USC-SIPI REPORT # 129

# On the Hardware Requirement for 2-D Image Convolution in Electro-Optical Systems

*Mary Eshaghian, D. K. Panda*
*and*
*V. K. Prasanna Kumar*

August 12, 1988

# On the Hardware Requirement for 2-D Image Convolution in Electro-Optical Systems[1]

**Mary Eshaghian, D. K. Panda and V. K. Prasanna Kumar**

**University of Southern California**
**Department of Electrical Engineering-Systems**
**Los Angeles, CA 90089-0781, USA**
August 12, 1988
(213)-743-5236

**Abstract**

In this paper, we study the hardware requirement for 2-D image convolution on an electro-optical model. We identify the basic components of an Optical model of parallel computation using optical interconnects and their relationship with well known VLSI grid model from a computational perspective. We show a lower bound of $\Omega(nw)$ on the volume requirement of any electro-optical chip for 2-D image convolution using a $w \times w$ window and a $n \times n$ image. This bound also applies to the area requirement of a VLSI chip. Designs available in the literature for computing the 2-D image convolution are compared with respect to their hardware requirement.

[1]This research was supported in part by the National Science Foundation under grant IRI-8710836.

# 1 Introduction

The computational throughput of a parallel architecture is largely dependent on it's communication bandwidth. The communication bandwidth available in the present day parallel systems is reaching saturation due to the inherent limitations of transmitting electronic signals [1]. To overcome this problem, researchers have considered contention-free optical beams as an efficient means of interconnection [2,3,4,5]. Unlike the electronic signal, the optical interconnection offers dual advantages of larger bandwidth and fan-out [6,7]. With the availability of such interconnection, the field of Optical computing is being diversified from the development of analog optical processors [8,9,10] to digital optical and electro-optical computers [11,12,13,14,15]. Such electro-optical systems, capable of exploiting the speed and parallelism of optical systems together with the programmability and accuracy of electronic computers, promise tremendous computational power.

In this paper, we consider an electro-optical system from a computational perspective and study some inherent limitations of such a system in parallel computation. As an example, we study the electro-optical resource requirements for solving a fundamental, computationally intensive operation such as 2-D image convolution. The 2-D image convolution is extensively used in signal and image processing [16]. A lower bound on the storage(memory) requirement of an electro-optical chip to solve a problem reflects the hardware requirement for fabricating such a system. We present a lower bound of $\Omega(nw)$ on the volume requirement of an electro-optical chip for computing image convolution. Irrespective of the I/O scheme and the order of computation, we show that any image convolution design must satisfy this bound for convolving a $w \times w$ kernel with a $n \times n$ image, as long as the input bits are given to the system once only. Most of the VLSI designs for 2-D image convolution [17,18,19,20] use the input image only once. All these designs satisfy this bound.

The rest of the paper is organized as follows. In the next section, we describe an optical model of computation, the relationship between the minimum volume requirement in this model and information transfer. In section 3, we define image convolution and input formats. We also derive a lower bound on the information transfer for image convolution under several input formats used in practice. In the last section, we compare the memory requirements of the known VLSI designs for image convolution.

# 2 An Optical Model and Information Transfer

In this section, we define an Optical model of computation which is an abstraction of currently implementable optical and electro-optical computers [21,14]. Similar to the VLSI

[2] A function $f(n)$ is said to be $\Omega(g(n))$ if there exist positive constants $c$ and $n_0$ such that for all $n > n_0$, $f(n) \geq c \cdot g(n)$. A function $f(n)$ is said to be $O(g(n))$ if there exist positive constants $c$ and $n_0$ such that $f(n) \leq c \cdot g(n)$ for all $n \geq n_0$.

model of computation [22], this model can be used to understand the limits on computational efficiency in using optical technology. We show that minimum volume requirement of an optical model of computation is same as the minimum VLSI area in the VLSI model. Using information transfer argument, we also show a methodology to determine the minimum volume requirement of an electro-optical system for solving a problem.

## 2.1 An Optical Model

An optical model [14] is shown in figure 1. More formally this model is defined as follows (see [21] for a similar model for volume-time tradeoff):

**Definition 1** ***An optical model of computation represents a network of processors each associated with a deflecting unit and a receiving unit capable of establishing direct optical connection to another processor or a set of processors.***

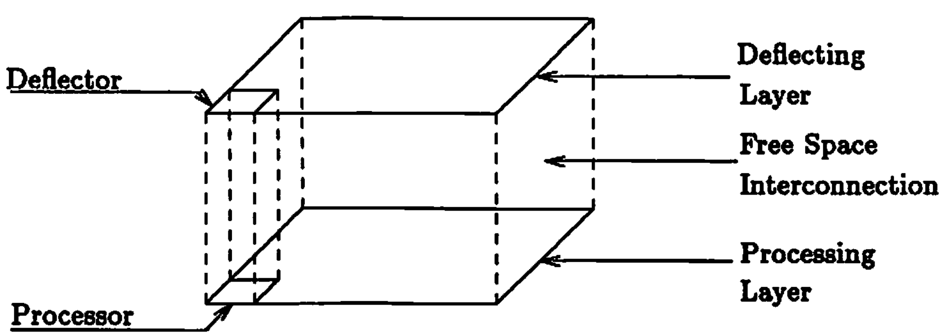



Figure 1: The optical model of computation

We make the following assumptions to capture the real life optical designs.

1. The processing layer consists of processors and memory elements. In one unit of time, a processor can compute a simple arithmetic/logic operation and a deflector can redirect an incident beam.
2. The intercommunication is done through free-space optical beams. An optical beam carries a constant amount of information in one unit of time, independent of the distance to be covered.
3. I/O is performed at I/O pads. Each I/O pad occupies one unit of volume. Similarly, one bit of memory contributes to at least one unit of volume. The volume occupied by processors, deflecting elements, memory, and I/O pads together determine the total volume of the system.
4. The time, T for computation is the time between the arrival of the first input to the departure of the last output.

5. The input and output are performed according to a pre-determined sequence of time instants at pre-specified locations which depends entirely on the circuit design, not on the data [22].

6. Each input bit is read by the chip exactly once.

## 2.2 Optical Volume, VLSI Area and 1-way Information Transfer

The minimum VLSI area requirement for computing a problem is related to the *lower bound* on the *1-way information transfer* [23,24]. In this section, we briefly discuss the 1-way information transfer for computing *single output function* and then extend the argument to *multiple output functions*. We also relate this 1-way information transfer to the Optical volume requirement of an electro-optical system to solve a problem.

The following abstract setting has been shown to be useful in estimating the minimum VLSI chip area [24,25]. Two sets of processors $P1$ and $P2$ each receive $\frac{n}{2}$ bits of an $n$ input function $f$ to be computed. The input partition can be denoted by $(\pi_1^i, \pi_2^i)$ where $\pi_1^i(\pi_2^i)$ are the inputs known to $P1(P2)$ and $\pi_1^i \cap \pi_2^i = \phi$. Also $\left|\pi_1^i\right| = \left|\pi_2^i\right| = \frac{n}{2}$. The rectangle corresponding to this partition is defined as $M \times N$ where $M(N)$ = set of all values known to $P1(P2)$. It is clear that $|M| = |N| = 2^{\frac{n}{2}}$. In the 1-way information transfer model, $P2$ computes $f$ and outputs the result. Hence, given an input partition, some information based on the input bits of $\pi_1^i$ are transferred to $P2$ in order to complete the computation. The minimum *information transfer* from $P1$ to $P2$ to compute $f$ over all possible input partitions is denoted by $I_1(f)$ and is defined as follows:

$$I_1(f) = \underset{input\ partition}{Min} \left\{ \begin{array}{c} worst\ case\ information\ transfer \\ from\ P1\ to\ P2 \end{array} \right\}$$

Two rows $i_1, i_2 \in M$ in the *computational rectangle* are said to be *distinct* if $\exists$ a $j \in N$ such that $f(i_1, j) \neq f(i_2, j)$. If $d(f)$ is the minimum number of such *distinct* rows over all possible input partitions, then $I_1(f)$ is equal to $\log d(f)$ [26]. The area requirement $A$ of any chip computing $f$ is $\Omega(I_1(f))$ [23,24].

The above 1-way protocol for single output function can be easily extended to multiple output functions by introducing a suitable output partition over the output functions. Let $F = \{f_1, f_2, \ldots, f_l\}$ be the set of output functions. The output partition, similar to the input partition, can be denoted by $(\pi_1^o, \pi_2^o)$. The output partition satisfies the conditions $\pi_1^o \cap \pi_2^o = \phi$ and $\pi_1^o \cup \pi_2^o = \{f_1, f_2, \ldots, f_l\}$. Both processors $P1$ and $P2$ are allowed to compute the output functions belonging to their respective subsets. Before proceeding further, we state some definitions and restate some earlier results.

[3] All logarithms in this paper are to base 2.

Since the 1-way communication link is from $P1$ to $P2$, it is not possible to transfer data from $P2$ to $P1$ to compute a function belonging to $\pi_1^o$. Hence, an output partition requiring transfer of data from $P2$ to $P1$ is not *feasible*. This leads to the following definition of a *feasible* output partition:

**Definition 2** *An output partition is feasible iff on any input partition, all functions in $\pi_1^o$ can be computed at $P1$ using only the input bits of $\pi_1^i$.*

The *1-way information transfer* for multiple output functions can depend on the output partition. Hence, $I_1(F)$, the 1-way complexity of computing a set of output functions is defined as follows:

$$I_1(F) = \begin{array}{c} Min \\ feasible\ output \\ partition \end{array} \left\{ \begin{array}{c} Min \\ input\ partition \end{array} \left\{ \begin{array}{c} worst\ case \\ information\ transfer \\ from\ P1\ to\ P2 \end{array} \right\} \right\}$$

The relationship between $I_1$ and the minimum area requirement $A$ of a chip stated earlier for single output function holds good for multiple output functions too.

**Theorem 1** *The volume $V_o$ of any electro-optical system computing $F$ satisfies $V_o = \Omega(I_1(F))$.*

**Proof:** Consider an electro-optical system as shown in figure 2. $P1$ can be viewed as the electro-optical system and $P2$ as the memory. Consider the state of the system after reading $\frac{n}{2}$ input bits. These bits can be looked upon as bits belonging to $\pi_1^i$. Based on this input, the system would have computed some set of output functions. Denote these functions as $\pi_1^o$. If the volume of the system is $V_o$, then the system would not have memorized more than $V_o$ bits of information. It is easy to design a 1-way protocol with $V_o$ bits of information transfer from $P1$ to $P2$ to compute the rest of the output functions. Hence, the system should have at least $I_1(F)$ memory elements. Thus, the volume of the system must be $\Omega(I_1(F))$. □

The computation of $F$ over an input and output partition can be represented in the form of a *computational parallelepiped* $P$ as shown in figure 3. $M(N)$ are the set of possible values of the input bits known to $P1(P2)$. For a fixed value of input bits, the output functions in set $F$ are represented by a vector of length $l$ in the third dimension of $P$. Given an input partition, the output functions can be divided into the following subsets:

$F_{11}$ - The subset of the output functions which can be computed at

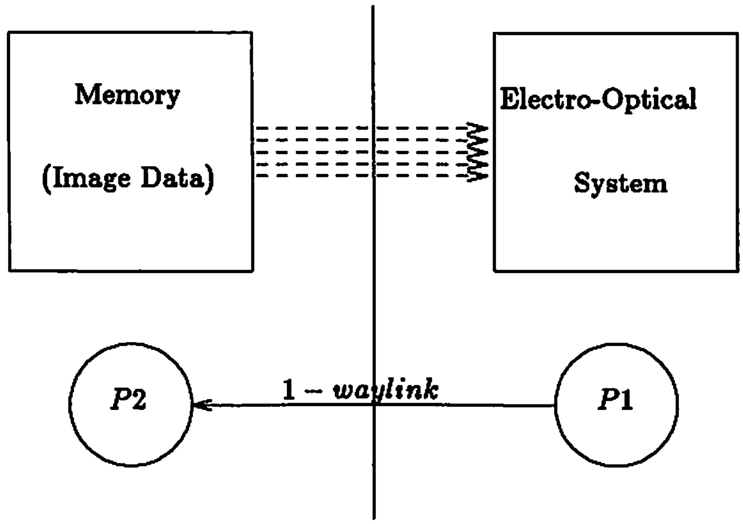



Figure 2: 1-way information transfer

| | | |
|---|---|---|
| | | $P1$ (on every input) using the input bits of $\pi_1^i$ only |
| $F_{22}$ | - | The subset of the output functions which can be computed at $P2$ (on every input) using the input bits of $\pi_2^i$ only |
| $F_{12}$ | - | The rest of the output functions. |

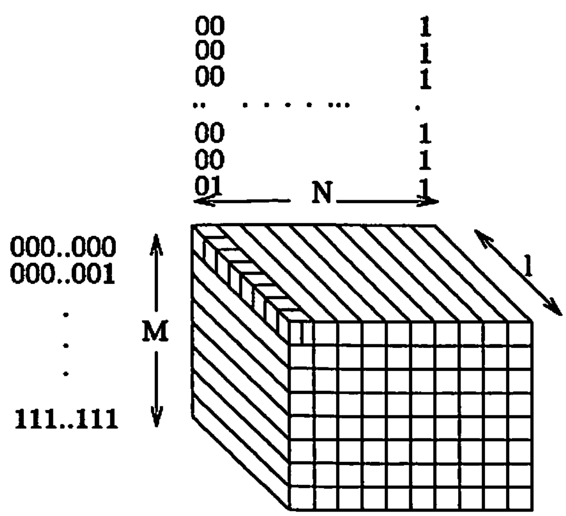



Figure 3: The computational parallelepiped P

The computation of the output functions belonging to the subsets $F_{12}$ and $F_{22}$ require data from $\pi_2^i$. According to the definition of a *feasible* output partition, these functions can not be computed at $P1$ and, hence, to be computed at $P2$. Therefore, for a given input partition $(\pi_1^i, \pi_2^i)$, the output partition is *feasible* iff $F_{12} \subseteq \pi_2^o$ and $F_{22} \subseteq \pi_2^o$.

Since the output functions $F_{11}$ and $F_{22}$ are computed at $P1$ and $P2$ respectively using exclusively the input bits assigned to them, there is no information transfer associated with these computations. The computation of $F_{12}$ only requires information transfer from $P1$ to $P2$. This information transfer $I_1(F)$ is determined from the number of *distinct planes* in $P$.

**Definition 3** *Two planes* $i_1, i_2 \in M$ *in* $P$ *are distinct iff there exists a* $j \in N$ *and a function* $f_k \in F$ *such that* $f_k(i_1, j) \neq f_k(i_2, j)$.

Based on the above definition of *distinct planes*, $I_1(F)$ can be estimated in a similar spirit as in $I_1(f)$ [26]. This leads to:

**Proposition 1** *For a fixed input partition* $(\pi_1^i, \pi_2^i)$ *and a fixed output partition* $(\pi_1^o, \pi_2^o)$ *, the minimum number of bits of information transfer from* $P1$ *to* $P2$ *to compute* $F$ *is* $\log d(F)$, *where* $d(F)$ *is the number of distinct planes in* $P$. *Also, the 1-way complexity* $I_1(F)$ *is equal to* $\log d$, *where* $d$ *is the minimum* $d(F)$ *over all possible input and feasible output partitions.*

# 3 Optical Volume for Computing Image Convolution

In this section, we derive a lower bound on the information transfer, $I_1(F)$, for image convolution under several input formats using the technique of last section. These bounds are translated to lower bounds on optical volume for computing image convolution. We begin with digital image convolution and the input formats.

## 3.1 Digital Image Convolution

Given a $n \times n$ image $I(i,j)$, $1 \leq i,j \leq n$ and a $w \times w$ kernel $W(s,t)$, $1 \leq s,t \leq w$, the 2-D digital image convolution to compute $C(i,j)$, $1 \leq i,j \leq n-w+1$ is given by,

$$C(i,j) = \sum_{t=1}^{w} \sum_{s=1}^{w} I(i+s-1, j+t-1)\, W(s,t).$$

The convolution operation is generally carried out for pixels $(i,j)$, $1 \leq i,j \leq n-w+1$, to avoid the wrap-around of the input image. Since in most cases, $n \gg w$, $O(n^2)$ input data are used in the convolution to generate $O(n^2)$ output data. Overall, $O(n^2w^2)$ computations are involved in the operation.

### 3.2 Image Input Formats

Due to the computationally intensive nature of the convolution operation, most of the designs in the literature perform several computations per input pixel fetch to achieve high throughput and reduced memory bandwidth. Figure 4 shows the computing environment to carry out convolution using an Electro-optical system. The input to the system is a $n \times n$ image and a $w \times w$ kernel. The electro-optical system must be able to compute the image convolution on any input $n \times n$ image and any $w \times w$ kernel. The host is responsible for acquiring the input data and feeding it to the system which contains processing units for computation and memory elements for storing intermediate results. Though, most of the practical designs organize cells to input the image pixels from the host in a rasterscan *i.e.* *scanline* fashion, designs are possible to input data in arbitrary sequence of rows/columns or arbitrary pixel manner.

Different designs are possible depending on whether the image from the host is fed to the system *once only* or *more than once*. The memory requirement for convolution can be traded off with the computation time and the number of times each input pixel is fed to the system. The extreme case is that of serial computation having $O(1)$ memory requirement. But, this demands each input pixels to be fed $O(w^2)$ times resulting $O(n^2w^2)$ computation time. In the present discussion, we restrict our analysis to *once only* constraint which results in high throughput designs, yet requiring reduced storage [17,18,19,20]. Without loss of generality, we make the following assumptions for sake of simplicity in the presentation of ideas.

1. The image size $n$ and the kernel size $w$ are assumed to be multiples of 2 and $n \gg w$.
2. All image pixels and kernel weights are assumed to be 1-bit wide and take binary values. However, the analysis can be easily extended to $r$-bit pixels and $s$-bit weights by scaling up the result by a factor of $rs$.

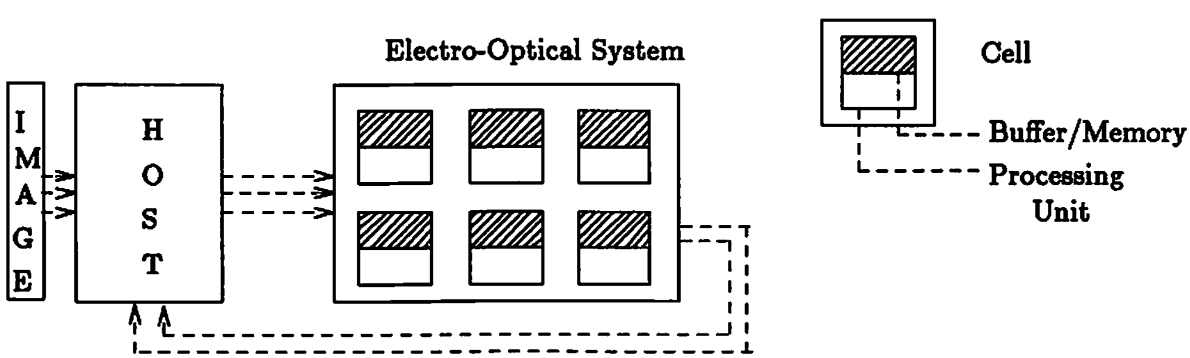



Figure 4: Environment for computing image convolution

It is obvious that $O(n^2)$ memory is sufficient under any input format. As the problem size becomes larger, efficient implementation with fixed resource requirement is desirable by partitioning the problem into smaller sizes [27]. Notice that, the known VLSI designs for image convolution [17,18,19,20] use partitioning techniques and require $\Omega(nw)$ memory.

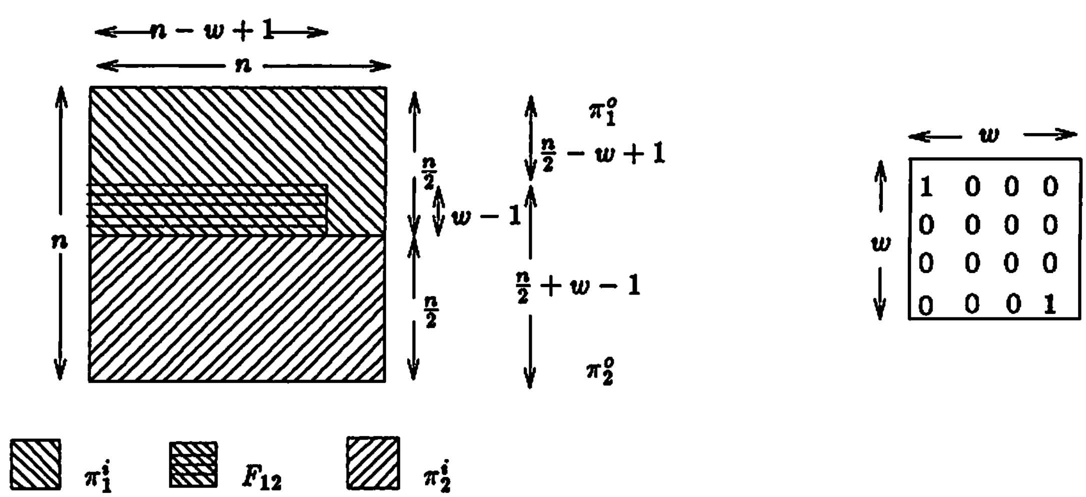
### 3.3 Lower Bounds on Optical Volume

For a $n \times n$ image, the convolution operation produces $O(n^2)$ output. The information transfer for such operation can be analyzed based on the ***multiple output function*** model discussed in the last section. We first consider the *scanline* input format as an illustration of the methodology.

#### 3.3.1 Scanline Input Case

In this case, the input partition corresponds to $P1$ receiving the upper half of the image and $P2$ receiving the lower half as shown in figure 5 (a). Let the kernel $K$ of size $w \times w$ be as shown in figure 5 (b). All kernel weights are zero except for $K(1,1)$ and $K(w,w)$ entries. With the given kernel $K$, the output $C(i,j)$ of the convolution becomes:

$$C(i,j) \;=\; I(i,j) \;+\; I(i+w-1, j+w-1) \quad for \;\; 1 \le i,j \le n-w+1.$$




(a) (b)

Figure 5: (a)The input and output partition for scanline case. (b)A kernel

**Lemma 1** *The volume $V_o$ of any electro-optical design using scanline input format for image convolution satisfies $V_o = \Omega(nw)$.*

**Proof:** The computation of $C(i,j)$, $\frac{n}{2}-w+2 \le i \le \frac{n}{2}; 1 \le j \le n-w+1$, shown as a shaded strip of $(w-1)$ rows in figure 5 (a) requires input bits $I(k, j+w-1)$, $\frac{n}{2}+1 \le k \le \frac{n}{2}+w-1$. But, these input bits belong to $\pi_2^i$. Thus, according to the definitions of $F_{12}$ and *feasible* output partition, these output functions belong to $\pi_2^o$.

Let the parallelepiped P as shown in figure 6 represent the computational parallelepiped for the present case. The $\frac{n^2}{2}$ input bits belonging to $\pi_1^i$ are ordered as $(X_1, X_2)$, where $X_1$ corresponds to the desired $(n-w+1)(w-1)$ bits of the rows $\frac{n}{2}-w+2$ to $\frac{n}{2}$ and $X_2$ corresponds to the rest of the bits.

The number of *distinct planes* in $P$ is at least equal to the number of distinct rows in any vertical plane. Consider the vertical plane corresponding to value of 0 for all input bits in $\pi_2^i$. With the given kernel $K$, the output of the convolution operation for bits in $X_1$ are identical to the respective input values. The $(n-w+1)(w-1)$ bits in $X_1$ can take $2^{(n-w+1)(w-1)}$ distinct values, resulting in $2^{(n-w+1)(w-1)}$ distinct rows in the vertical plane under consideration. This leads to, $d \ge 2^{(n-w+1)(w-1)}$. By proposition 1, $I_1(F) \ge (n-w+1)(w-1)$ or $I_1(F) = \Omega(nw)$. Using theorem 1, the result follows. □

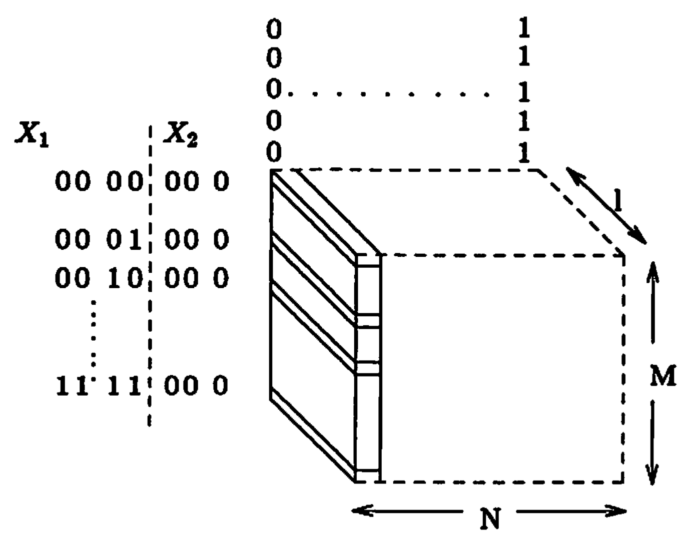



Figure 6: The computational parallelepiped P for scanline case

### 3.3.2 The General Input Case

In this case, we consider designs, where the input is received by the electro-optical system from the host in arbitrary sequence of pixels. We first determine a bound for a special case of this, where the input is restricted to arbitrary row(column) major sequence. Later we derive the bound for the general case by reducing it to the arbitrary row(column) major input format.

Let the $n$ rows of the input image be arbitrarily colored as R(Red) or B(Blue) such that equal number of R and B rows exist. The $\frac{n}{2}$ R(B) rows correspond to $\pi_1^i(\pi_2^i)$. Define $wd(x)$ to be a window covering $w$ consecutive rows starting at row $(x)$. By sliding this window over the $n$ rows, $(n-w+1)$ different windows are possible.

**Claim 1** *For $n \geq w(\frac{w}{2}+2)$, there exists a window $wd(x)$, $1 \leq x \leq n-2w+2$, consisting of $R(B)$ rows with indices $i_1,\ i_2,\ \ldots i_l$, $l \geq \frac{w}{16}$, and an integer $k$, $1 \leq k \leq w-1$, such that the rows with indices $i_j+k$, $1 \leq j \leq l$, are $B(R)$.*

**Proof:** Since there are equal number of $R$ and $B$ rows, it is true that there exists at least one window $wd(x)$, $1 \leq x \leq n-2w+2$, consisting of equal number of $R$ and $B$ rows. Partition this window WD into two equal halves, WD1 consisting of the first $\frac{w}{2}$ rows and WD2 having the rest. Each half contains either $\geq \frac{w}{4}$ $R$ or $\geq \frac{w}{4}$ $B$ rows.

Without loss of generality, let the number of $R$ rows in WD1 be $\geq \frac{w}{4}$. Let the indices of the $B$ rows in WD2 be $j_1, j_2, \ldots j_q$. Note that $q \geq \frac{w}{4}$. Construct a table as shown in figure 7 (a), where $(j_r, i), 1 \leq r \leq q$ is a 1 iff row $(j_r - i)$ in WD is a $R$ row. Now, each column in this rectangle has at least $\frac{w}{4}$ 1's. Since the number of $B$ rows in WD2 is $\geq \frac{w}{4}$, the rectangle contains at least $\frac{w^2}{16}$ entries of 1's. Thus, there exists a row $k$ in this rectangle having $\geq \frac{w}{16}$ 1's. Let $b_1, b_2, \ldots, b_p$ be the row indices of $B$ rows such that the entry $(b_r, k)$ is a 1, $1 \leq r \leq p$. Set $i_r = r_r - k$, $1 \leq r \leq p$. □

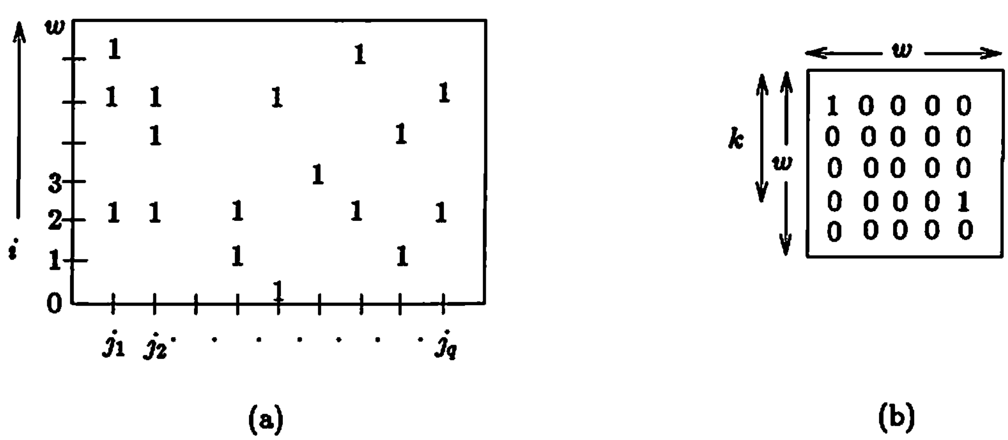



Figure 7: (a)The rectangle $d_{rb}$ vs $B_u$. (b) A kernel

**Lemma 2** *The volume $V_o$ of any electro-optical design using arbitrary row(column) major input format for image convolution satisfies $V_o = \Omega(nw)$.*

**Proof:** Let $k$ be as defined in claim 1. Choose the kernel $K$ as shown in figure 7 (b). All kernel weights are zero except for $K(1,1)$ and $K(k,w)$ entries. With the given kernel, the output $C(i,j)$ of the convolution becomes:

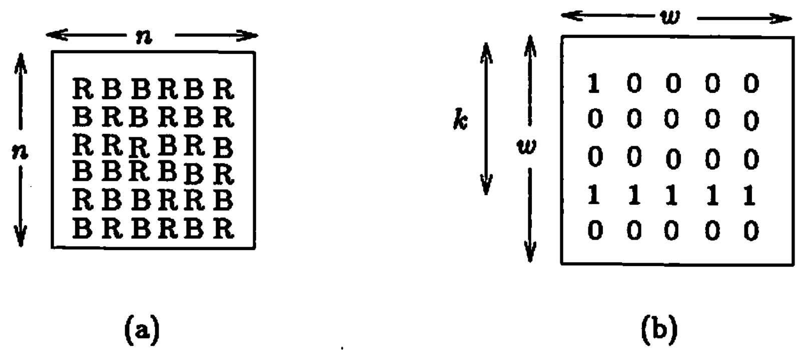



Figure 8: (a)Arbitrary coloring of pixels. (b) A kernel for general case

$$C(i,j) = I(i,j) + I(i+k, j+w-1) \quad for\ 1 \le i,j \le n-w+1; 1 \le k \le w-1.$$

Based on the above claim, it is clear that the output functions $C(i,j), i \in \{i_1, i_2, \ldots, i_l\}$; $1 \le j \le n - w + 1$, belong to $\pi_2^o$. The argument proceeds similar to that of *scanline input* case. $X_1$ corresponds to the bits associated with the output functions of the rows with indices $i_1, i_2, \ldots, i_l$. It is clear that $|X_1| \ge \frac{w}{16}(n - w + 1)$. Hence, the computational parallelepiped for the present input format contains at least $2^{\frac{w}{16}(n-w+1)}$ distinct planes. Since the result is true for any arbitrary partitioning of the input rows, we have, $d \ge 2^{\frac{w}{16}(n-w+1)}$. Thus, by proposition 1, $I_1(F) \ge \frac{w}{16}(n - w + 1)$ or $I_1(F) = \Omega(nw)$. Similar argument holds for arbitrary partitioning of the columns. Using theorem 1, the result follows. □

Now, we are ready to derive the bound for the *general case*. Let the pixels of the input image be arbitrarily colored as R(Red) or B(Blue) such that equal number of R and B pixels exist. Figure 8 (a) demonstrates one such coloring. The $\frac{n^2}{2}$ R(B) pixels correspond to $\pi_1^i(\pi_2^i)$. This *general case* can be reduced to the *arbitrary row(column) major* input case as follows.

Define the color of a row(column) to be the majority of the color of the pixels available in that row(column). Thus, a row or column is $R(B)$ if at least $\frac{n}{2} + 1$ pixels in that row or column are $R(B)$. All rows(columns) with equal number of $R$ and $B$ pixels are arbitrarily colored into $R$ or $B$ so as to minimize the difference in the number of $R$ and $B$ rows(columns). Let $n_r^r(n_r^c)$ be the number of $R$ rows(columns) in the image after the above coloring. It is easy to verify:

**Claim 2** *Any arbitrary coloring of pixels with equal number of R and B pixels gives rise to $n_r^r$ R rows and $n_r^c$ R columns, such that either (a). $(1 - \frac{1}{\sqrt{2}})n \le n_r^r \le \frac{n}{\sqrt{2}}$ or (b). $(1 - \frac{1}{\sqrt{2}})n \le n_r^c \le \frac{n}{\sqrt{2}}$.*

Without loss of generality, assume that the image consists of $n_r^r$ $R$ rows, $(1 - \frac{1}{\sqrt{2}})n \le n_r^r \le \frac{n}{\sqrt{2}}$. Define $\alpha$ to be the ratio of the number of $R$ to the number of $B$ rows. It is clear that $(\sqrt{2}-1) \leq \alpha \leq \frac{1}{(\sqrt{2}-1)}$. It is easy to verify that there exists at least one window $wd(x)$, $1 \leq x \leq n - 2w + 2$, consisting of $\frac{\alpha}{\alpha+1}w$ $R$ and $\frac{1}{\alpha+1}w$ $B$ rows. An argument similar to that of claim 1 leads to:

**Claim 3** *For $n \geq w(\frac{w}{2}+1)$, there exists a window $wd(x)$, $1 \leq x \leq n-2w+2$, consisting of $R(B)$ rows with indices $i_1, i_2, \ldots, i_l$, $l \geq \frac{w}{20}$, and an integer $k$, $1 \leq k \leq w-1$, such that the rows with indices $i_j + k$, $1 \leq j \leq l$, are $B(R)$.*

The above window WD can now be used to compute a lower bound on volume requirement for the general case.

**Theorem 2** *The volume $V_o$ of any electro-optical design for convolving a $w \times w$ kernel with a $n \times n$ image satisfies $V_o = \Omega(nw)$.*

**Proof:** Let $k$ be as defined in claim 3. Choose the kernel $K$ as shown in figure 8 (b). With the given kernel $K$, the output $C(i,j)$ of the convolution becomes:

$$C(i,j) \;=\; I(i,j) + \sum_{l=0}^{w-1} I(i+k, j+l) \;\; for\; 1 \leq i,j \leq n-w+1; 1 \leq k \leq w-1.$$

Any $R(B)$ row has at least $\frac{n}{2}$ $R(B)$ pixels in it. With the given kernel $K$, the computation of at least $(\frac{n}{2} - w + 1)$ bits in each of the $R(B)$ rows with indices $i_1, i_2, \ldots, i_l$ belong to $\pi_2^o$. Considering an environment similar to *arbitrary row major input* case, $|X_1| \geq \frac{w}{20}(\frac{n}{2} - w + 1)$. Now, it is easy to verify that $I_1(F) \geq \frac{w}{20}(\frac{n}{2} - w + 1)$. Thus, $V_o = \Omega(nw)$. □

# 4 Comparisons and Conclusions

In this paper, we considered an electro-optical system from a computational perspective. As an example, we studied the electro-optical resource requirements of a computationally intensive operation such as 2-D image convolution. We showed that any electro-optical system, regardless of implementation, must have $\Omega(nw)$ volume for convolving a $w \times w$ kernel with a $n \times n$ image, as long as the input pixels are given to the system only once. This lower bound on the volume requirement of an electro-optical system is also same as the minimum VLSI area requirement of a VLSI chip to carry out such a computation. Several known VLSI designs for 2-D image convolution are shown in figure 9.

| *Architecture* | *Design feature* | *Input format* | *Memory requirement* |
|---|---|---|---|
| Kung, Picard [17] | Pipelined 1-D Systolic array with cache | once only scan line | $2w^2 + (w-1)n$ |
| Kung, Ruane, Picard [18] | cascade of $w$ pipelines separated by buffers | once only scanline | $(n-w+2)w$ |
| Doshi, Varman [19] | cascade of $w^2$ cells without buffers | once only scanline | $nw$ |
| Shukla, Agarwal [20] | Semi-systolic array $w^2$ MAC cells in $w \times w$ array | once only scanline | $O(nw)$ |

Figure 9: Comparison of VLSI designs for image convolution

# References


[1] R. W. Keys. Physical limits in digital electronics. In *Proc. IEEE*, pages 740–767, 1975.

[2] T. E. Bell. Optical computing: a field in flux. *IEEE Spectrum*, August 1986.

[3] J. W. Goodman, F. J. Leonberger, S. Y. Kung, and R. A. Athale. Optical interconnections for VLSI systems. In *Proc. IEEE*, pages 850–866, 1984.

[4] H. I. Jeon and A. A. Sawchuk. Optical crossbar interconnections using variable grating mode devices. *Applied Optics*, 26:261, Jan 1987.

[5] A. A. Sawchuk, B. K. Jenkins, C. S. Raghavendra, and A. Varma. Optical crossbar networks. *IEEE Computer*, June 1987.

[6] J. W. Goodman. Fan-in and fan-out with optical interconnects. *Optica Acta*, 1489–1496, Dec 1985.

[7] L. D. Hutcheson and P. Haugen. Optical interconnects replace hardwire. *IEEE Spectrum*, 30–35, Mar 1987.

[8] D. Cassent. Acoustooptic linear algebra processors: architectures , algorithms and applications. In *Proc. IEEE*, pages 831–849, 1984.

[9] H. J. Caulfield, W. T. Rhodes, M. J. Foster, and S. Horvitz. Optical implementation of systolic array processors. *Optical Communication*, 40:86–90, 1981.

[10] W. T. Rhodes. Acousto-optic signal processing: convolution and correlation. In *Proc. IEEE*, pages 65–79, 1981.

[11] R. P. Bocker, S. R. Clayton, and K. Bromley. Electro-optical matrix multiplication using the twos complement arithmetic for improved accuracy. *Applied Optics*, 22(13):2019–2021, 1983.

[12] A. Huang. Parallel algorithms for optical digital computer. In *Proc. 10th Int. Optical Comp. Conference*, pages 13–17, 1983.

[13] A. Huang. Architectural considerations involved in the design of an optical digital computer. In *Proc. IEEE*, pages 780–786, 1984.

[14] Mary Eshaghian and V. K. Prasanna Kumar. VLSI electro-optical computers for signal and image processing. In *Third International Conference on Supercomputing*, 1988.

[15] S. Fujita, R. Aibara, M. Yamashita, and Tadashi Ae. A template matching algorithm using optically-connected 3-d VLSI architecture. In *14th Annual International Symposium on Computer Architecture*, pages 64–70, 1987.

[16] A. Rosenfield and A. Kak. *Digital Image Processing.* Academic Press, 1976.

[17] H. T. Kung and R. L. Picard. Hardware pipelines for multi-dimensional convolution and resampling. *IEEE Computer Society Workshop on Computer Architecture for Pattern Anal. and Image Database Managemaent*, 273–278, 1981.

[18] H. T. Kung, L. M. Ruane, and W. L. Yen. A two level pipelined systolic array for convolutions. In *CMU conference on VLSI Systems and Computations*, pages 255–264, 1981.

[19] K. Doshi and P. Varman. A modular architecture for image convolutions. In *Proceedings 14th annual International Symposium on Computer Science*, pages 56–63, 1987.

[20] S. B. Shukla and D. P. Agarawal. An architecture for real-time image convolution with varying kernel sizes. In *IEEE International Conference on Computer Architecture*, 1988.

[21] R. Barakat and J. Reif. Lower bounds on the computational efficiency of optical computing systems. *Applied Optics*, 26(6):1015–1018, 1987.

[22] J. D. Ullman. *Computational Aspects of VLSI.* Computer Science Press, 1984.

[23] A. C. Yao. The entropic limitations on VLSI computations. In *Proc. 13th Annual ACM Symposium on Theory of Computing*, pages 308–311, 1981.

[24] V. K. Prasanna Kumar. *Communication Complexity of various VLSI Models.* PhD thesis, Pennsylvania State University, 1983.

[25] C. D. Thompson. Area time complexity for VLSI. In *Proc. 11th Annual ACM Symposium on Theory of Computing*, pages 81–88, 1979.

[26] A. C. Yao. Some complexity questions related to distributed computing. In *Proc. 11th Annual ACM Symposium on Theory of Computing*, pages 209–213, 1979. Atlanta.

[27] J. N. Navarro, J. M. Llaberia, and M. Valero. Partitioning: an essential step in mapping algorithms into systolic array processors. *IEEE Computer*, 77–89, July 1987.