Reconfigurable Optical Interconnects for Computer Vision Applications

Poornima Lalwaney and Israel Koren
Department of Electrical and Computer Engineering
University of Massachusetts
Amherst, MA 01003

Abstract

We evaluate the advantages of reconfigurable optical interconnects within massively parallel systems due to their ability to provide versatile application-dependent network configurations. Furthermore, they are being considered as alternatives to electronic interconnects within high-performance computers because of their advantages of high bandwidth, low wire density and low power requirement at high data rates. Fiber optic interconnects based on wavelength division multiplexing and free-space holographic interconnects are two classes of optical interconnects that can support network reconfiguration. Using computer vision applications, we compare these two classes of optical interconnects with electronic interconnects taking into account the combined effects of link speeds, link latencies, system size, message size and network topologies feasible with current implementation capabilities.

1 Introduction

High bandwidth and low latency are the requirements for interconnection networks for high-performance computers. As individual processor data rates and complexities grow, electrical interconnects may not be able to support the speeds required by large multiprocessor systems [11]. Further, as the applications executed on these systems may require diverse communication patterns between processors, the underlying interconnection network has to be flexible so as to fully utilize the benefits of high-speed processing. Free-space and guided-wave optical interconnect configurations using a variety of switching devices and data multiplexing schemes have been investigated in parallel computers [11] [12]. Recent efforts in incorporating optical interconnects in multiprocessors includes a joint Honeywell-Thinking Machines project using fiber-optic interconnects in a Connection Machine prototype to lower wire density and enhance performance. By multiplexing a number of data channels onto a single fiber, a reduction in wiring density by a factor of 512 was attained [14]. Intel used fiber-optic interconnects in a Touchtone Supercomputer prototype to achieve the bandwidth of 1.6 Gbps per mesh interface node. This was required for scaling the mesh from 512 nodes to 1024 nodes [9]. Among the free-space interconnects efforts, a 64-processor three-dimensional mesh topology has recently been implemented at NTT Systems Laboratories using board-to-board free-space interconnects [20]. To study the advantages of network flexibility offered by optical interconnects, we consider in this paper, two promising schemes for optical interconnects in high performance multiprocessors. These are fiber-optic interconnects based on wavelength division multiplexing and free-space holographic interconnection schemes.

In optical link implementations, every unidirectional link requires a transmitter at the source node and a receiver at the destination node, transmitting and receiving data at a common wavelength. In wavelength division multiplexing (WDM) based networks, the bandwidth of the optical fiber is split into many wavelength channels, each channel carrying data at a particular wavelength [22]. The logical connectivity is obtained by assigning wavelengths to the system's transmitters and receivers. Reconfiguring the interconnection network to a different topology is a simple matter of wavelength reassignment, if the transmitters and/or receivers are tunable over the entire range of wavelengths used. Some performance issues on the reconfigurabilty offered by WDM-based fiber-optic networks in multiprocessors can be found in [1] [5]. Free-space interconnects is another class of optical interconnection schemes being considered for interprocessor networks [6] [11]. In free-space interconnects, a switching device inserted in the path of optically encoded data from the transmitter, directs the data to the appropriate receiver. This switching can be achieved by recording the desired source-destination communication patterns in a hologram. As holograms

0-8186-5830-4/94 $3.00

can encode a high density of data, a large number of switching patterns can be realized with this scheme. In this paper, we demonstrate the advantages of high speeds and network reconfigurabilty offered by these two classes of optical interconnects over electronic interconnects with the current implementation capabilities of optical and electronic technologies.

Existing massively parallel systems are interconnected using low node degree topologies (e.g., multidimensional mesh, fat tree) because of limitations on available bandwidth and wiring density when interconnecting a large number of processors using electronic interconnects. Besides, these topologies are suitable for local communication patterns, often encountered in scientific and engineering computations. In order to evaluate various optical and electronic interconnect options, we analyze the performance of these interconnects in image processing and computer vision systems. Image processing and computer vision tasks need high computation and communication rates [3]. The incoming data rate to a typical system is in millions of bytes of data per second. As the applications that are run on vision systems have diverse communication requirements, it is important that the communication network topology matches the communication structure of the executed algorithm. An important advantage of optical interconnects over electronic interconnects, is their ability to *dynamically change the system topology to best match the ezecuted algorithm.* To evaluate the performance of optical interconnects, we derive estimates for the computation times and communication overheads for a representative set of vision algorithms on the mesh and tree topologies.

Issues in realizing versatile network topologies using guided wave fiber-optic interconnects and free-space holographic interconnects are discussed in Section 2. The execution time estimates derived for a representative set of image processing and vision algorithms are presented in Section 3. In Section 4, we address some issues relevant to real-time image interpretation.

2 Network Flexibility Using Optical Interconnects

In this section, we outline the principles involved in realizing a network topology using WDM-based fiber-optic interconnects and free-space holographic optical interconnects.

2.1 Wavelength Division Multiplexing and Reconfigurability

In wavelength division multiplexing the large capacity of the optical fiber is utilized by splitting the available bandwidth into independent, noninteracting channels, thereby supporting simultaneous communication between many source-destination pairs on a common medium. In WDM-based systems, wavelength encoded signals from the transmitting nodes are multiplexed onto the fiber using a device called the passive star coupler. The transmitters at the network nodes are connected to the input ports of the star coupler. The signal power at each input port is divided uniformly amongst the output ports. Thus, wavelengths from all the transmitters appear at each output port. Demultiplexing is performed at the receivers by recovering the desired input port signal from the common output port signal. The desired connectivity amongst the system nodes can be obtained by appropriate wavelength assignments to the transmitters and receivers. This is illustrated for a seven node binary tree in Figure 1. In the figure, all links are bidirectional and are implemented using fixed wavelength transmitters and receivers. Thus, the degree of a node determines the number of transmitters and receivers at that node.

By using tunable transmitters and receivers, versatile interconnections can be realized. As the logical connectivity between nodes can be achieved by tuning the transmitters and receivers to the desired wavelengths, reconfiguration of the interconnection network to support a different topology can be achieved by wavelength reassignment. With the availability of narrow line width semiconductor lasers and inexpensive passive star couplers [2], WDM-based optical interconnects that support reconfiguration seem feasible for multiprocessors. Rapid progress is being made in the development of tunable devices, both in the range over which they are tunable, and their tuning times [4] [18]. Limited tuning range restricts the kinds of topologies that can be supported, and the tuning times of devices constitute the reconfiguration overheads. Both these parameters are decided by the method of tuning used. Current tuning ranges are in the $4 - 10nm$ range and the tuning times vary from nanoseconds to milliseconds [4]. In this paper, we assume conservatively that reconfiguration overheads are in the milliseconds range.

Currently available passive stars can handle a maximum of a hundred wavelengths. Star couplers with a maximum of 128×128 ports are feasible with currently available technology. An experimental ISDN switch architecture using eight 128×128 multiple passive stars to handle over ten thousand input port lines has recently been reported [4].

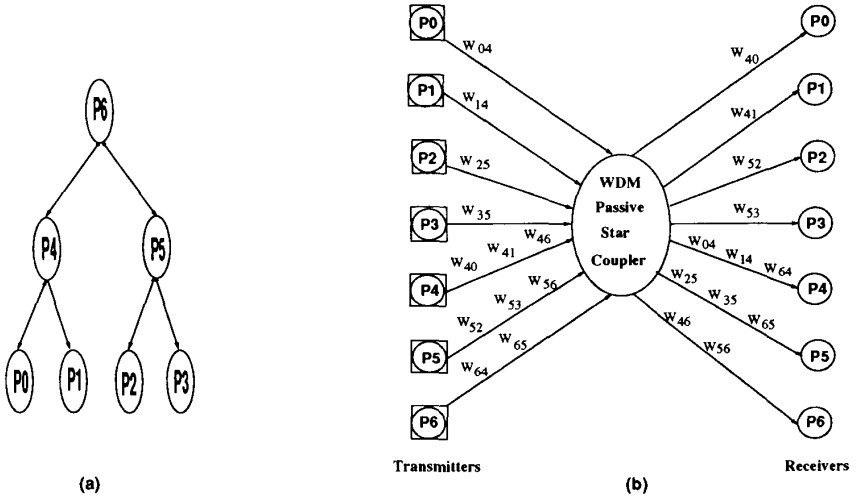


Figure 1: (a) A binary tree with seven nodes. (b) WDM star embedding of the tree. W_{ij} is the wavelength assigned for communicating from node i to node j. W_{ij} is assigned to the transmitter at processor i and the receiver at node j.

2.2 Holographic interconnects for implementing versatile topologies

In the model for free-space holographic interconnnects assumed here, light from the transmitters is coupled into fibers. The ends of the fibers from all the transmitters are aligned into a vertical plane referred to as the transmitter plane. Similarly, there is a receiver plane parallel to the transmitter plane. Light incident on the fibers in the receiver plane is carried to the individual receivers at the nodes. The network connectivity can be achieved, if the light from a transmitter is coupled to the desired fiber in the receiver plane. This switching of the incident light direction is acheived by inserting a hologram in the path of light from the transmitter plane. As seen in Figure 2, the fiber from the transmitter at node i illuminates a portion of the hologram, referred here as subhologram i. Information on the positions of the receivers at nodes adjacent to node i in the desired topology is prerecorded in the subhologram. Light encoded data from the transmitter at node i is directed to all the desired receivers by the subhologram by space-variant imaging. Details on different kinds of holograms (transmissive, reflective) and the available imaging and recording techniques can be found in [6], [15] and [19].

Note that all transmitters and receivers operate at a single wavelength, hence there is no wavelength limitation as in the WDM scheme. Also, if the degree of connectivity desired at each node is low, then these source-destination patterns for several topologies may be recorded in a single hologram. Thus as all the desired connectivity could be established by the subhologram for the node, no reconfiguration overhead will be incurred. However, the transmitter and receiver power limitations may limit the fan-in and fanout of nodes. Crossbar interconnections of size 64 may be achieved with current holographic fabrication capabilities [19]. With improvements in holographic fabrication and recording techniques, larger crossbar networks and many versatile topologies may be realized. In these free-space interconnects, the alignment of the transmitter plane, the individual subholograms and the receivers is a critical issue. Misalignment degrades the received power and increases the bit error rate.

3 Image Processing and Vision Related Algorithms

Computer vision uses algorithms from image processing, graph theory, databases and artificial intelligence [23]. These algorithms are generally classified as low-level or high-level algorithms. Existing computer vision and image analysis systems are organized as hierarchical structures. The lower levels of the system perform regular, computationally inten-

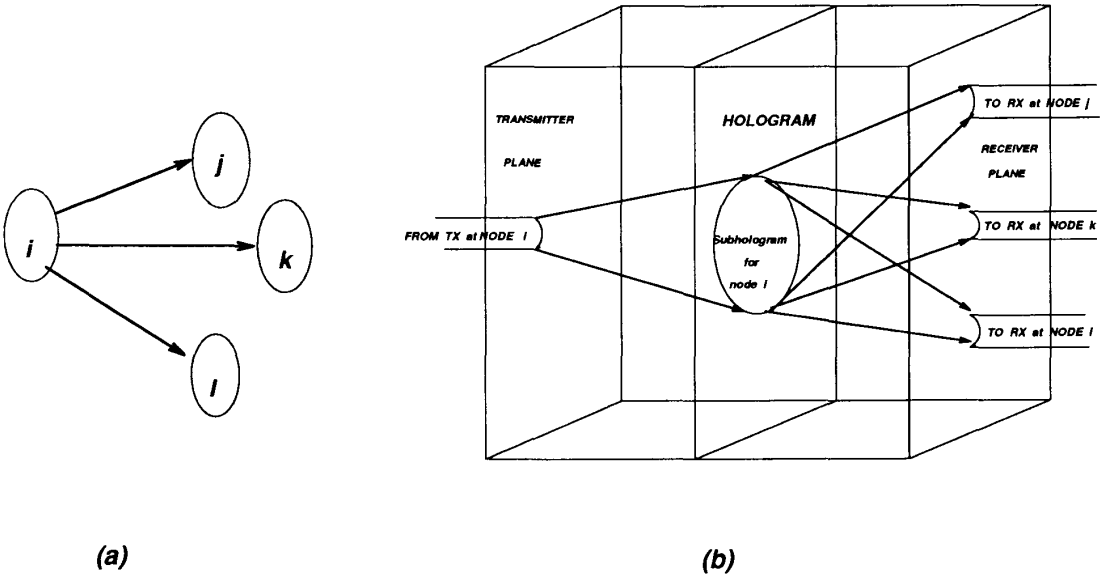


Figure 2: Free-space holographic implementation for node i. (a) The desired interconnection for node i. (b) Holographic implementation.

sive tasks on all pixels of the image while the higher levels perform image analysis tasks with data dependent communication requirements. In this section, we analyze some representative algorithms from various levels of image processing. The algorithms considered are the image smoothing application which is a typical window-based low-level image processing application, the Hough Transform algorithm which uses pixel data to detect patterns in an image, and an orthogonal clustering algorithm which is a high-level algorithm used to classify patterns into disjoint groups. This algorithm has a data dependent convergence criterion.

The window-based operations like image smoothing require local communication and so the preferred topology is the mesh. Both the Hough Transform and the clustering algorithm include broadcast and fan-in communication phases in some steps of algorithm execution. The tree topology or some variation of it is therefore the preferred topology for the two algorithms. We analyze the performance of the three algorithms on both the mesh and tree topologies. The performance measure considered is the algorithm completion time. The algorithm completion time consists of the computation time and the communication time. Some implementation related issues considered in estimating the communication overheads are discussed below.

a. Physical link implementation

The properties of electronic and optical link implementation is reflected in the effect of link latency and link speed on the communication time. The setup time for the link is denoted by α, and β denotes the time for a 32-bit floating-point word transfer. The setup time for the optical link includes the time for electrical to optical conversion, and is therefore assumed to be higher than that of the electronic interconnects. The link latency and link speed parameter values assumed for electronic interconnects are in the range of current or upcoming standards for high-speed electronic interconnects like the Scalable Coherent Interface and the Futurebus+ [10]. We assume, conservatively, that the setup time for the electronic link, denoted by α_e, is $140ns$. Moreover, for the electronic implementation, a 32-bit word can be transferred over parallel lines at 100 Mb/s (i.e, $\beta_e = 10ns$). Fiber optic transmission is serial. A parallel 32-bit transfer would require 32 transmitters and receiver pairs per node, thereby making the system costs prohibitive. For the analysis in this paper, we assume that an internode optical connection is implemented as four 1 Gb/s optical links. The time to transfer a floating-point word between two nodes denoted by β_o, is therefore $8ns$. The link

speed of 1 Gb/s is considered feasible for commercial systems. Interface devices for 1 Gb/s links are being designed using the Fiber Channel Standard [8]. The observed latency in Intel's effort in incorporating fiber-optic interconnects in the Touchtone supercomputer program was 400 ns [9]. We therefore assume the value of α_o to be 400 ns. The variation in algorithm performance with the latency and bandwidth of optical interconnects is addressed in Section 4.

b. Hardware limitations on topologies

Some limitations of electronic and optical topologies are discussed here. For electronic interconnects, we assume no flexibility in the underlying interconnection network as existing massively parallel systems have fixed topologies. The performance of the algorithms is therefore studied on the fixed topologies considered, namely, the electronic mesh and the electronic tree. Among the optical interconnects, flexibility can be achieved by using WDM-based fiber-optic interconnects or holographic free-space interconnects. Flexibility in WDM-based interconnects is achieved by retuning the transmitters and receivers to a different wavelength. As each tuning phase has an associated reconfiguration overhead, within limitations of current technology, we do not allow for frequent tuning of these interface devices. This implies that although broadcast can be achieved in a single phase using WDM-based implementations, we implement the broadcast and fan-in phases in the algorithms using a tree topology in $log_2 N$ steps. This limits both the number of wavelengths used in the system and the reconfiguration overheads.

In holographic interconnects, flexibility is achieved by the hologram sections that selectively focus incoming data from transmitters onto appropriate recievers. Due to fabrication limitations and power constraints, fanout per node is limited to 64 and fan-in is limited to 8 [19]. For an N node network, broadcast can be achieved in $log_{64} N$ steps. Due to fan-in restrictions per node, we assume a tree topology for fan-in.

We derived time estimates for the execution time of the image smoothing, Hough Transform and clustering algorithms on the mesh and tree topologies for various implementations. A brief explanation of the algorithms followed by the results for the algorithm completion time is now presented. Details of the derivations of all the expressions that appear in this section can be found in [17]. We assume that an N processor system is used to process the incoming $M \times M$ image. The time for a single word addition, multiplication and division operation is denoted by t_{add}, t_{mult} and t_{div}, respectively.

3.1 Image Smoothing : A Representative Window Based Operation

In image smoothing, a pixel in the smoothed image contains its pixel value averaged over a $w \times w$ window around it. If $A(i,j)$ represents the pixel value in the original image at position (i,j), the pixel value $A'(i,j)$ in the smoothed image within a 3×3 window is given by the following expression [21]:

$$A'(i,j) = [A(i-1,j-1) + A(i,j-1) + A(i+1,j-1) + A(i-1,j) + A(i,j) + A(i+1,j) + A(i-1,j+1) + A(i,j+1) + A(i+1,j+1)]/9.$$

The algorithm completion time estimates derived for the image smoothing algorithm on the mesh and tree topologies taking implementation issues into account are presented below.

$$T_{mesh}^{smooth} = \frac{M^2}{N} \cdot (8 \cdot t_{add} + t_{div}) + 4 \cdot \alpha + 4 \cdot \beta \cdot (\frac{M}{\sqrt{N}} + 1) \quad (3.1)$$

$$T_{tree}^{smooth} = \frac{M^2}{N} \cdot (8 t_{add} + t_{div}) + + \left(\frac{k}{4} \cdot (\frac{k}{2} + 1) + 2^{\frac{k}{2}} \cdot (\frac{3k^2}{8} + \frac{k}{4})\right) \cdot (\alpha + \beta \cdot (\frac{M}{\sqrt{N}} + 1)) \quad (3.2)$$

where $k = \lceil \log_2 N \rceil$. As can be seen from the above expressions, there is a larger communication overhead in implementing the image smoothing algorithm on the tree topology.

3.2 Hough Transform for Line Detection

The Hough Transform is a widely used technique for feature detection in images. The pixels in the image space are mapped onto the parameter space of the features being detected. We consider here the Hough Transform for line detection. A line segment can be characterized by two parameters ρ and θ, where ρ denotes the distance of the normal to the line from the origin, and θ represents the angle made by the normal with the positive direction of the horizontal image axis. The principle of the Hough Transform is to transform the pixel (i,j) in the image space to a curve in the (ρ,θ) space. All pixels in the image representing a line characterized by the values (ρ,θ), satisfy the equation $\rho = i \sin\theta + j \cos\theta$. To compute the Hough Transform, the (ρ,θ) space is quantized, with ρ_{max}, θ_{max} bins along the ρ and θ axes, respectively. An accumulator array of dimension $\theta_{max} \times \rho_{max}$ stores the image pixels count in the various (ρ,θ) bins. Image pixels are scanned over the θ quantization values, and the corresponding (ρ,θ) bins are incremented in the

accumulator array. After the image is scanned completely, local maxima in the accumulator array indicate the presence of lines in the image space.

The data partitioning method to implement the Hough Transform [3] of an $M \times M$ image on an N processor system was considered in deriving the expressions presented below. This is a three phase algorithm consisting of parallel computation of partial accumulator arrays, merging of partial accumulator arrays, followed by the broadcast of the final accumulator array and parallel computation of local maxima. Note that in the expressions presented below, the superscripts refer to the phase of the algorithm and the subscripts refer to the topology.

$$T^1 = (2 \cdot t_{mult} + t_{add}) \cdot \frac{M^2}{N} \cdot f \cdot \theta_{max}$$

$$T^2_{tree} = \rho_{max}\theta_{max} \cdot \log N \cdot t_{add} + (\alpha + \rho_{max}\theta_{max}\beta) \cdot \log N$$

$$T^2_{mesh} = \rho_{max}\theta_{max} \cdot (2(p-1) \cdot t_{add} + 2\frac{\sqrt{N}}{p} t_{add})$$

$$+2 \cdot (\sqrt{N} - 1) \cdot (\alpha + \rho_{max}\theta_{max} \cdot \beta)$$

$$T^3_{mesh} = \rho_{max} \cdot \frac{\theta_{max}}{N} \cdot w^2 \cdot t_{add} + (\alpha + \rho_{max}\theta_{max} \cdot \beta) \cdot 2\sqrt{N}$$

$$T^3_{tree} = \rho_{max} \cdot \frac{\theta_{max}}{N} \cdot w^2 \cdot t_{add} + (\alpha + \rho_{max}\theta_{max} \cdot \beta) \cdot \log N$$

where f denotes the maximum fraction of dark pixels in the image and w is the window size for computing local maxima. Note that p denotes the submesh size for parallel merge and should be $N^{1/4}$ to minimize algorithm completion time. The total time for Hough Transform on the tree and mesh topologies are given below [17]

$$T^{hough}_{tree} = T^1 + T^2_{tree} + T^3_{tree} \quad (3.3)$$

$$T^{hough}_{mesh} = T^1 + T^2_{mesh} + T^3_{mesh} \quad (3.4)$$

On the holographic interconnects, the broadcast takes $log_{64}N$ steps as opposed to $log_2 N$ steps on the tree. The time to perform the Hough Transform on the holographic interconnects is given by the following expression [17]

$$T^{hough}_{holo} = T^1 + T^2_{tree} + \rho_{max} \cdot \frac{\theta_{max}}{N} \cdot w^2 \cdot t_{add}$$

$$+(\alpha + \rho_{max}\theta_{max} \cdot \beta) \cdot \log_{64} N \quad (3.5)$$

A substantial overhead is incurred in implementing the Hough Transform on the mesh. For large system sizes, the computation time on the mesh implementation increases with the number of processors. When increasing the number of processors, the number of partial accumulator arrays to be merged increases and as a result, more additions have to be performed. The communication overhead on the mesh varies as $\sqrt{N}$, and for large system sizes, the communication time dominates the computation time on the mesh. The variation in algorithm completion time with the number of available processors for holographic and tree based implementations is shown in Figure 3. The image size is 512×512. The values of w, f, ρ_{max}, and θ_{max} were taken as 3, 0.5, 512 and 180, respectively. A $10ns$ processor clock was assumed for this system and the time for a single addition, multiplication and division operation is taken as $10ns$, $80ns$ and $120ns$, respectively. The communication patterns of fan-in and broadcast required in some phases of the Hough Transform are better met with the tree topology than the mesh topology. Mesh based implementation curves are not included in the figure as their execution time is two to ten times larger than the electronic tree for the system sizes considered. This shows that network topology is an important factor in algorithm performance. The holographic interconnect has the lowest communication overhead and therefore the lowest overall execution time. Among the implementations of the tree topology, the WDM-based optical tree implementation performs better than the electronic tree implementation. Note here that although optical links have a higher latency than the electronic links, the optical tree performs better because the amount of data transferred over a link in a communication phase is large (512×180 words). Thus the speed advantage of optical interconnects offsets the high latency.

3.3 Orthogonal Clustering for Pattern Classification

Orthogonal clustering is a high-level image processing algorithm used to classify input patterns into disjoint classes. The input to the algorithm is a set of P pattern vectors, each vector having m attributes. The goal of the algorithm is to classify the input pattern into K clusters. We apply the $K - means$ algorithm to classify patterns into different clusters [13]. A pattern X is assigned to cluster j, if its Euclidean distance from the center of the jth cluster, is less than that from all other cluster centers. Once all the P pattern vectors are assigned cluster labels, the cluster means are updated. The cluster labelling procedure is repeated until the algorithm converges. At this point, the cluster means are stable. We computed an upper bound on the number of iterations the algorithm would need to converge. This turns out to be $P \cdot (K-1) - \frac{(K-1)\cdot(K-2)}{2}$ and is denoted by

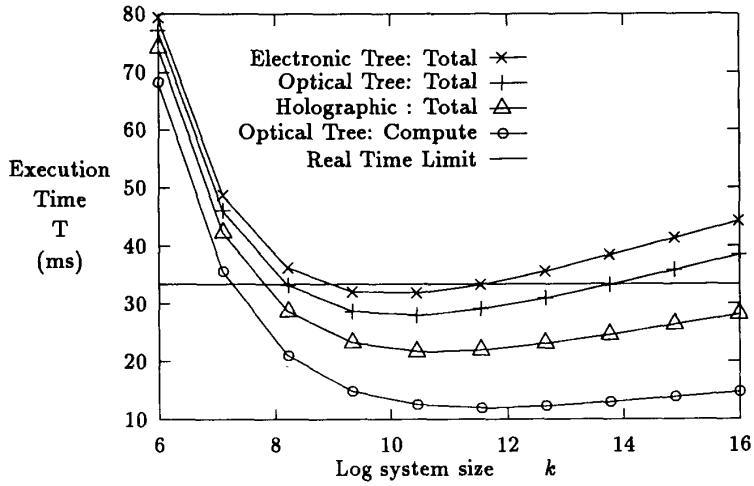


Figure 3: The execution time of Hough Transform for line detection on the mesh and tree as a function of the system size (2^k processors).

I_{max}. This was required to estimate the worst case algorithm completion time on an N node distributed memory system. A single iteration of the algorithm was implemented in three phases. The expressions for the time taken by the three steps in a single iteration are summarized below [17].

$$T^1_{mesh} = 2 \cdot (\sqrt{N} - 1) \cdot (\alpha + \beta \cdot K \cdot m)$$

$$T^1_{tree} = \log N \cdot (\alpha + \beta \cdot K \cdot m)$$

$$T^1 = \frac{P}{N} \cdot (3m \cdot K + m + 1) \cdot t_{add} + K \cdot m \cdot t_{mult}$$

$$T^2_{mesh} = K \cdot (m+1) \cdot (4 \cdot N^{\frac{1}{4}} - 4) \cdot t_{add} + 2 \cdot (\sqrt{N} - 1) \cdot (\alpha + \beta \cdot K \cdot (m+1))$$

$$T^2_{tree} = K \cdot (m+1) \cdot \log N \cdot t_{add} + \log N \cdot (\alpha + \beta \cdot K \cdot m)$$

$$T^3 = K \cdot m \cdot (t_{div} + t_{add})$$

Here, the superscripts refer to the phase of the algorithm, and the subscripts refer to the topology. T^1 denotes the computation time for the label assignment phase and T^3 denotes the computation time for updating cluster means and checking for algorithm convergence. The worst case algorithm completion time on the two topologies is given by the following expressions:

$$T^{clus}_{tree} = (T^1_{tree} + T^1 + T^2_{tree} + T^3) \cdot I_{max} \quad (3.6)$$

$$T^{clus}_{mesh} = (T^1_{mesh} + T^1 + T^2_{mesh} + T^3) \cdot I_{max} \quad (3.7)$$

To evaluate the performance of different implementations, we consider an example by assuming a set of 128 input patterns that have to be assigned to 8 clusters. Each pattern is described by two attributes. The parameters values for the arithmetic operations, the link latencies and speeds are the same as those used in the Hough Transform. Our estimates show that on a 128 node system (the maximum number of processors that can be used for clustering if $P = 128$), it takes a maximum of 9.41 *ms* on the free-space holographic implementation, 11.72 *ms* on the WDM-based optical tree implementation, and 26.26 *ms* on the WDM-based optical mesh implementation. The corresponding worst case algorithm completion time on the electrical implementation of the tree and mesh is 9.02 *ms* and 18.16 *ms*, respectively. Note that the performance of the electronic interconnects is better than that of the optical interconnects for fixed topology implementations. As the latency of the optical interconnects is high and the amount of data transferred per link in a communication phase is small, the

latency of the optical interconnects dominates. The performance of the electronic tree is better than all other schemes for this algorithm.

4 Real-Time Vision Considerations

The performance of an image processing and vision system depends on the sequence of tasks performed and on the system architecture. A typical image processing system receives medium resolution (512×512 pixels) images at a standard frame rate of 30 frames/second and with three bytes per color pixel. This represents a rate of almost 24 million bytes of data per second [3]. To interpret images or data at this rate, a large number of high-speed processors are required. The algorithms used in image processing and analysis have varying computational and communication requirements. The performance of these systems can be improved if the communication structure offered by the interconnection network topology matches the communication requirements of the algorithm. Existing vision systems have hierarchical and partitionable architectures, and provide limited reconfigurability in their electronically implemented networks [23].

Optical interconnects between high-speed processors may provide real-time image interpretation capabilities through the speed and flexibilty they offer in the underlying network. In this section, we compare the performance of the fixed and flexible optical implementations against the fixed topology implementations of the electronic topology by considering a sample vision application. This application is later used to study issues in real-time image processing.

4.1 Effect of Reconfiguration, Communication Latency and Speed

Typically, an image processing application consists of a sequence of low-level algorithms followed by a few high-level algorithms. The low-level algorithms extract preliminary data from the incoming image. This is then used by the high-level algorithms in image analysis tasks. The sample application considered here, therefore, involves a sequence of window-based operations followed by the Hough Transform to detect lines in an image. The orthogonal clustering algorithm is then applied to the detected lines to group them into disjoint classes.

We now compare the performance of the optically implemented topologies against the electronically implemented ones for the sample application. The performance measure is the total time to execute the sequence of algorithms in the application. We assume that the time estimate derived for the image smoothing operation is representative of a window-based operation. Algorithm completion time expressions derived for the Hough Transform and clustering algorithm are used here. The parameter values for link properties, time for arithmetic operations, and parameters of the individual algorithms are identical to the ones in Section 3.

Figure 4 shows the performance of an application on a 512 node system. The application consists of 20 window-based operations followed by a Hough Transform and a single clustering algorithm. The figure shows the variation in computation and communication time for implementing the application on various electronic and optical implementations. Here *E mesh* and *E tree* refer to the electronic implementations of the mesh and tree topologies, respectively. *O mesh* and *O tree* denote the optical mesh and optical tree implementations based on fiber-optic WDM interconnects, respectively. *O rec* refers to the reconfigurable scheme implemented using fiber-optic WDM-based interconnects, and *O holo* refers to the holographic interconnects. The mesh is the preferred topology for window-based operations and the tree is the preferred topology for the Hough Transform and the clustering operation. The Hough Transform is the dominant algorithm in the application as it requires the largest component of the application execution time. As seen from the figure, the communication overhead in implementing the application on the optical and electronic mesh topologies are high. This shows that network topology is as important as the implementation technology. Among the optical and electronic implementations of the tree topology, we see that the electronic tree performs better than the optical tree. This seems to be counter-intutive. This behavior is explained by considering the overhead in implementing the mesh-based algorithms on the tree topology. The latency of the optical interconnects dominates in this case, because the number of words transferred in a communication phase is small, the speed advantage of optical interconnects cannot compensate for the high latency. Optical interconnects on a preferred topology can outperform the electronic implementations when the amount of data transferred in a communication phase is large. This was observed in Figure 3 for the Hough Transform algorithm.

The WDM-based reconfigurable optical interconnects use the mesh topology for window-based operations and the tree topology for the Hough Transform and clustering algorithm. This reduces the communication overheads, as can be seen from Figure 4. The reconfiguration overhead is not included in the figure. It varies from *ns* to *ms* depending on the method

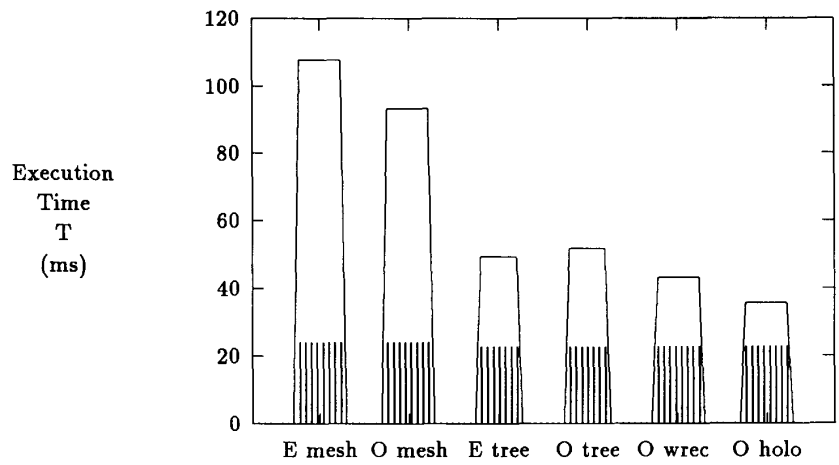


Figure 4: The variation of the sample application execution time for various implementations on a 512 node system. The shaded part of each bar plot is the computation time.

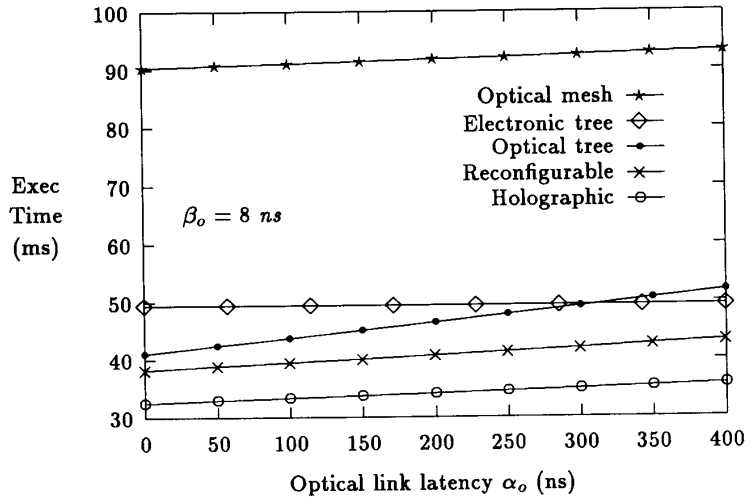
of tuning used. Taking the reconfiguration overheads into account, the WDM-based reconfigurable optical interconnects perform better than the electronic tree interconnects if the reconfiguration overhead is less than 6 ms. For holographic interconnects, the mesh and tree topologies, and the broadcast communication patterns are encoded in the hologram. They therefore exhibit the lowest communication overhead. Free-space holographic interconnects can provide a 17 % improvement in execution time, and a 37 % improvement in communication time over the WDM-based reconfigurable schemes. They provide a 27 % improvement in overall execution time and a 51 % improvement in communication time over the electronic tree implementations. As can be seen from the figure, the flexible optical interconnects perform better than the other fixed optical and electronic schemes.

Based on speed and power considerations, electronic interconnects beyond the speeds of a few Gbps may not be feasible for multiprocessors. Optical interconnects on the other hand are expected to support much higher data rates [7]. With improvements in electro-optic interface technology, it may be possible to achieve lower latency for optical interconnects. Figure 5 depicts the effect of each of these parameters on the execution time of the sample application for various optical implementations on a 512 node system. The optical tree and the reconfigurable schemes perform better than the electronic tree for $\alpha_o < 300$ ns and $\beta_o < 7$ ns. We note that over the wide range of communication parameter values considered in the figure, the holographic interconnects have the best performance.

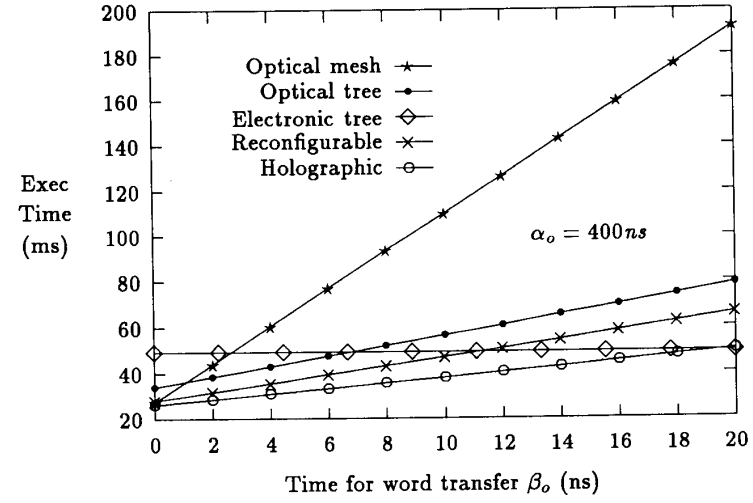
4.2 Real-Time Constraints

In this section, we study the effect of imposing a constraint on the total execution time for the sample application. The objective is to execute the given application in real time. Typically, an image processing system receives images at the rate of 30 images per second. We therefore assume that the real-time limit on the sample application is $33.33ms$. By varying the system size and the algorithm mix in the sample application, we estimate the system sizes and interconnection topology implementations that meet the real-time limit.

As noted earlier, the Hough Transform is the dominant algorithm in the application. As seen in Figure 3, a system of 2^8 nodes using holographic interconnects can just meet the real-time constraints for a single Hough Transform. With the optical tree implementation, system sizes from 512 to 8192 nodes are required to perform a single Hough Transform within real-time



(a)



(b)

Figure 5: The variation of execution time with (a) latency, and (b) time for a floating-point word transfer, for a sample application consisting of 20 window based operations, a single Hough Transform and one clustering algorithm on a 512 node system.

limits. For the electronic tree, system sizes between 2^9 and 2^{11} nodes meet the real-time limit considered here. Thus it may be possible to execute an application consisting of a sequence of window-based algorithms followed by a single Hough Transform and a clustering algorithm. As the optical and electronic mesh implementations do not meet real-time limits for a single Hough Transform, they are excluded from further discussion.

We now vary the system size and look for implementations that would execute the complete application within the real-time limit. Figure 6 shows the variation of the total execution time of the application with the number of window-based operations, on a 1024 node system. As can be seen from the figure, the holographic free-space implementation is the only implementation that meets the real-time limit. The maximum number of permissible window-based operations is 37. None of the other electronic or optical schemes considered meet the real-time limit. On varying the system size in powers of 2, we have noted [17] that for system sizes below the 1024 node system, none of the optical or electronic implementations meet the real-time limit. For a 2048 node system, the holographic implementation is still the only one that meets the real-time limit.

We now consider the possibility of pipelined execution of algorithms that make up the sample application. As noted earlier, the minimum system size to implement a single Hough Transform on the free-space holographic, optical tree and electronic tree is 256, 512 and 512 nodes, respectively. The Hough Transform and the clustering algorithm can individually meet the real-time limit for these system sizes. We therefore assume that they are performed on seperate sets of nodes so that their execution may be overlapped. Following our earlier assumption, we use 128 nodes connected according to the tree topology for the clustering algorithm. As the window-based operations and the Hough Transform use pixel data from the incoming image, we assume that they are performed on the same set of nodes. As the line patterns detected by the Hough Transform are used as inputs to the clustering algorithm, we assume that this pattern data is transferred between the two sets of nodes through a dedicated link connecting the root node of the tree used for the Hough Transform and the root node of the tree used for the clustering algorithm. The execution of the clustering algorithm on data from one image is overlapped with the execution of the window-based operations and the Hough Transform on the next image data. Using this pipelining scheme, Table 1 summarizes the system sizes required, and the permissible number of window-based operations that can be executed within the real-time limit for different topology implementations. We have shown in [17] that the sum of the time required to transfer the largest pattern set between the two sets of nodes, and the time to execute the clustering algorithm in the implementations considered is less than the time required to execute the Hough Transform. The system size required is the sum of the system sizes required for the Hough Transform and the clustering algorithm. We see that although the electronic tree meets the real-time constraints, the maximum number of permissible window-based operations that can be performed within the real-time limit is small. Among the flexible optical schemes, holographic interconnects can meet the real-time limit with the minimum system size of 384 nodes and the maximum number of permissible window-based operations limited to 13. A system size of 640 nodes allows a reasonable number of window-based operations.

In this section we have demonstrated that systems with hundreds of nodes are needed to meet the real-time processing constraints. As noted in Section 2, there are limitations on the number of wavelengths and number of inputs to a passive star coupler for implementing WDM-based fiber-optic interconnects. We have shown [17] that by using multiple star couplers and appropriately mapping nodes onto these couplers, networks with hundreds of nodes connected according to these topologies can be realized. Our report [17] also includes some issues in realizing large networks using holographic interconnects. For optical interconnects to be commercially viable in multiprocessors, the hardware interface costs have to be reduced. Some variations of network topologies to reduce interface costs have been reported in [16].

5 Conclusions

We have demonstrated that reconfigurable optical interconnects have advantages over fixed-topology electronic interconnects in massively parallel systems. This is due to the higher bandwidth they provide, and their ability to provide versatile application dependent network configurations. In this paper, we estimated the performance improvement of optical interconnects over electronic interconnects in image processing and computer vision systems. The performance of a sample vision application was analyzed taking into consideration the effect of latency, bandwidth, available network size, the preferred topology of the algorithm in the application and limitations of electronic and optical technologies. To study the performance en-

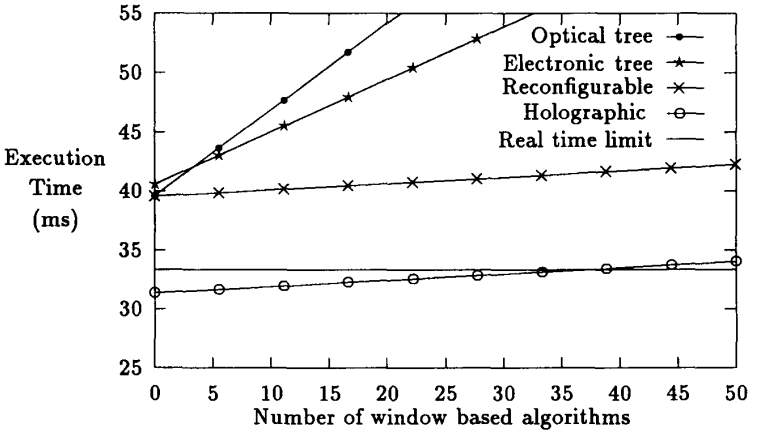


Figure 6: The execution time for a sample application on a 1024 processor system with different interconnection network implementations.

Type of interconnect	*System Size H.T + Clus = Total*	*Time for H.T ms*	*Max. Window operations*
Holographic	256 + 128 = 384	30.45	13
	512 + 128 = 640	24.26	86
	1024 + 128 = 1152	21.99	587
WDM Reconfigurable	512 + 128 = 640	29.42	37
	1024 + 128 = 1152	27.89	102
Electronic Tree	512 + 128 = 640	32.71	1
	1024 + 128 = 1152	31.58	3

Table 1: System sizes that meet real-time limit using pipelining for various interconnects.

hancements that can be achieved by using flexible optical interconnects, we considered two classes of optical interconnects that may be used to implement versatile topologies. These are fiber-optic interconnects based on wavelength division multiplexing and free-space holographic interconnects. We demonstrated that flexible optical interconnects performed better than fixed topology electronic and optical interconnects. Among the flexible schemes, the free-space holographic interconnects performed better than fiber-optic interconnects based on wavelength division multiplexing. Real-time limits on the sample application could be best met with holographic interconnects.

References

[1] K. A. Aly and P. W. Dowd, "Parallel Computer Reconfigubarility Through Optical Interconnects," *Proceedings of the International Conference of Parallel Processing*, August 1992.

[2] C. A. Brackett, "Dense Wavelength Division Multiplexing Networks: Principles and Applications," *IEEE Journal on Selected Areas in Communications*, Vol. 8, No. 6, pp. 948-964, August 1990.

[3] A. N. Choudhary, J. H. Patel, *Parallel Architectures and Parallel Algorithms for Integrated Vision Systems*, Kluwer Academic Publishers, Boston, 1990.

[4] A. Cisneros and C. A. Brackett, "A large ATM Switch Based on Memory Switches and Optical Star Couplers," *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 8, pp. 1348-1360, October 1991.

[5] P. W. Dowd, "High Performance Interprocessor Communication Through Optical Wavelength Division Multiple Access Channels," *Proceedings of the Computer Architecture Conference*, pp. 96-105, May 1991.

[6] M. R. Feldman and C. C. Guest,"Computer Generated Holographic Optical Elements for Optical Interconnection of Very Large Scale Integrated Circuits", *Applied Optics*, Vol. 26, No. 20, pp. 4377-4384, October 1987.

[7] M. R. Feldman, S. C. Esener, C. C. Guest and S. H. Lee,"Comparison Between Optical and Electronic Interconnects Based on Power and Speed Considerations", *Applied Optics*, Vol. 27, No. 9, pp. 1742-1751, May 1988.

[8] Fiber Channel Physical Layer Specification (FC-PH), Rev 2.1, FC-P/91-001R2.1, ANSI X3T9.3/90-019, May 1991.

[9] B. E. Floren et al, "Optical Interconnects in the Touchtone Supercomputer Program," *Integrated Optoelectronics for Communication and Processing*, Proc. SPIE 1582, pp. 46-54, 1991.

[10] Futurebus+ Interface Family Data Manual, Texas Instruments, Revision 4.1, Sept. 1992.

[11] J. W. Goodman, F. J. Leonberger, S. Y. Kung and R. A. Athale, "Optical Interconnections for VLSI Systems," *Proceedings of the IEEE*, Vol. 72, No. 7, July 1984.

[12] A. Guha, J. Bristow, C Sullivan and A. Husain, "Optical Interconnections for Massively Parallel Architectures," *Applied Optics*, Vol. 29, No. 8, March 1990.

[13] K. Hwang, H. M. Alnuweiri, V. K. Prasanna Kumar and D. Kim,"Orthogonal Multiprocessor Sharing Memory with an Enhanced Mesh for Integrated Image Understanding, " *CVGIP: Image Understanding*, Vol. 53, No. 1, pp. 31-45, January 1991.

[14] B.O. Kahle, E.C. Parish, T.A. Lane and J. A. Quam,"Optical Interconnects for Interprocessor Communications in the Connection Machine," *IEEE Conference on Computer Design*, Cambridge, MA, October 1989.

[15] R. K. Kostuk, J. W. Goodman and L. Hesselink, " Design Considerations for Holographic Optical Interconnects," *Applied Optics*, Vol. 26, No. 18, pp. 3947-3952, September 1987.

[16] P. Lalwaney, L. Zenou, A. Ganz and I. Koren, "Optical Interconnects for Multiprocessors: Cost Performance Trade-offs," *Proceedings of the* 4th *Symposium on Frontiers of Massively Parallel Computation*, pp. 278-285, October 1992.

[17] P. Lalwaney and I. Koren, "Optical Interconnects for Vision Systems," TR-93-CSE-9, University of Massachusetts, 1993.

[18] P. F. Moulton,"Tunable Solid State Lasers," *Proceedings of the IEEE*, Vol. 80, No. 3, pp. 348-364, March 1992.

[19] T. M. Pinkston, "The GLORI Strategy For Multiprocessors: Integrating Optics Into the Interconnect Architecture, " CSL-TR-92-552, Stanford University, 1992.

[20] T. Sakano, T. Matsumoto, and K. Noguchi, "A Three-Dimensional Mesh Multiprocessor System Using Board-to-Board Free-Space Optical Interconnects: COSINE-III," *International Conference on Computer Design* , pp. 278-283, Oct. 1993.

[21] H. J. Siegel, J. B. Armstrong, and D. W. Watson, "Mapping Computer-Vision-Related Tasks onto Reconfigurable Parallel-Processing Systems," *IEEE Computer*, Vol. 25, No. 2, February 1992.

[22] Special Issue on "Dense Wavelength Division Multiplexing Techniques for High Capacity and Multiple Access Communication Systems," *IEEE Journal on Selected Areas in Communications*, August 1990.

[23] C. C. Weems, "Architectural Requirements of Image Understanding with Respect to Parallel Processing," *Proceedings of the IEEE*, Vol. 79, No. 4, April 1991.