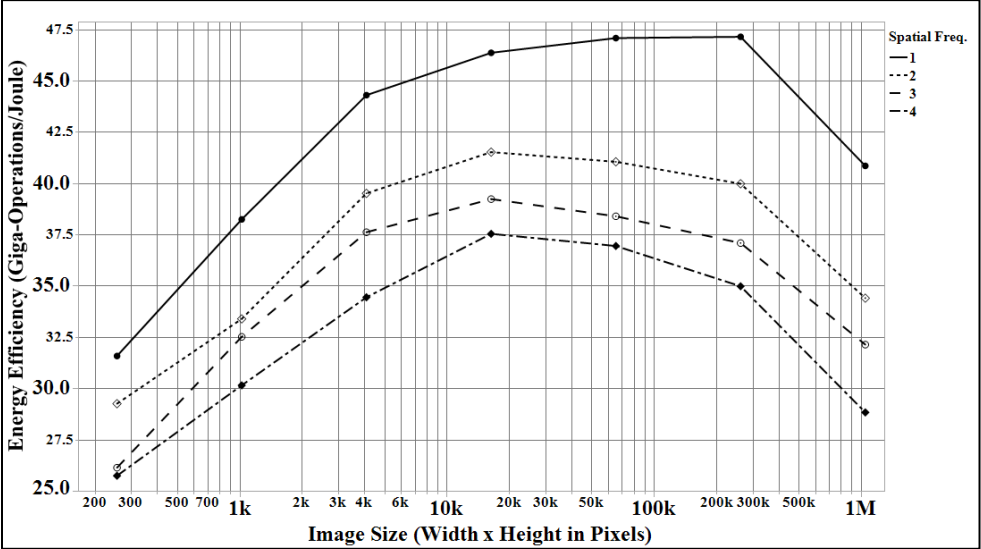

Fig. 16. Energy efficiency vs image size while varying spatial frequency and orientation is set to 0 degrees

Figure 17 shows the graph of energy efficiency in Giga operations per Joule versus image size overlaid with spatial frequency and orientation set to 90 degrees. This graph is plotted using a linear-log scale. As it can be seen that the energy efficiency increases for smaller images however decreases for larger images as the image size and spatial frequency increases.
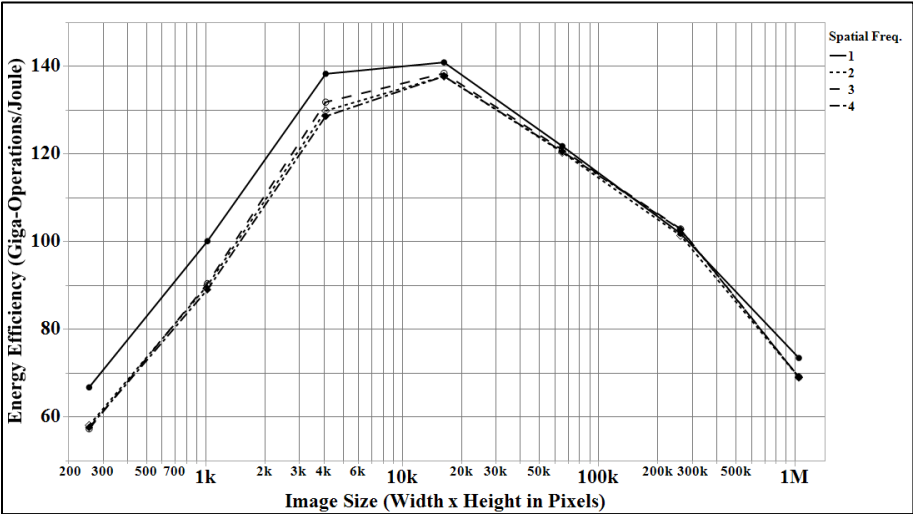

Fig. 17. Energy efficiency vs image size while varying spatial frequency and orientation is set to 90 degrees

Orientation

Table 9 shows selected results (Orientations 0, 30, 45, 60, 90, 120, 135, 150 out of 0, 11.25, 22.5, 30, 33.75, 45, 56.25, 60, 67.5, 78.75, 90, 101.25, 112.5, 120, 123.75, 135, 146.25, 150, 157.5 and 168.75 degrees) from the experiment where the orientation was varied and the spatial frequency was set to 1 and 32 for image size of 128x128 and the template Gaussian filter with kernel size 5x5 and clock frequency 100 MHz. The time taken to process one image of 128x128 pixels was 180499 Nano Seconds. The energy consumption and energy efficiency of the black image was considered as the base line for the analysis of the results. The energy efficiency was considered in terms of Giga operations per joule.

It can be seen from the table that for spatial frequency one cycle per image the energy efficiency drops to 24.41% for the 0 degrees orientation however peaks at 74.17% 90 degrees orientation. However, for spatial frequency 32 cycles per image, the energy efficiency is at a minimum, at 13.15% at 0 degrees orientation and peaks at 69.55% at 90 degrees orientation.

Table 9. Dynamic energy consumption vs orientation, spatial frequency 1 & 32 cycles per image and image size 128x128

Orientation, degrees	Total Power, mW	Dynamic Power, mW	Static Power, mW	Dynamic Energy, uJ	Energy Efficiency, Giga Ops per Joule	% Energy Efficiency
Spatial Frequency set to 1 cycle per image						
Plain Black Image	3145.92	11.95	3133.97	2.16	189.90	100
0	3184.06	48.95	3135.11	8.84	46.36	24.41
30	3182.91	47.84	3135.08	8.64	47.43	24.98
45	3180.19	45.19	3134.99	8.16	50.21	26.44
60	3177.3	42.39	3134.91	7.65	53.53	28.19061
90	3150.2	16.11	3134.09	2.91	140.87	74.17754
120	3177.33	42.43	3134.91	7.66	53.48	28.16404
135	3180.16	45.16	3134.99	8.15	50.25	26.46147
150	3182.83	47.76	3135.07	8.62	47.51	25.02094
Spatial Frequency set to 32 cycles per image						
0	3227.25	90.85	3136.41	16.39833	24.98	13.15
30	3211.74	75.8	3135.94	13.68182	29.94	15.77
45	3208.11	72.28	3135.83	13.04647	31.40	16.53
60	3206.03	70.26	3135.77	12.68186	32.30	17.01
90	3151.31	17.18	3134.13	3.100973	132.09	69.56
120	3206.53	70.74	3135.78	12.7685	32.08	16.89
135	3210.68	74.77	3135.91	13.49591	30.35	15.98
150	3214.92	78.89	3136.04	14.23957	28.76	15.15

Figure 18 is the graph between energy efficiency in Giga operations per Joule and various orientation values in degrees. Again, it is important to note that the energy efficiency is at a maximum when the orientation is 90 degrees while it is at a minimum when the orientation is 0

degrees in images with both the spatial frequencies. Therefore, in the experiments, the orientations were limited to 0 and 90 degrees as this covered the entire population.
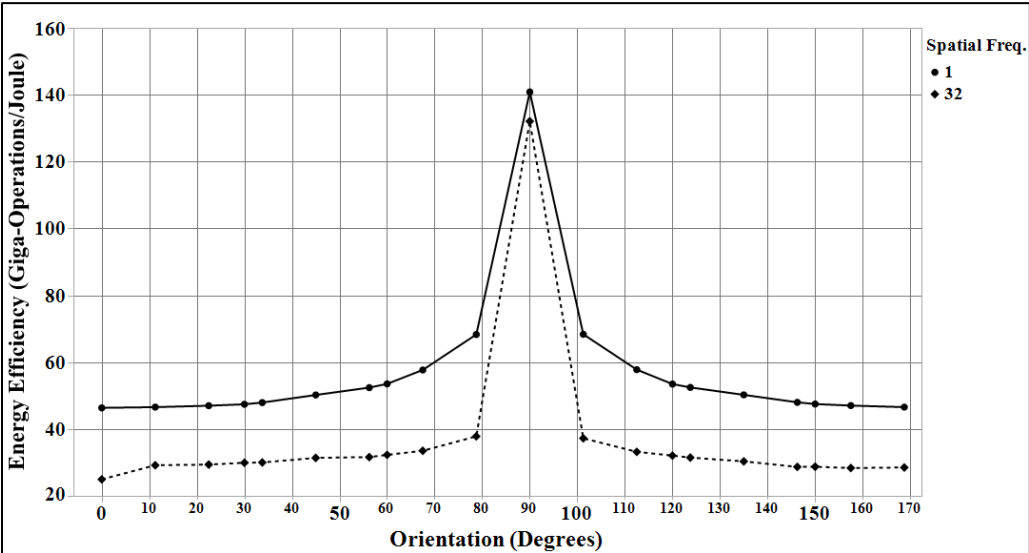

Fig. 18. Energy efficiency vs Orientations in degrees

Image Operations

In this section, the impact of energy consumption on various hardware accelerated spatial filter operations is explored. These include line buffers, Difference of Gaussian (DoG), SIFT Detector, Gaussian 3x3, Gaussian 5x5, Gaussian 7x7, Gaussian 9x9, Gaussian Separable 5x5, Laplacian 3x3, Mean Filter 5x5, Median Filter 3x3, Morphological Filter 5x5 and Sobel Filter 3x3.

Figure 19 and Figure 20 show the graphs of dynamic energy efficiency given by Giga Ops Per Joule against various filter operations while spatial frequencies are varied and orientation is set to 0 and 90 degrees respectively. The energy efficiency for 0-degree orientation shows a slightly decreasing trend as the spatial frequency increases. Here the most complex image processing pipeline is the SIFT detector which consumes the largest amount of power and hence it is the least energy efficient. Whereas the energy efficiency for 90 degrees orientation is nearly constant for spatial frequency 1 onwards. It is important to note that all the lines in the graph follow the general curve as explored previously with the template spatial filter of Gaussian 5x5. This means the results for the template filter can be generalized for any spatial filter which follows the same architecture.
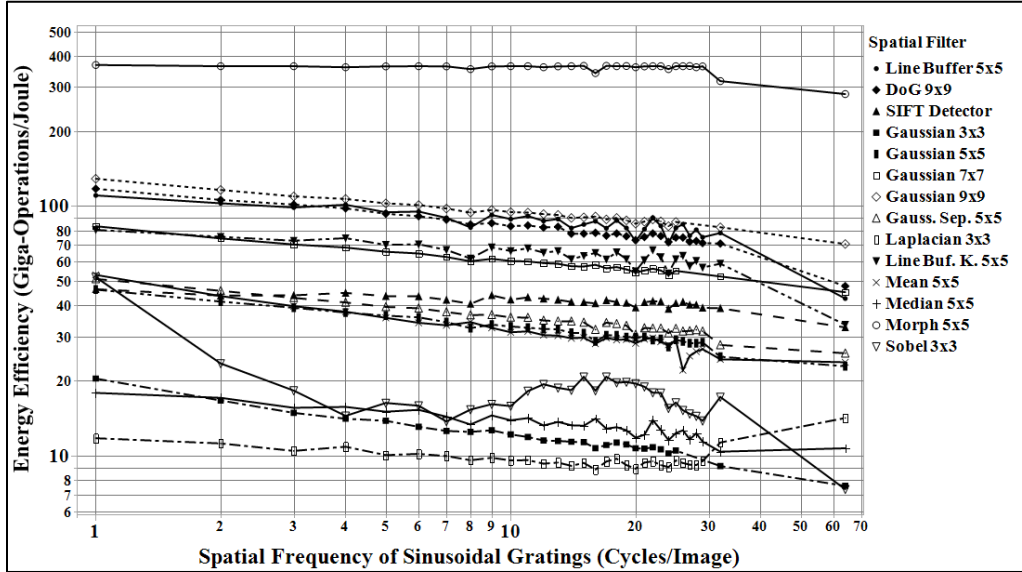
Fig. 19. Energy efficiency vs spatial frequency and filter operations while the orientation is set to 0 degrees
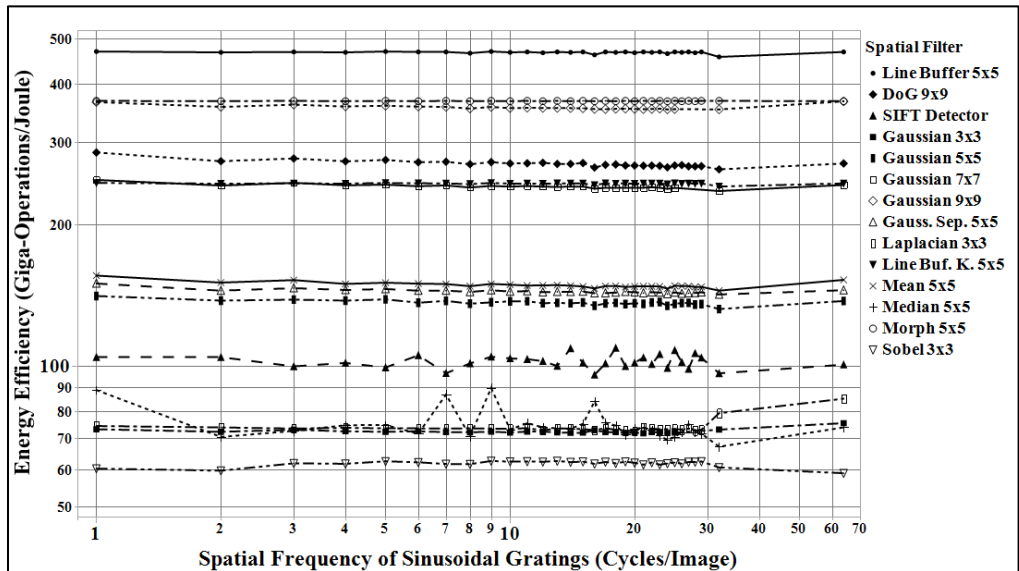

Fig. 20. Energy efficiency vs spatial frequency and filter operations while the orientation is set to 90 degrees

Analysis

This section presents the analysis of the results and explains the results.

Impact of Spatial Frequency

It is important to note that, as shown in the graph in Figure 9, Figure 10 and Figure 11, the maximum power consumption occurs when the spatial frequency of the image is 64 sinusoidal gratings. As defined by the Nyquist-Shannon theorem, the maximum number of sinusoidal grating that can be fitted into an image is image width divided by two. Therefore, for an image of size 128x128, the maximum sinusoidal gratings that can be fitted is 64. This creates an image of a pixel wide black and white stripes at 0 degrees orientation which is an image of two-dimensional square wave i.e. vertical bars/stripes of black and white. Since the value of a white 8-bit pixel is 255 in decimal or 0xFF in hexadecimal and the value of a black pixel is 0 in decimal or 0xFF in hexadecimal, traversing in a horizontal direction in the image, the pixel values change from 0x00 to 0xFF and 0xFF to 0x00 which means, every bit of the 8-bit wide pixel changes from 0 to 1 and 1 to 0. This results in a maximum number of transitions at the I/O ports of the FPGA when the image is scanned in row-major order. This also contributes to the maximum amount of switching in logic.

Moreover, the power consumption of black and white images is at the lowest and is almost the same because pixel values in the black and white images do not change. In a black image, all the pixel values are 0x00 and in a white image all the pixel values are 0xFF without any variation in them.

In the middle, the power consumption generally increases with the spatial frequency of the image, this is mainly due to the variation in the content which is due to the variation in the spatial frequencies present in the image and this means there is variation in the pixel values which then contribute to the amount of switching at the I/O ports and in the logic. This is defined by the switching activity factor α_t.

In order to understand the effect of the spatial frequency as shown in the middle region in the graph, where the energy consumption increases with increase in spatial frequency as shown in Figure 9, Figure 10 and Figure 11, the two main factors that impact the amount of switching in the circuit should be understood. The transition density and static transition probability of every bit of an 8-bit binary number because a pixel is typically represented in as an 8-bit binary number in a digital image. The transition density of a signal, denoted by α_t, is given by the average number of transitions of the signal per unit time. . The static transition probability of the signal is the probability of the signal being high at any given time [28]. As seen previously in Figure 3, it is in the transition from 0 to 1, the current is drawn into the circuit which contributes to the power consumption of the circuit.

An 8-bit binary number ranges from 0 to 0xFF (255_{10}). The normal binary sequence goes from 00000000_2 to 11111111_2.

Switching activity, $P_{0->1}$ has two components a static component that is a function of the logic topology and a dynamic component which is a function of the timing behaviour of the logic circuit (includes glitching). In this paper, only the static component is considered for two reasons. Firstly, the dynamic component, for an example glitching, depends on the exact implementation of the logic circuit which cannot be foreknown and secondly, to limit the scope of this research work.

The static transition probability of a binary single bit can be given by:

$$P_{0->1} = P_{out=0} \times P_{out=1} \tag{7}$$

Where $P_{0->1}$ is the probability of the output bit to transition from 0 to 1, $P_{out=0}$ is the probability of the output bit to be 0 and $P_{out=1}$ is the probability of the output bit to be 1.

Moreover, in an 8-bit binary number, as moving from the Least Significant bit (LSb) to the Most Significant bit (MSb), the significance of the bits in the binary number and as a result the value of the 8-bit binary number is given by $= 2^7 \times bit7 + 2^6 \times bit6 + 2^5 \times bit5 + 2^4 \times bit4 + 2^3 \times bit3 + 2^2 \times bit2 + 2^1 \times bit1 + 2^0 \times bit0$.

Therefore, as moving from LSb to MSb, each bit switches from 0 to 1 in a decreasing order. As an example, the LSb toggles from 0 to 1 and 1 to 0 alternatively, however the second bit going from right to left, toggles every $1/2^n$ times the toggle rate of the LSb, where n is the bit position. Therefore, the transition density decreases by half moving from LSb to MSb.

Therefore, for a binary number represented with more than one-bit width, 8-bit in this case, each bit has a different static transition probability. For an example, the probability of transition $P_{LSB(0->1)}$ of the Least Significant bit (LSb) can be given by multiplying the probability of the bit being '0', P_0 and the bit being '1', P_1 .

The static transition probability of the LSb in an 8-bit binary number can be given by:
$P_{LSB(0->1)} = (128/256) \times (128/256) = 1/4$.

Moving from LSb to MSb, the static transition probability reduces by 4. Therefore, the individual static transition probabilities of each bit in an 8-bit binary number is given in Table 10 [7].

Table 10 Static transition probabilities of each bit in an 8-bit binary number

Bit7 (MSb)	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0 (LSb)
1/65536	1/16384	1/4096	1/1024	1/256	1/64	1/16	1/4

However, in the case of the spatial frequency being the maximum, 64, the pixel transition from black to white alternatively which is 00000000 to 11111111 and back to 00000000. This means the static transition probability of each bit is at the maximum of 1/4, which is the same as Bit0, the LSb. Also, the transition density of each of the bits is same as the LSb. This means as the spatial frequency is increased, the number of 8-bit binary values, samples, that represent a cycle of the two-dimensional sinusoidal grating is reduced.

Impact of Image Size

The same thing happens when the image size is reduced but the spatial frequency is kept constant. This means that the transition density and static transition probabilities of the bits going from LSb to MSb, right to left, increases and reaches at maximum 1/4 depending on the spatial frequency or the size of the image. Transition density together with the static transition probability, increases the switching at the primary IO ports of the spatial filter as the input to the filter is the streaming of 8-bit pixel scanned from the image in row-major order.

This is the reason why the IO power consumption increases when the spatial frequency of the sinusoidal gratings is increased, or image size is reduced while keeping the spatial frequency constant. However, since the amount of logic used in implementing the spatial filter in hardware to process a smaller say 16x16 image is considerably less than a larger say image of 1024x1024 size image, the impact of the power consumed by the logic is not as significant as the power consumed by the IO. Therefore, for smaller images, the IO power dominates the total power consumption.

Similarly, for larger image size the proportion of the IO power in the total power consumption is reduced as the image size is increased for a given spatial frequency as the static transition probabilities of middle bits (between LSb and MSb) reduces.

Signal Rate and Transition Density

Let us explore another empirical evidence to study the impact of the signal rate or transition density, by way of extracting the switching activity information from the Xilinx Power Analyser (XPA) for the input and output ports. In order to extract this information, the information provided under the term "signal rate" in the XPA power consumption report was used. Xilinx [41] defines the signal rate by the number of millions of transitions per second (2xClockRate in MHz) for the signal under consideration.

Since the clock frequency is kept fixed in most of experiments, only the pixel values at the 8-bit primary input (gateway_in) and output (gateway_out) ports of the spatial filter hardware implementation are considered. For an example the gateway_in port from the most significant bit gateway_in (7) to the least significant bit gateway_in(0) and similarly for the gateway_out port. In a clock driven synchronous design, the maximum value of transitions can be given the half of the input clock frequency. Therefore, in the case that the clock frequency is 100MHz and the transitions for the same are 200 Million Transactions per second (Mtps) while the data only changes once every clock cycle the data transmission rate would be 100 Mtps maximum. If the synchronous design has components that change on each of the clock edges, i.e. positive and negative, then for the given clock frequency of 100MHz, the data signal rate would be 200Mtps. Figure 21 shows the graph of the mean of each bit in the 8-bit input pixel values given in Mtps versus the image size and confirms the theoretical findings as explained. This graph is plotted using a linear-log scale. As it can be seen in the graph, the signal rate of the most significant bits increases as the image size decreases thus increasing the power consumed in the IO ports. Here the spatial frequency is fixed to one cycle per image and the orientation is set to 0 degrees.
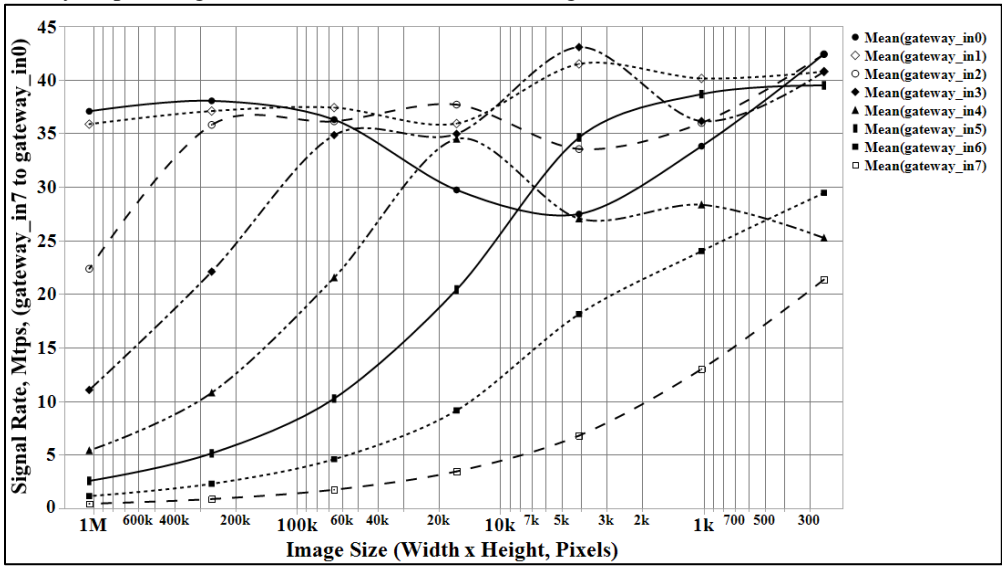

Fig. 21 Signal rate at the 8-bit image pixel input port gateway_in vs image size

Figure 22 shows the same impact of varying image size on the output port gateway_out, using a linear log scale. The mean signal rate of each of the bits in the 8 bits of gateway_out increases from least significant bit to the most significant bit as the image size decreases.
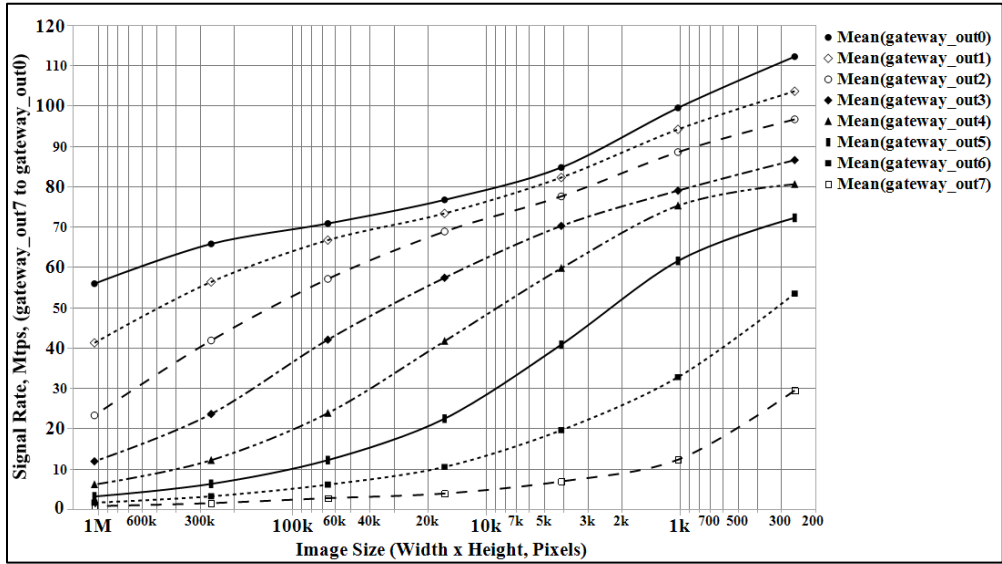
Fig. 22. Signal rate at the 8-bit image pixel output port gateway_out vs image size

For the case where the spatial frequencies are varied by providing varying frequency grating images to the spatial filter, Figure 23 shows the graph of signal rate of each of the gateway_in bits versus spatial frequency, using a linear-log scale. Regression lines are fitted through each of the data points of gateway_in bits to extract the trend. It can be observed from the graph that as the spatial frequency increases, the trend is that there is an increase in the signal rate from the least significant bit (gateway_in(0)) to the most significant bit (gateway_in(7)) in gateway_in input port. Here the orientation is fixed to 0 degrees.
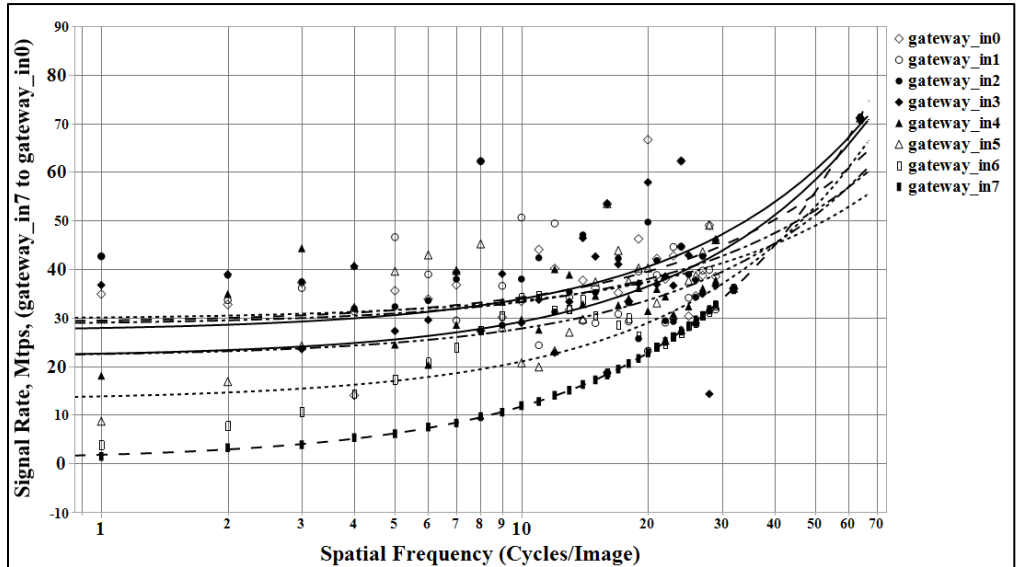

Fig. 23. Signal rate at the 8-bit image pixel input port gateway_in vs spatial frequency, orientation 0 degrees

The same general trend of increasing of the signal rate continues for the pixel output port
gateway_out of the spatial filter as shown in Figure 24. This graph is plotted using a linear-log
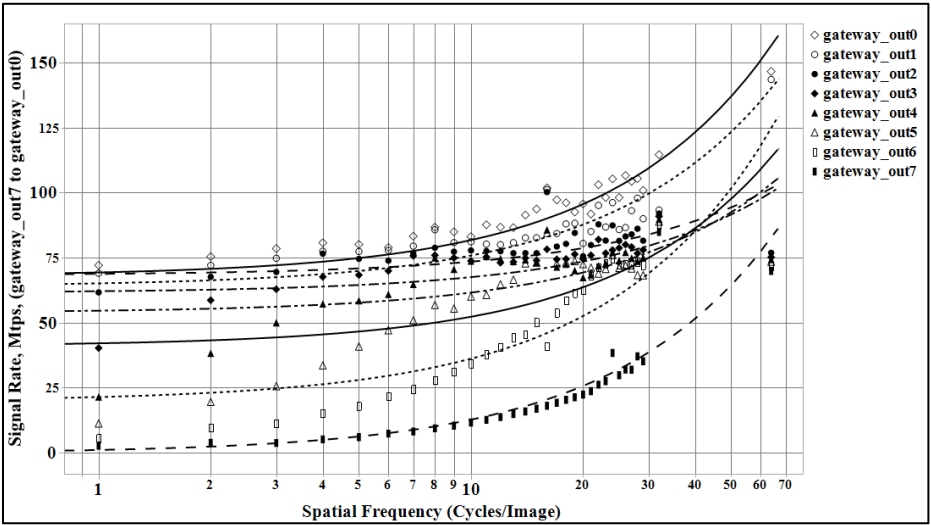scale.

Fig. 24. Signal rate at the 8-bit image pixel output port gateway_out vs spatial frequency, orientation 0 degrees

In the case where the spatial frequency of an image is varied while keeping the size of the image
fixed, since a larger image requires more logic in the spatial filter to process the image, the rise in
spatial frequency increases the power consumption in the logic thus increasing overall power and
energy consumption. This can be seen in Figure 25 where the area occupied by the hardware
implemented spatial filter in terms of FPGA slices versus the size of an image to be processed is
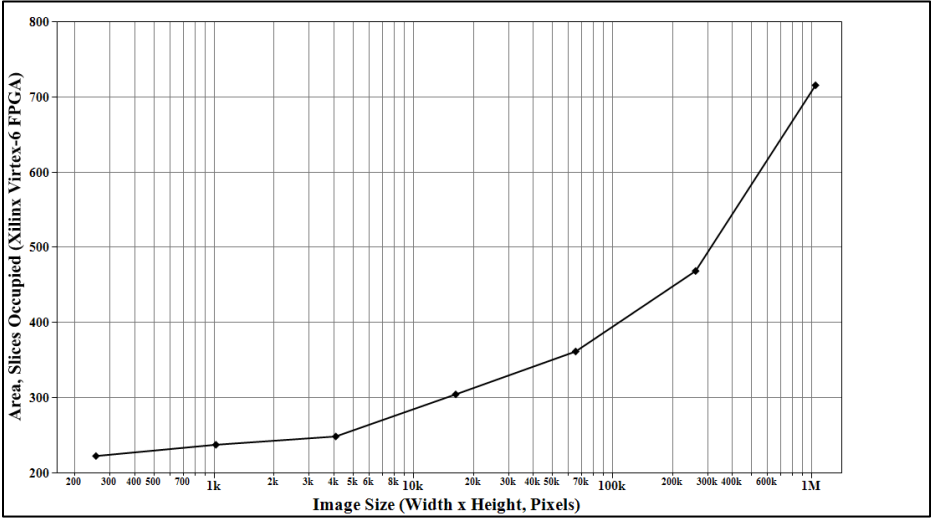plotted. The graph uses a log-log scale.

Fig. 25. FPGA area occupied by the spatial filter implementation vs Image size

However, the power efficiency which is power consumed multiplied by the time to process the image, is given by energy consumption and is as expected whereby smaller images take less energy to process in comparison with larger images as shown in Figure 25, of dynamic energy consumption vs the image size.
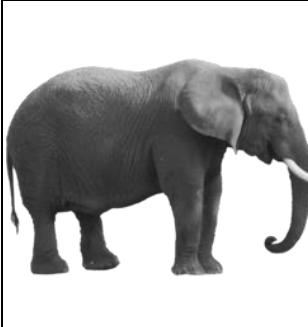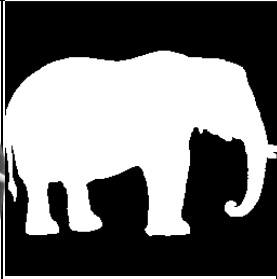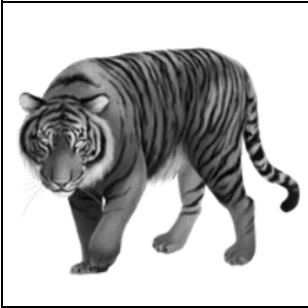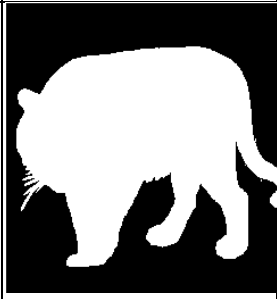
Impact of Orientation

In order to assess the impact of the orientation on the energy efficiency of the spatial filter in processing the image we need to refer back to the anatomy of the image and the hardware implementation of a spatial filter. In an image of a sinusoidal grating with 0 degrees orientation, discounting the black circular mask in the image, the change in content i.e. the variation in the content as given by a change in the pixel values is only in the horizontal direction however, in the vertical direction, the variation in the content is zero and the pixel values remain constant. Now, in the template spatial filter, the pixels are scanned in row-major order. Therefore, when the pixel values are presented at the input ports of the hardware implemented spatial filter, the variation in the values of pixels as experienced by the spatial filter is maximum. This increase in switching at the IO ports, contributes to the IO power consumption in the FPGA and contributes to the dynamic power consumption in the logic due to the increased amount of switching. However, when the sinusoidal grating is aligned horizontally, i.e. orientation at 90 degrees, the change in pixel values is non-existent in the horizontal direction and thus when the image is in scanned row-major order, the values presented at the IO ports of the spatial filter do not switch in the same amount as any other orientation. This has a direct impact on the power and energy consumption of the spatial filter used to process the image. If the spatial filter scanned the image in a column major order, the effect would be reversed.

Validation on Natural Images

Given any two images of a Tiger and an Elephant as shown in figures below, one could pose a question as to: whether the image of the Tiger would consume a different amount of energy than that of the Elephant while filtering them using the same digital circuit?

Table 11 shows the energy consumption of an image of an Elephant and a Tiger. These images were processed in the Gaussian 5x5 template spatial filter and the energy consumption was estimated. Care was taken to ensure that the Region Of Interest (ROI), in this case, the elephant and the tiger, was re-sized to have a very similar area in order for an objective comparison to happen. As seen in the table, there is a clear difference in the power and energy consumption between the image of the Elephant and the Tiger. Here, the image of the Tiger consumes more power and energy than the Elephant.

Table 11. Power and energy consumption of Elephant versus Tiger

Image, 256x256	ROI	ROI area, pixels	Spatial Filter Operation	Power Consumption mW	Energy Consumption uJ
		~2.9e+04	Gaussian 5x5	40.38	29.12088
		~2.9e+04	Gaussian 5x5	47.56	34.29889

Analysis

This difference in the energy consumption between the image of Tiger and Elephant could be explained by Figure 26 and Figure 27 which shows the graph of the signal rate of 8 bits of gateway_in input port and the 8 bits of gateway_out output port versus various images of size 256x256 pixels of the two animals Elephants (7 arbitrary greyscale images of Elephants) and Tigers (4 arbitrary greyscale images of Tigers). Here, the Region of Interest is not exactly scaled to be similar in the area however the size of the images was kept the same. This is mainly because the impact of the images of animals on the signal rate of the IO ports was observed. Graph of mean of each of the bits of input port gateway_in and output port gateway_out were plotted for signal rate. It is quite clear that the general trend in the signal rate between the elephant and the tigers in both input and output ports is rising. The only reason this could be explained is that the images of the Tigers have higher spatial frequencies present in them due to the stripes of the tigers however since the images of the elephants are predominantly shades of grey their spatial frequencies are lower. These differences in spatial frequencies contribute to the differences in the IO port signal rate which then contribute to the power and energy consumption of the spatial filtering circuit.
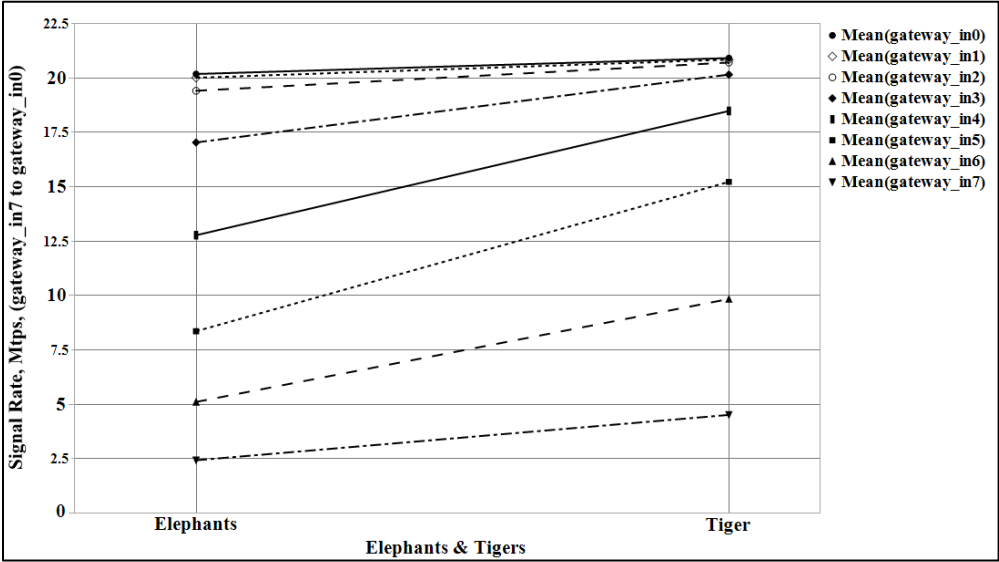
Fig. 26. Signal rate at the input port gateway_in vs images of Elephants and Tigers
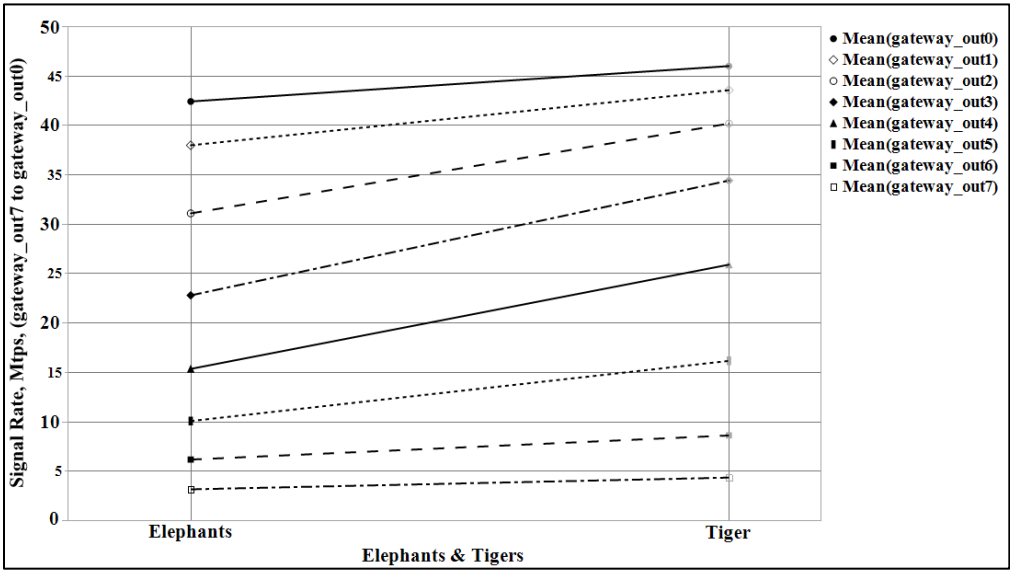

Fig. 27. Signal rate the output port gateway_out vs images of Elephants and Tigers

Conclusion

An experimental framework was developed comprising of a library of spatial filters implemented in hardware and a reference dataset. This included the development of software utilities to customise the spatial filters automatically in order to create hardware design instances on which to perform the empirical exploration. Accordingly software utilities were developed to create a dataset of synthetic images comprising of two-dimensional sinusoidal gratings and utilities to automate the experimental process. The developed HW library of spatial filters was deployed in the respective series of experiments conducted in this research to enable the empirical demonstration of the results. Thus a reference framework has been established for quantification of image content, an image

content processing complexity metric, for modelling the computational energy efficiency in digital
processing of images. The results of experiments have shown that the hardware accelerated spatial
filter consumed more energy to process an image with a higher complexity metric e.g. the image of
a Tiger required more energy to process than the image of an Elephant. This is because of the
spatial frequencies present in the image of the Tiger due to its stripes. These spatial frequencies
contribute to a higher number of switching of signals as required in processing the image; thus
increasing the overall dynamic power consumption. Some of the notable contributions made in this
paper by empirical demonstration are:

- Even a plain grey image consumes dynamic power when processed in a digital circuit.
  This is mainly due to the inherent switching present in the pixels represented in binary
  format.
- The impact of contrast and phase in a sinusoidal grating image on the dynamic power
  consumption of a spatial filter is not statistically significant.
- The maximum amount of energy is consumed when the orientation of the sinusoidal
  grating in the image is at 0 degrees and the least energy is consumed when the orientation
  is at 90 degrees. It was discovered that this effect was due to the row-major order scanning
  of the image and the horizontal symmetry of the hardware blocks to store the image rows.
- The variation in the spatial frequency of an image has a significant impact on the energy
  efficiency of the spatial filter used for processing it. It was confirmed that this was due to
  the gradual increase in the transition density and the static transition probabilities of the
  individual bits of the input port from the least significant bit to the most significant bit of
  the spatial filter due to the binary pixel values.
- The variation in the orientation of the spatial frequencies also has a significant impact on
  the energy efficiency of the spatial filter.
- Different types of spatial filters consume different amounts of energy; however, they
  follow the same model and the difference in energy consumption is constant based on the
  filter used.
- At lower sizes of images such as 16x16, 32x32 and 64x64 consume more dynamic power
  than the larger sizes. This was found to be largely due to the power consumed by the
  primary inputs and outputs (IO). As the image size is increased the IO power is decreasing
  due to the number of IO switching in a unit time is more for the smaller image than it is
  for the larger image. Energy efficiency increases for smaller images however decreases
  for larger images as the image size and spatial frequency increases.

Accordingly the results should serve to motivate insights and further research in pursuit of the
optimisation of the computational energy efficiency of hardware-accelerated image processing
algorithms.

Discussion and Future work

Hadizadeh et al [15] proposed a method for producing energy-efficient images for energy-
adaptive OLED displays while preserving the perceptual quality of the original images. Similarly,
the findings presented in this paper should motivate the consideration of the attributes of images
that influence the energy-efficiency in their processing and accordingly the attempts to optimize the
trade-off in design, messaging and rendered perceptual quality objectives of such images so as to
enhance the energy efficiency of the hardware accelerated spatial filter used to process the images.
For example, images where the content in the image has large vertically (90 degrees) orientated
structures, could be rotated to 0 degrees so that the vertical structures are orientated horizontally
and then processed through a hardware accelerated digital spatial filter that scans the image in a

row-major order which could potentially result in fewer switching and energy savings than otherwise. Furthermore, spatial frequencies of an image could be reduced, without affecting its ultimately rendered perceptual quality, prior to processing in the spatial filter which could potentially result into energy savings.

Further research could progress this interesting area of research, beyond the scope of this paper, to evaluate the energy-efficiency in processing additional images including other types of synthetic and natural images and/or device/algorithm/configuration/ workflow variations; essentially exploring the computational energy-efficiency correlates of image processing, for example:

- Varying image sizes with different aspect ratios to a square image.
- Using natural images and varying the content in them.
- Varying image content and investigating the impact on the other two major dependent variables, namely, the area and the performance of the hardware accelerated spatial filters.
- Exploring other types of spatial filter architectures, different than the commonly used architecture presented in this paper.
- Varying the hardware implementation platforms such as other FPGA devices and ASIC implementations.
- Varying the content present in a colour image on the energy efficiency of a software or hardware accelerated image processing algorithm.
- Varying configurations of algorithms and workflows involving GPUs, an embedded microprocessor like ARM and traditional single core and multi-core CPUs.

References

[1]   Gene M Amdahl. 1967. Validity of the single processor approach to achieving large scale computing capabilities. In *Proceedings of the April 18-20, 1967, spring joint computer conference*, 483–485.

[2]   Manel Ammar, Mouna Baklouti, and Mohamed Abid. 2016. The Performance-Energy Tradeoff in Embedded Systems Design : A Survey of Existing Design Space Exploration Tools and Trends. *Int. J. Comput. Sci. Inf. Secur.* 14, 5 (2016), 381–391.

[3]   D S August. 2015. *Virtex-6 Family Overview Summary of Virtex-6 FPGA Features*. Xilinx Corporation. Retrieved April 22, 2017 from https://www.xilinx.com/support/documentation-navigation/silicon-devices/fpga/virtex-6.html?resultsTablePreSelect=documenttype:SeeAll#documentation

[4]   Davide Bellizia, Simone Bongiovanni, Pietro Monsurro, Giuseppe Scotti, and Alessandro Trifiletti. 2016. Univariate Power Analysis Attacks Exploiting Static Dissipation of Nanometer CMOS VLSI Circuits for Cryptographic Applications. *IEEE Trans. Emerg. Top. Comput.* (2016), 1–1. DOI:https://doi.org/10.1109/TETC.2016.2563322

[5]   J S Brunhaver. 2015. Design and optimization of a stencil engine. Department of Electrical Engineering, Stanford University.

[6]   Wayne Burleson, Russell Tessier, Dennis Goeckel, Sriram Swaminathan, Prashant Jain, Jeongseon Euh, Subramanian Venkatraman, and Vidhya Thyagarajan. 2001. Dynamically parameterized algorithms and architectures to exploit signal variations for improved performance and reduced power. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP '01). 2001 IEEE International Conference on*, 901–904.

[7]   Anantha P Chandrakasan and Robert W Brodersen. 1995. Minimizing power consumption in digital CMOS circuits. *Proc. IEEE* 83, 4 (1995), 498–523.

[8]   Robert H Dennard, Fritz H Gaensslen, Hwa-Nien Yu, V Leo Rideout, Ernest Bassous, and Andre R Leblanc. 1999. Design of ion-implanted MOSFET's with very small physical dimensions. *Proc. IEEE* 87, 4 (1999), 668–678.

[9]   Geoff Dougherty. 2009. *Digital Image Processing for Medical Applications*. Cambridge University Press. Retrieved from www.cambridge.org/9780521860857

[10]   J Eklund, Christer Svensson, and Anders Astrom. 1995. Near-sensor image processing, a VLSI realization. In *ASIC Conference and Exhibit, 1995., Proceedings of the Eighth Annual IEEE International*, 83–86.

[11]   Hadi Esmaeilzadeh, Emily Blem, Renee St Amant, Karthikeyan Sankaralingam, and Doug Burger. 2011. Dark silicon and the end of multicore scaling. In *ACM SIGARCH Computer Architecture News*, 365–376.

[12]   L. Fanucci and S. Saponara. 2002. Data driven VLSI computation for low power DCT-based video coding. *Proc. IEEE Int. Conf. Electron. Circuits, Syst.* 2, 1 (2002), 541–544. DOI:https://doi.org/10.1109/ICECS.2002.1046221

[13]   Dennis. Greveler, Ulrich; Glösekötterz, Peter; Justusy, Benjamin; Loehr. 2012. Multimedia content identification through smart meter power usage profiles. In *Proceedings of the International Conference on Information and Knowledge   Engineering   (IKE)*. Retrieved   from http://www.nds.rub.de/media/nds/veroeffentlichungen/2012/07/24/ike2012.pdf

[14]   Agrim Gupta and Viktor K Prasanna. 2013. Energy Efficient Image Convolution on FPGA. In *Viterbi India 2013 Program*. Retrieved from http://web.stanford.edu/~agrim/pdfs/fpga.pdf

[15]   H Hadizadeh. 2017. Energy-Efficient Images. *IEEE Trans. Image Process.* 26, 6 (2017), 2882–2891. DOI:https://doi.org/10.1109/TIP.2017.2690523

[16] Yu Haiqian and Miriam Leeser. 2006. Automatic sliding window operation optimization for FPGA-based computing boards. *Proc. - 14th Annu. IEEE Symp. Field-Programmable Cust. Comput. Mach. FCCM 2006* (2006), 76–85. DOI:https://doi.org/10.1109/FCCM.2006.29

[17] Matthew Halpern, Yuhao Zhu, and Vijay Janapa Reddi. 2016. Mobile cpu's rise to power: Quantifying the impact of generational mobile cpu design trends on performance, energy, and user satisfaction. In *High Performance Computer Architecture (HPCA), 2016 IEEE International Symposium on*, 64–76.

[18] Hugo Hedberg. 2008. Image processing architectures for binary morphology and labeling. Lund University.

[19] Prashant Jain, Andrew Laffely, Wayne Burleson, Russell Tessier, and Dennis Goeckel. 2004. Dynamically parameterized algorithms and architectures to exploit signal variations. *J. VLSI Signal Process.* 36, 1 (2004), 27–40.

[20] Jonathan Koomey, Stephen Berard, Marla Sanchez, and Henry Wong. 2011. Implications of historical trends in the electrical efficiency of computing. *IEEE Ann. Hist. Comput.* 33, 3 (2011), 46–54.

[21] Steven Lehar. 2014. An Intuitive Explanation of Fourier Theory. 2–9. Retrieved July 10, 2017 from http://cns-alumni.bu.edu/~slehar/fourier/fourier.html

[22] Chung Lian Jr, Shao-Yi Chien, Chia-ping Lin, Po-Chih Tseng, and Liang-Gee Chen. 2007. Power-aware multimedia: concepts and design perspectives. *IEEE Circuits Syst. Mag.* 7, 2 (2007), 26–34.

[23] Brigitte Marco Tiemann, Atta Badii, Matthias Kalverkamp, Sauro Vinci, Florian Bonacina Trousse, Caroline Tiffon, Xavier Augros, Guillaume Pilot, and et al. Yves Lechevallier. 2013. *Report on IOT Living Labs Continuous Exploration and Evaluation (final)*. FP7 Seventh Framework Program, EU Research.

[24] Sparsh Mittal, Saket Gupta, and S Dasgupta. 2008. System generator: The state-of-art FPGA design tool for dsp applications. In *Third International Innovative Conference On Embedded Systems, Mobile Communication And Computing (ICEMC2 2008)*, 187–190.

[25] Gordon E Moore. 2006. Cramming more components onto integrated circuits, Reprinted from Electronics, volume 38, number 8, April 19, 1965, pp. 114 ff. *IEEE Solid-State Circuits Soc. Newsl.* 20, 3 (2006), 33–35.

[26] Wolfgang Nebel and Jean Mermet. 1997. *Low power design in deep submicron electronics* (1st ed.). Springer US. DOI:https://doi.org/10.1007/978-1-4615-5685-5

[27] Stephen E Palmer. 1999. *Vision science: Photons to phenomenology*. The MIT Press.

[28] Kara K W Poon, Steven J E Wilton, and Andy Yan. 2005. A detailed power model for field-programmable gate arrays. *ACM Trans. Des. Autom. Electron. Syst.* 10, 2 (2005), 279–302.

[29] A. Walker and E. Wolfart. R. Fisher, S. Perkins. 2003. Fourier Transform. *HIPR2*. Retrieved July 10, 2017 from http://homepages.inf.ed.ac.uk/rbf/HIPR2/fourier.htm

[30] A. Walker and E. Wolfart. R. Fisher, S. Perkins. 2003. Contrast Stretching. *HIPR2*. Retrieved July 26, 2017 from http://homepages.inf.ed.ac.uk/rbf/HIPR2/stretch.htm

[31] Jan M. Rabaey, Anantha Chandrakasan, and Borivoje Nikolic. 2003. *Digital Integrated Circuits* (2nd ed.). Prentice Hall Upper Saddle River.

[32] John Sall. 1989. JMP for Statistics. Retrieved August 4, 2017 from https://www.jmp.com/en_gb/software.html

[33] John Sall, Ann Lehman, Mia L Stephens, and Lee Creighton. 2012. *JMP Start Statistics: A Guide to Statistics and Data Analysis using JMP*. SAS Institute.

[34] Abul Sarwar. 1997. *CMOS Power Consumption and Cpd Calculation*. Texas Instruments.

[35] Saktiswarup Satapathy. 2016. Data Path Implementation for a Spatially Programmable Architecture Customized for Image Processing Applications. Arizona State University.

[36] Onur Ulusel, Kumud Nepal, R Bahar, and Sherief Reda. 2014. Fast design exploration for performance, power and accuracy tradeoffs in fpga-based accelerators. *ACM Trans. Reconfigurable Technol. Syst.* 7, 1 (2014), 4.

[37] Gerald Westheimer. 2001. The Fourier theory of vision. *Perception* 30, 5 (2001), 531–541. DOI:https://doi.org/10.1068/p3193

[38] Xilinx. 2013. Xilinx System Generator. *2013*. Retrieved from http://www.xilinx.com/tools/sysgen.htm

[39] Xilinx. 2013. Xilinx Virtex-6 FPGA ML605 Evaluation Kit. *2013*. Retrieved from http://www.xilinx.com/products/boards-and-kits/EK-V6-ML605-G.htm

[40] Xilinx. 2014. Xilinx Power Estimator User Guide (v2014.2). 440, (2014), 1–109. Retrieved from http://www.xilinx.com/support/documentation/sw_manuals/xilinx2015_4/ug440-xilinx-power-estimator.pdf

[41] Xilinx Inc. 2013. AR# 36742: 12.x XPA - What are the signal and toggle rates? Retrieved August 4, 2017 from https://www.xilinx.com/support/answers/36742.html

[42] Ian T Young, Jan J Gerbrands, and Lucas J Van Vliet. 1998. *Fundamentals of Image Processing*. Delft University of Technology, Delft.