International Journal of Image Processing and Vision Science

Volume 2 | Issue 1 Article 7

October 2013

PERFORMING MULTIPLICATIONS IN IMAGE FILTERING PROCESS USING VEDIC MATHEMATICS

PRANOTI Y. PANCHBHAI
Department of Computer Science and Engineering, Shri Ramdeobaba College of Engg. & Management, Nagpur, India., pranoti@gmail.com

DR. NILESHSINGH V. THAKUR
Department of Computer Science and Engineering, Prof . Ram Meghe College of Engg. & Management, Badnera-Amravati, India, nilesh.singh@gmail.com

Recommended Citation
PANCHBHAI, PRANOTI Y. and THAKUR, DR. NILESHSINGH V. (2013) "PERFORMING MULTIPLICATIONS IN IMAGE FILTERING PROCESS USING VEDIC MATHEMATICS," *International Journal of Image Processing and Vision Science*: Vol. 2 : Iss. 1 , Article 7.
DOI: 10.47893/IJIPVS.2013.1065
Available at: https://www.interscience.in/ijipvs/vol2/iss1/7

PERFORMING MULTIPLICATIONS IN IMAGE FILTERING PROCESS USING VEDIC MATHEMATICS

PRANOTI Y. PANCHBHAI[1], DR. NILESHSINGH V. THAKUR[2]

[1]Department of Computer Science and Engineering, Shri Ramdeobaba College of Engg. & Management, Nagpur, India.
[2]Department of Computer Science and Engineering, Prof . Ram Meghe College of Engg. & Management, Badnera-Amravati, India.

Abstract- Image filtering is a very important step in image processing. Filtering involves lots of multiplications which consumes time. Time required increases with the increase in the number of pixels. This paper proposes an approach for image filtering using Vedic Mathematic which performs faster multiplication compared to the conventional algorithms namely Booth 2, Booth 3 and Array Multiplication Algorithm thus reducing the time required for filtering of images. Experimentation is done using C language. Time required by the algorithms for filtering are then compared using the experimental results.

Keywords*- image filtering, vedic mathematics, multiplication algorithms*

I. INTRODUCTION

Image filtering is defined by a neighborhood and an operation that is performed on the pixels inside the neighborhood. It transforms the pixel intensity value to reveal certain image characteristics [1]. Often, images are corrupted due to noise, poor contrast, illumination, etc. Image filtering corrects these changes and hence it is very important in image processing. Basically, filtering involves lots of iterative multiplications. It is a very complex arithmetic operation. Hence, filtering will take a lot of time especially for large images made up of huge number pixels. Therefore, it is vital to use faster multiplication algorithms, thus reducing the time required for filtering.

There exist various multiplication algorithms namely Booth 2 and Booth 3 multiplication algorithm [2] and Array multiplication [3]. Booth 2 and Booth 3 are binary multiplication algorithms which in general, involves the grouping of the bits of the multiplier and selecting the partial products from the selection table which are then added to give the final product. Both differ in the logic which selects the partial product from the selection table.

Array multiplication is also a binary multiplication algorithm for multiplying two binary numbers. It multiplies all the bits at once. The problem with Booth 2 and Booth 3 algorithms is that it involves creation of selection table and partial products. Partial products lead to more amount of hardware causing latency to generate the final product. Another drawback is that the complex partial product selection logic causes extra cost and delay [2]. Array multiplication is impractical for large numbers and is less economical [3], [4]. The Vedic multiplication algorithm based on the vertically and crosswise algorithm of Ancient Indian Vedic Mathematics. It involves parallel calculation of partial products and their addition [5]. This algorithm is used for binary multiplication. Its advantage is that as the number of bits increase, the delay increase slowly compared to other algorithms.

This paper presents an approach for image filtering based on Vedic Mathematics. Main issue of multiplication in image filtering process is handled using Vedic Mathematics. The obtained result justifies how Vedic Mathematics improves the total time required for the filtering process. The results are compared with basic multiplication algorithms. This paper is organized as follows. Section 2 summarizes the basics of image filtering. Section 3 discusses about the various multiplication algorithms. Section 4 describes the proposed approach for image filtering using Vedic Mathematics. Section 5 describes the experimental setup and results obtained as well as the timing comparisons. Section 6 presents the conclusion followed by references.

II. BASICS OF IMAGE FILTERING

Many image enhancement techniques are based on spatial operations performed on local neighborhoods of input pixels. The image is usually convolved with a finite impulse response filter called spatial mask. The use of spatial masks on a digital image is called spatial filtering. Typically, the neighborhood is rectangular and its size is much smaller than that of f(x, y) e.g., 3 × 3 or 5 × 5 Assume the origin of mask is the centre of the mask. For a 3 × 3 Mask, filtering is

$$g(x, y) = \sum_{s=-1}^{1}\sum_{t=-1}^{1} w(s,t)\, f(x+s, y+t)$$

Where w(s, t) is the mask, f(x, y) is the input image and function g(x, y) is the output filtered image. A filtered image is generated as the centre of the mask moves to every pixel in the input image. There are low pass filters as well as high pass filters.

In this paper, the filter mask used for experimentation purpose is smoothing linear filter mask (averaging filter) which is a low pass filter used for blurring and noise reduction. The smoothing linear filter 3 × 3 mask used is:

$$\frac{1}{16}\times\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix}$$

III. MULTIPLICATION ALGORITHMS

This section discusses the basic multiplication algorithms which are used in general.

A. Booth 2 and Booth 3 Algorithm

Instead of simple multiplication, Booth 2 groups the bits of the multiplier into pairs of 3. Each group is decoded to select a single partial product as per the selection table.

The selection set is {0, M, 2M} where M is the multiplicand. All the multiples from this set can be obtained by simple shifting and complementing.

The number of partial products are reduced from 16 (General Booth's algorithm) to 9. The partial products are then added to obtain the final product. Smaller number of partial products will allow the partial product summation to be faster and use less hardware.

Booth 3 is similar to Booth 2 algorithm. But here, the bits of the multiplier are grouped into pairs of 4 reducing the number of partial products to 6. The selection set is {0, M, 2M, 3M, 4M}.

Booth 3 algorithm is most efficient in power and area but is slower due to the need for expensive carry propagate addition when computing some hard multiples. The Booth 2 algorithm is fastest, but is also quite power and area hungry.

B. Array Multiplication Algorithm

Array multiplication is much like simple multiplication of two numbers. Here, the binary numbers are considered as two arrays and multiplication is implemented in a purely combinational two-dimensional logic array forming all the product bits at once.

It is quite simple and easily scalable but requires many gates resulting in large chip area.

C. Vedic Mathematics Multiplication Algorithm

Multiplication in Vedic Mathematics is done using vertically and crosswise algorithm. Vedic Multiplication for 3 bit is explained below. Consider two 3 bit numbers Multiplicand M 101(5) and Multiplier N 110(6). Final product is stored in variable Result. For implementation carry is used which stores the carry from previous step addition.

Step 1: If 0^{th} bit of Multiplier i.e. N(0) is 1 then 0^{th} bit of Result i.e. Result(0) is 0^{th} bit of Multiplicand i.e. M(0) otherwise Result(0) is 0. Here, Carry is 0.

Step 2: If N(0) is 1 then add M(1) to Result(1). Then, if N(1) is 1 then add M(0) to Result(1). Add Result(1) with the previous carry and generate new carry. Here, carry is 0.

Step 3: If N(0) is 1 then add M(2) to Result(2). Then if N(1) is 1 then add M(1) to Result(2). Then if N(2) is 1 then add M(0) to Result(2). Add Result(2) with previous carry and generate new carry. Here, carry is 0.

Step 4: If N(1) is 1 then add M(2) to Result(3). Then if N(2) is 1 then add M(1) to Result(3). Add Result(3) with the previous carry and generate new carry. Here, carry is 0.

Step 5: If N(2) is 1 then add M(2) to Result(4) else Result(4) is 0. Add Result(4) with the previous carry and generate new carry. Here, carry is 0.

Step 6: Result(5) is the carry lastly generated.

All performed computations are shown in Fig. 1 (a) and Fig. 1 (b).

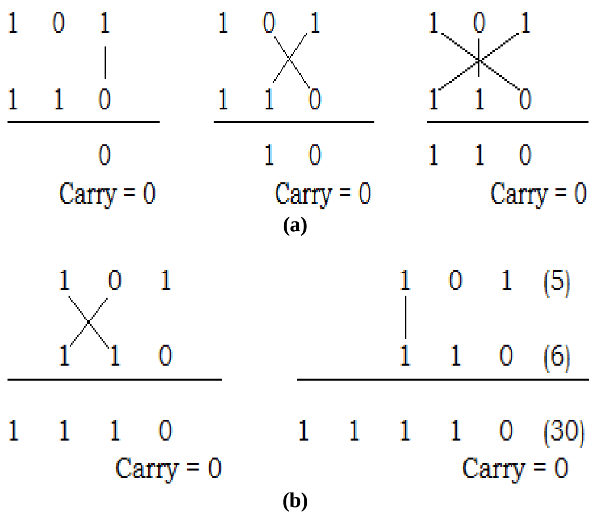

Figure 1. Performed Computations for 5 × 6

Thus, the variable Result of 6 bits gives multiplication. In the above explained example, binary 5(101) is multiplied by binary 6(110) to generate result 30(11110). This method can be extended to n bit Multiplication.

IV. PROPOSED APPROACH

Proposed approach is based on Vedic Mathematics which is used for multiplication to be performed in image filtering process for the given image. This approach is depicted in the Fig. 2.

In proposed approach, the image is first acquired. For boundary values, the image is zero padded. The image is then multiplied with the filter mask. Thus, mask changes the intensity value of each pixel.

The main operation of multiplication is done through the multiplication algorithms of Booth 2, Booth 3, Array Multiplication and Vedic Mathematics as explained in section 3. The time required for filtering by each of the algorithm is measured. The timings are then compared.

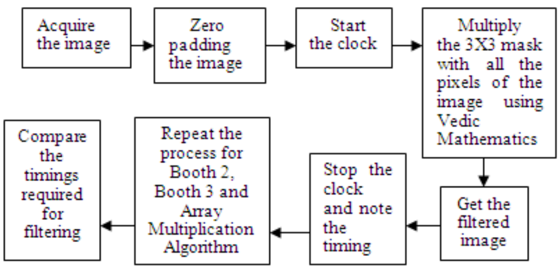

Figure 2. Flowchart for the proposed approach

Pseudo code for image filtering using Vedic Mathematics is shown in Fig. 3.

```
Program Vedicmulp(unsigned int a1,unsigned int b1)
    // program gets two integers of 16 bits in a1 and
b1 to      be multiplied using Vedic Mathematics from
the main filtering program
unsigned char carry=0,a[16]={0},b[16]={0},c[32]={0};
    //a n b has number c has final product of 32 bits
    //logic to convert numbers from base 10 to base 2
 // each bit of multiplier i.e. b is checked and
 additions are done according to Vedic Mathematics to
 get the final product
     if(b[0]==1)
            c[0]=a[0];c[1]=c[1]+a[1];c[2]=c[2]+a[2];
            c[3]=c[3]+a[3];c[4]=c[4]+a[4];
            c[5]=c[5]+a[5];c[6]=c[6]+a[6];
            c[7]=c[7]+a[7];c[8]=c[8]+a[8];
            c[9]=c[9]+a[9];c[10]=c[10]+a[10];
            c[11]=c[11]+a[11];c[12]=c[12]+a[12];
            c[13]=c[13]+a[13];c[14]=c[14]+a[14];
            c[15]=c[15]+a[15];
    end if
 // similarly all the other 15 bits of multiplier i.e. b
is checked
  //code for converting number from base 2 to base 10
and returning the product to the main filtering program
end.
```

Figure 3. Pseudo code for Image Filtering using Vedic Mathematics

V. EXPERIMENTAL SETUP AND RESULTS

This section discusses the experimental setup and obtained results. The experiment is carried out on a 10 × 10 image which is operated n number of times to study the time comparisons. The implementation is carried on Pentium(R) Dual-Core CPU T4200 @ 2.00GHz 1.20 GHz, 2.93 GB of RAM system using C language.

Screenshots of the experiments are shown Fig. 4 and Fig. 5. Vedic mathematics results are summarized in Table I. These obtained results are compared with Booth 2, Booth 3 and Array multiplication algorithms. These algorithms are also implemented for the same 10 × 10 image using the same approach as Vedic Mathematics. Comparison of results is shown in Table II.

It can be observed from Table I and Table II that Vedic Mathematics requires less time for filtering the image compared to other conventional algorithms. For example, for an image of 70,000 pixels (i.e. 10 × 10 image for n=700 times) Vedic Mathematics required 1.3187 seconds in comparison with 1.4286, 1.5934, and 1.7032 of Booth 2, Array Multiplication and Booth 3 algorithms respectively. Likewise, time requirement is less for the other images of different number of pixels.

Comparison of various multiplication algorithms is shown in Fig. 6. This figure shows the time in seconds required by Vedic Mathematics, Booth 2, Booth 3, Array Multiplication Algorithms for image filtering for various number of pixels.

Table III shows that Vedic Mathematics is faster by 7.69%, 17.24%, 22.58% than Booth 2, Array and Booth 3 respectively. Graphical representation of same is shown in Fig. 7.

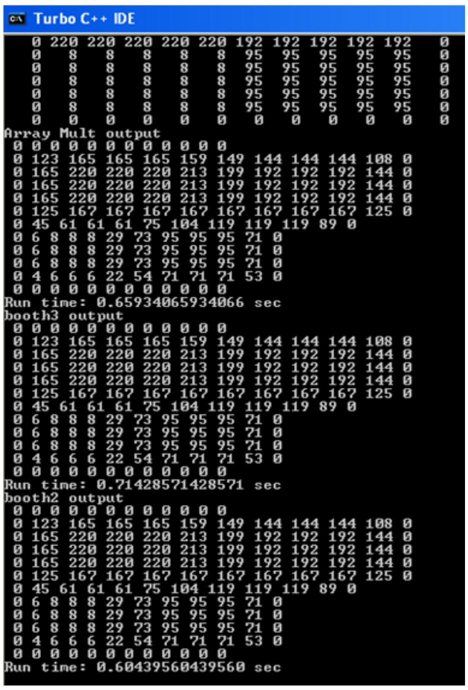

Figure 4. Screenshot of the experiment in C language

Figure 5. Screenshot of the experiment in C language

TABLE I

TIME IN SECONDS FOR FILTERING USING VEDIC MATHEMATICS

Number of Pixels (10 × 10 image n times)	70,000 (n=700)	100,000 (n=1000)	150,000 (n=1500)	200,000 (n=2000)
Vedic Approach	1.3187	1.9231	2.8571	3.4065

TABLE III

TIME IN SECONDS FOR FILTERING USING BOOTH 2, ARRAY MULTIPLICATION, BOOTH 3 ALGORITHMS

Number of Pixels (10 × 10 image n times)	70,000 (n=700)	100,000 (n=1000)	150,000 (n=1500)	200,000 (n=2000)
Booth 2	1.4286	2.0879	3.0769	3.6813
Array	1.5934	2.2527	3.4066	4.0109
Booth 3	1.7032	2.7425	3.6813	4.3956

TABLE IIIII

PERCENT BY WHICH VEDIC MATHEMATICS IS FASTER THAN BOOTH 2, ARRAY MULTIPLICATION, BOOTH 3 ALGORITHMS

Number of Pixels (10 × 10 image n times)	70,000 (n=700)	100,000 (n=1000)	150,000 (n=1500)	200,000 (n=2000)
Booth 2	7.69	7.89	7.14	7.46
Array	17.24	14.63	16.13	15.07
Booth 3	22.58	22.22	22.39	22.50

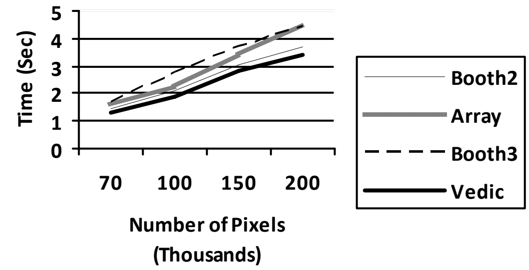

Figure 6. Time in seconds required by Vedic Mathematics, Booth 2, Booth 3, Array Multiplication Algorithms for image filtering for various numbers of pixels

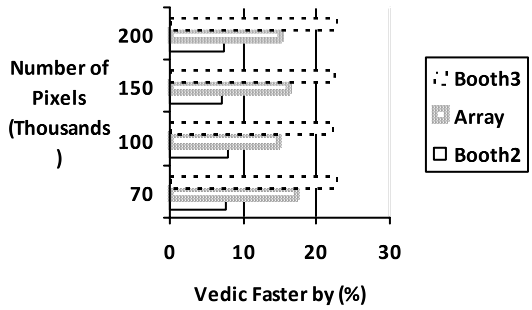

Figure 7. Percent by which Vedic Mathematics is faster than Booth 3, Array and Booth 2 algorithms for various number of pixels

VI. CONCLUSION

Vedic Mathematics performs better in comparison with other multiplication algorithms. This helps in performing faster image filtering using Vedic Mathematics. It will help in developing faster architectures.

ACKNOWLEDGMENT

This paper is based on the work carried out in final year B. E. project at Shri Ramdeobaba College of Engineering and Management, Nagpur, India. The first author was a student of 2012 batch of Department of Computer Science and Engineering. The second author was guide of the project when he was working with Shri Ramdeobaba College of Engineering and Management, Nagpur, India. The authors are very much thankful to all the faculty of the Department of Computer Science and Engineering for their constant appreciation during the tenure of the final year project.

REFERENCES

[1] R.C. Gonzalez and R.E. Woods, Digital Image Processing, 2nd ed., Pearson Education, Inc., pp. 167-178, 2002.

[2] Gary Berwick, “Fast Multiplication algorithms and implementation”, dissertation, dept of electrical, Stanford University, pp. 13-21, 1994.

[3] V.C. Hamacher, Z.G. Vranesic, S.G. Zaky, Computer Organisation, 3rd ed., The Mcgraw Hill Company, pp. 282-291, 1996.

[4] Morris Mano, Computer System Architecture, 3rd ed., PHI., pp. 346-347, 1993.

[5] A.P. Nicholas, K. R. Williams, J. Pickles, Vertically and Crosswise: Applications of the Vedic Mathematics Sutra, Motilal Banarsidass Publication, pp. 1-9, 1992.

◈◈◈