Parallel Computing: State-of-the-Art and Perspectives
E.H. D'Hollander, G.R. Joubert, F.J. Peters and D. Trystram

Effects of Nondeterminism on the Predicted Speedup of Scheduling of Low Level Computer Vision Algorithms on Networks of Heterogeneous Machines

Adam R. Nolan, anolan@ece.uc.edu, Bryan Everding, William Wee
Artificial Intelligence and Computer Vision Lab, University of Cincinnati, USA

Abstract

Defining an optimal schedule for arbitrary algorithms on a network of heterogeneous machines is an NP complete problem. This paper focuses on data parallel deterministic neighborhood computer vision algorithms. This focus enables the linear time definition of a schedule which minimizes the distributed execution time by overlapping computation and communication cycles on the network. The static scheduling model allows for any speed machine to participate in the concurrent computation but makes the assumption of a master/slave control mechanism using a linear communication network. We investigate the limitations of the static scheduling model based on statistical descriptions of the model parameters. Using statistical models, an approximation of the schedule length density function is derived. This statistical model is used to establish better approximations of schedule length.

Keywords: Computer Vision, Heterogeneous Scheduling, Distributed Algorithms, Nondeterminism

1. Introduction

In the past, many studies have been performed analyzing the capabilities of various parallel processor – vision algorithm mappings. Thorough surveys can be found in [17][3][2][4][5][16]. Most of these efforts focus on the mapping of a single machine to a single algorithm, or mapping a suite of algorithms to a single architecture [17]. Most of the conclusions made in these studies are based on the architectural similarities between the hardware communication configurations and the communication patterns inherent in the vision algorithm (i.e. vision tasks tend to have highly regular communication). Recent research efforts have discussed the mapping of computer vision tasks to networks of workstations with the assumption of homogeneous workstation clusters [11][7]. Additional efforts have focused on scheduling suites of independent programs onto networks of heterogeneous machines [10]. This paper relaxes the assumption of homogeneous workstation clusters and independent program suites. It focuses on the distribution of a single program on a set of architectures connected by the PVM message passing library [6][13]. A framework is presented for the polynomial time static scheduling of deterministic local communication algorithms onto a suite of heterogeneous machines using linear communication. In order to evaluate the effectiveness of the static schedule, a nondeterministic model is defined.

The paper is organized as follows. Section 2 presents the background and motivation. Section 3 develops the analytical description of the scheduling process and introduces a set of conditions necessary for the minimization of the execution time. Section 4 presents the scheduling algorithm. Section 5 presents several low level computer vision tasks and their corresponding scheduling models. Section 6 introduces a statistical model of the schedule. Section 7 demonstrates the use of the statistical model for predicting schedule length of a distributed convolution algorithm. Section 8 presents conclusions and future work.

2. Background

Efforts have been made to develop an Automatic Visual Inspection System (AVIS) for use with various aerospace engine components. These components include high pressure turbine blades (aircraft), injector baffles, oxidation posts and annulus rings (spacecraft). AVIS utilizes several scales of information abstracted from the original image with each scale requiring a set of low level vision operations [8][14]. The realization of the AVIS paradigm is limited by the tremendous computational burden of these low level vision operations. These algorithms include convolution, difference of Gaussian filtering, morphological filtering, Fourier transform, and Hough transform. In order to increase the speed of AVIS, distributed solutions were investigated. Defining an effective distribution onto the various machines available on the LAN requires models of algorithm decomposition, communication mechanisms, and machine speed for a given algorithm.

3. Schedule Model

Several assumptions are made in the definition of the schedule model. The communication time, execution time, and algorithm decomposition are assumed to be linear. The control structure is assumed to be master/slave with no interslave communication. Based on these assumptions, we construct:

$\mathcal{S}(n)$ – a scheduling of N homogeneous tasks onto n machines, Eq.3.1

where $N = \sum_{i=1}^{i=n} \eta_i$, with each slave machine(i) receiving a task size of η_i .

The time to send data to the slave(i) in the schedule is described by:

$$T_{s(i)} = T^0_{s(i)}(\text{overhead}) + T^{\eta_i}_{s(i)}(\text{transmit}) . \quad \text{Eq.3.2}$$

Similarly, the time to receive data from slave(i) is described by:

$$T_{r(i)} = T^0_{r(i)}(\text{overhead}) + T^{\eta_i}_{r(i)}(\text{transmit}) . \quad \text{Eq.3.3}$$

The computation time for slave(i) can also be described in a similar manner:

$$T_{c(i)} = T^{\eta_i}_{c(i)}(\text{compute } \eta_i) . \quad \text{Eq.3.4}$$

These parameters can be defined in terms of communication bandwidth β_i(secs/pixel), communication overhead, α_i(secs), and execution rate, γ_i(secs/operation) for each slave machine used in the schedule. Because the model is linear, the computation and communication times are proportional to their corresponding neighborhoods (data sizes), as described below.

$$T^1_{s(i)} = \phi_s \times \beta_i, \quad \text{Eq.3.5}$$

where ϕ_s is the neighborhood associated with sending a task size 1 to a slave(i) .

$$T^0_{s(i)} = \phi^0_s \times \beta_i + \alpha_i, \quad \text{Eq.3.6}$$

where ϕ^0_s is the send overhead neighborhood .

$$T^1_{r(i)} = \phi_r \times \beta_i, \quad \text{Eq.3.7}$$

where ϕ_r is the neighborhood associated with task size 1 received from slave(i) .

$$T^0_{r(i)} = \phi^0_r \times \beta_i + \alpha_i, \quad \text{Eq.3.8}$$

where ϕ^0_r is the receive overhead neighborhood .

$$T^1_{c(i)} = \phi_c \times \gamma_i, \quad \text{Eq.3.9}$$

where ϕ_c is the neighborhood associated with computing a single task on slave(i) .

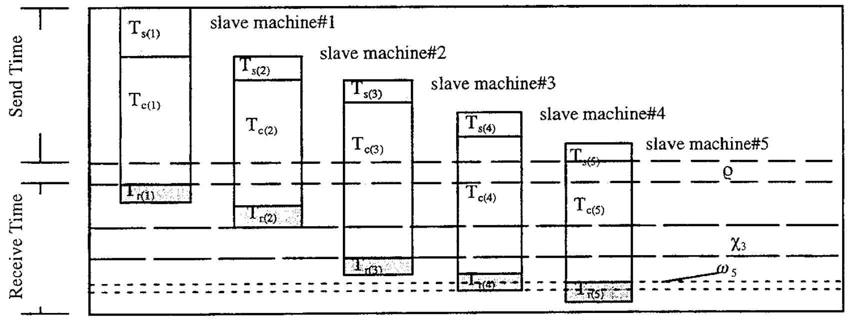

Fig.1 Depiction of Master/Slave Communication and Execution Model

As seen in Fig.1, the linear model enables a direct analysis of the optimality of a given schedule. The degree of nonconcurrency in the schedule is indicated by the values of ϱ, ω, and χ. The parameter ϱ is a measure of the time spent by the master machine waiting for the first processed data partition to be returned,

$$\varrho = T_{c(1)} - \sum_{i=2}^{n} T_{s(i)}. \qquad \text{Eq.3.10}$$

The parameter ω_i is a measure of slave wait time caused by contention with the previous slave machine for the linear communication channel,

$$\omega_i = \lfloor T_{c(i-1)} + T_{r(i-1)} - \left(T_{s(i)} + T_{c(i)} \right) \rfloor , \text{ where } \lfloor f \rfloor = \max (0, f). \qquad \text{Eq.3.11}$$

The parameter χ_i is a measure of the master wait time caused by excess computation of machine(i) after machine(i–1) has completed returning its data partition. The effects on the schedule time can be reduced by previous ω_j and ϱ values, i.e.

$$\chi_i = \lfloor T_{s(i)} + T_{c(i)} - T_{c(i-1)} - T_{r(i-1)} - \lambda_i \rfloor . \qquad \text{Eq.3.12}$$

The λ_i term represents the data queue waiting to be returned from the slave processes due to previous contention for the communication channel,

$$\lambda_i = \lfloor \lambda_{i-1} + \omega_{i-1} - \chi_{i-1} \rfloor,\ \lambda_1 = \lfloor -\varrho \rfloor . \qquad \text{Eq.3.13}$$

As one would expect, large values of ϱ, ω and χ result in suboptimal schedules. Typically, these values cannot be eliminated due to the constraint of integer task size. The total execution time for a given schedule is:

$$T(n) = \sum_{i=1}^{n} T_{s(i)} + \sum_{i=1}^{n} T_{r(i)} + \lfloor \varrho \rfloor + \sum_{i=2}^{n} \chi_i . \qquad \text{Eq.3.14}$$

In order to minimize this execution time, a set of criteria must be established to explicitly define the effects of the values ϱ, ω_i, and χ_i. In a previous paper [18], a set of conditions were derived which define these criteria. Based on these conditions, a linear time static scheduling algorithm was presented.

4. Scheduling Algorithm

As developed in[18], the minimum schedule length (Eq.3.14) for n machines can be found in which ω_i=0 and χ_i=0. This solution corresponds to $T_{c(i)} + T_{r(i)} = T_{c(i+1)} + T_{s(i+1)}$. Using this relation in conjunction with the definition of total task size, $\sum_{i=1}^{i=n} \eta_i = N$, the n values of η_i can be determined explicitly using Gaussian elimination on the n linear equations shown in Fig.2 .

$$
\begin{aligned}
&(1) \quad T_{c\,(1)} + T_{r(1)} = T_{c(2)} + T_{s(2)} \\
&\vdots \\
&(n-1) \quad T_{c\,(n-1)} + T_{r(n-1)} = T_{c(n)} + T_{s(n)} \\
&(n) \quad \sum_{i=1}^{n} \eta_i = N
\end{aligned}
\qquad \text{Eq.4.1}
$$

Fig.2 Set of linear equations defining schedule with $\chi_i = 0$, $\omega_i = 0$ for all i

Although no closed form solution for n exists, an appropriate number of slave machines can be determined by the evaluation of ϱ as described by conditions developed in[18]. Because the solution to Eq.4.1 is based on the deterministic values of γ_i, α_i and β_i, the static schedule is susceptible to any observed deviation of these parameters. An analysis of these nondeterministic effects will be presented in section 6.

5. Low Level Vision Modeling

The low level computer vision tasks needed for the AVIS computations are convolution, difference of Gaussian, Fourier transform, Hough transform, and morphological filtering operations. Although explication of these algorithms is beyond the scope of this paper, thorough discussions can be found in [1][4][5][12][15][16][17]. For the purpose of this study, the key algorithmic elements are contained in the model parameters, $\gamma_i, \phi_c, \phi_s, \phi_c, \phi_s^0$. These algorithm specific parameters are presented in Fig.3.

	Convolution	DoG	Morphological	2D FFT	Hough
γ_i	$\frac{\text{secs}}{\text{mult + add}}$	$\frac{\text{secs}}{\text{mult + add}}$	$\frac{\text{secs}}{\text{2logical Ops}}$	$\frac{\text{secs}}{\text{cmplx(mult + add)}}$	$\frac{\text{secs}}{\text{mult + trig\_lookup}}$
ϕ_c	M^2N	$(M_1^2 + M_2^2)N$	B N	$N\log_2 N$	#image pixels
ϕ_s	N	N	N	N	0
ϕ_r	N	N	N	N	1
ϕ_s^0	$(M^2 + (M-1)N)$	$(M_1^2 + M_2^2 + (M_1 - 1)N)$	$(B + (M-1)N)$	0	N^2
ϕ_r^0	0	0	0	0	0

where M_i is the size of the square kernel, N is the image size and B is the maximum size of the rectangular kernel.

Fig.3 Table of scheduling parameters for the low level vision algorithms

6. Statistical Properties

In order to proceed with a statistical analysis, it is necessary to make assumptions about the density distribution of the schedule parameters α_i, β_i and γ_i. Normal densities are assumed, and the sample means and variances of these parameters are used as unbiased estimates of the mean and variances. These means and variances are defined as:

μ_{α_i} = sample mean of communication overhead to slave machine i

σ_{α_i} = standard deviation of communication overhead to slave machine i

μ_{β_i} = sample mean of communication bandwidth to/from slave machine i

σ_{β_i} = standard deviation of communication bandwidth to/from slave machine i

μ_{γ_i} = sample mean of execution rate of slave machine i

σ_{γ_i} = standard deviation of execution rate of slave machine i

We represent the communication overhead as a random variable O_i, the communication bandwidth as random variable C_i, and the execution rate as random variable G_i. The density distributions for these random variables are given in Eq.6.1 .

$$f(O_i) = \frac{1}{\sigma_{\alpha_i}\sqrt{2\pi}}\, e^{-\frac{(x-\mu_{\alpha_i})^2}{2\sigma^2_{\alpha_i}}},\quad f(C_i) = \frac{1}{\sigma_{\beta_i}\sqrt{2\pi}}\, e^{-\frac{(x-\mu_{\beta_i})^2}{2\sigma^2_{\beta_i}}},\ f(G_i) = \frac{1}{\sigma_{\gamma_i}\sqrt{2\pi}}\, e^{-\frac{(x-\mu_{\gamma_i})^2}{2\sigma^2_{\gamma_i}}} \quad \text{Eq.6.1}$$

These variables are assumed to be normal and independent, hence any linear combination of them is also a normal random variable with variance and mean defined as:

$$\sigma^2_{\sum_{i=1}^{n} c_i x_i} = \text{variance}\left[\sum_{i=1}^{n} c_i x_i\right] = \sum_{i=1}^{n} c_i^2 \sigma_i^2 \ ,\ \mu_{\sum_{i=1}^{n} c_i x_i} = \text{mean}\left[\sum_{i=1}^{n} c_i x_i\right] = \sum_{i=1}^{n} c_i \mu_i \ ,\text{Eq.6.2}$$

where x_i is a normal independent random variable and c_i is a linear multiplier .

Representing Eq.3.14 in terms of these random variables is complicated by the parameter λ_i (as defined in Eq.3.13). Assuming that $\lambda_i = 0$, an upper approximation of T(n) can be defined:

$$T(n) \approx \sum_{i=1}^{n} T_{s(i)} + \sum_{i=1}^{n} T_{r(i)} + \lfloor \varrho \rfloor + \sum_{i=2}^{n} \lfloor T_{s(i)} + T_{c(i)} - T_{c(i-1)} - T_{r(i-1)} \rfloor \ . \text{Eq.6.3}$$

Representing the observed time, T(n), as a random variable Z, we examine the upper approximation of Z in terms of the random variables W, X, and Y_i.

$$Z \approx W + \lfloor X \rfloor + \sum_{i=2}^{n} \lfloor Y_i \rfloor \ . \quad \text{Eq.6.4}$$

These random variables are defined as combinations of the random variables representing the start-up overhead, communication bandwidth, and execution rates of the slave machines, O_i, C_i, and G_i, respectively. The definition for W and its corresponding density is given in Eq.6.5 . The definition for X and its corresponding density is given in Eq.6.6 , and the definition of Y_i and its corresponding density is given in Eq.6.7 .

$$W = \sum_{i=1}^{n}\Big(O_i + \big(\eta_i \phi_s + \phi_s^0\big) C_i \Big) + \sum_{i=1}^{n}\Big(O_i + \big(\eta_i \phi_r + \phi_r^0\big) C_i \Big) \quad \text{Eq.6.5}$$

$$f(W) = \frac{1}{\sigma_W\sqrt{2\pi}}\, e^{-\frac{(x-\mu_W)^2}{2\sigma^2_W}} \ ,\ \text{where } \mu_W,\ \sigma_W \text{ are given by Eq.6.2 .}$$

$$X = (\eta_1\ \phi_c)\ G_1 - \sum_{i=2}^{n}\Big(O_i + (\eta_i\ \phi_s + \phi_s^0)\ C_i \Big) \quad \text{Eq.6.6}$$

$$f(X) = \frac{1}{\sigma_X\sqrt{2\pi}}\, e^{-\frac{(x-\mu_X)^2}{2\sigma^2_X}} \ ,\ \text{where } \mu_X\ ,\ \sigma_X \text{ are given by Eq.6.2 .}$$

$$Y_i = (\eta_i\phi_c)G_i - (\eta_{i-1}\ \phi_c)G_{i-1} + \left(O_i + (\eta_i\phi_s + \phi_s^0)C_i\right) - \left(O_{i-1} + (\eta_{i-1}\phi_r + \phi_r^0)\ C_i\right),$$

$$f(Y_i) = \frac{1}{\sigma_{Yi}\sqrt{2\pi}}\ e^{-\frac{(x-\mu_{Yi})^2}{2\sigma_{Yi}^2}}, \qquad \text{Eq.6.7}$$

where μ_{Yi}, σ_{Y_i} are given by Eq.6.2.

As described in Eq.6.5, Eq.6.6, and Eq.6.7, the density functions for W, X, and Y are normal. The mean for W is strictly greater than zero, whereas the means for X and Y_i can be greater or less than zero. If the X and Y_i random variables were not truncated at zero as indicated in Eq.6.4, the composite density function for the upper bound of Z would be a simple summation of normal variables. The truncated normal density is defined with a delta function at the origin as described in Eq.6.8.

$$\lfloor X \rfloor = \begin{cases} X & \text{if } X \geq 0 \\ 0 & \text{if } X < 0 \end{cases}, \quad f(\lfloor X \rfloor) = \begin{cases} \theta_X\,\delta(x) + \frac{1}{\sigma_X\sqrt{2\pi}}\ e^{-\frac{(x-\mu_X)^2}{2\sigma_X^2}} & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \qquad \text{Eq.6.8}$$

where $\theta_X = \int_{-\infty}^{0} \frac{1}{\sigma_X\sqrt{2\pi}}\ e^{-\frac{(x-\mu_X)^2}{2\sigma_X^2}}dx$, and $\delta(x)$ is the Dirac delta function.

The inclusion of the truncated normal curve into the composite density function makes the definition of f(Z) more difficult. Assuming the three random variables, W, $\lfloor X \rfloor$, $\lfloor Y_i \rfloor$ are independent, we can define the density of their sum as the convolution of their densities [19]. As such, the density of the upper bound of Z can be defined as:

$$f_z(Z) = \left(\ f(W) * f(\lfloor X \rfloor)\right) * f\left(\lfloor Y_2 \rfloor\right) * f\left(\lfloor Y_3 \rfloor\right) \ldots * f\left(\lfloor Y_n \rfloor\right), \qquad \text{Eq.6.9}$$

where $f(W) * f\left(\lfloor X \rfloor\right) = \int_{-\infty}^{\infty} f_W(Z - X) f_{\lfloor X \rfloor}(X)dX$.

The difficulty introduced by the truncated normal curve is based on the integral of Eq.6.9. The limits of integration are redefined as:

$$f(W) * f\left(\lfloor X \rfloor\right) = \int_{0}^{\infty} f_W(Z - X)\left[\theta_X\,\delta(X) + \frac{1}{\sigma_X\sqrt{2\pi}}\ e^{-\frac{(x-\mu_X)^2}{2\sigma_X^2}}\right]dX. \qquad \text{Eq.6.10}$$

This integral has no closed form solution. Numerical solutions, however, offer a simple solution, and can quickly define a numerical density for the lower bound of random variable Z. After this density is defined, numerical integration can be performed on f(Z) to define confidence intervals on the predicted schedule time.

7. Experiments

As explained in section 6, an approximate density function for the schedule time can be defined using Eq.6.9. Given a schedule derived using Eq.4.1, the probability that the schedule will exhibit an observed speedup can be obtained. To demonstrate the use of this information two disjoint slave sets were arbitrarily defined. These two slave sets were used to examine the correlation between the observed schedule times and those predicted by the deterministic and nondeterministic models.

The schedule length density approximations for slave set 1 and slave set 2 are depicted below in Fig.4 . In addition to the density, the observed schedule time (averaged over 30 trials), the predicted deterministic model time and the density means are presented. These values are summarized below in Fig.5 for several kernel sizes of the convolution algorithm. In general the observed schedule times were greater than that predicted by the deterministic model and less than the nondeterministic mean. An exception to this observation can be found in the 7x7 convolution for slave set 2 in Fig.5.

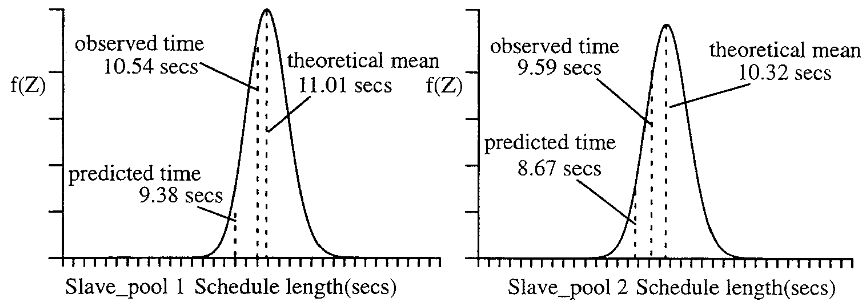

Fig.4 Density approximations of two disjoint slave pools for 17x17 convolution

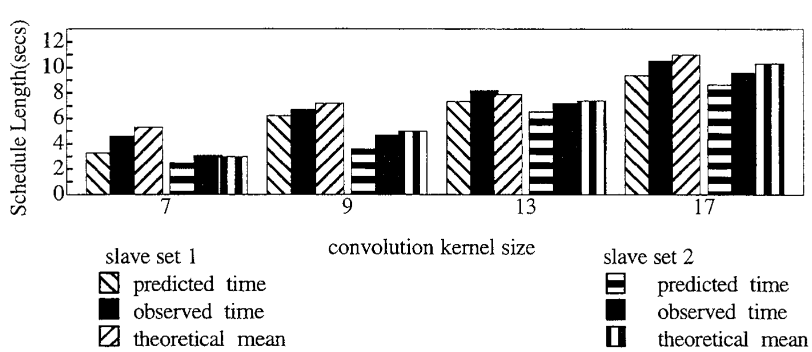

Fig.5 Predicted time, observed time and theoretical mean for slave sets 1&2

8. Conclusions

Focusing on linearly partitionable problems enables the definition of optimal scheduling conditions. These conditions enable a schedule to be defined in linear time. These conditions, however, are based on the assumption of deterministic execution and communication rates. As machines and networks exhibit nondeterministic behavior, this static model exhibits schedule times that deviate from the deterministically predicted values. The nondeterministic model enables an additional prediction of schedule behavior based on network variance. This nondeterministic prediction better describes the average observed schedule length than the deterministic prediction. The ability of the statistical model to accurately depict observed variance is currently under investigation. If It is hoped that future extensions of the scheduling model will incorporate the nondeterministic model into the algorithm decomposition method.

Acknowledgements

The authors would like to thank the reviewers for their time and helpful suggestions and also the University of Cincinnati Space Engineering Research Center for support of this project.

References

[1] Dana H. Ballard and Christopher M. Brown. *Computer Vision.* Prentice–Hall, Englewood Cliffs, New Jersey, 1982.

[2] Vipin Chaudhary and J. K. Aggarwal. *Parallel Algorithms for Machine Intelligence and Vision.* Vipin Kumar, Editor, Parallelism in Computer vision: A Review, Springer–Verlag, NY, 1990.

[3] Alok N. Choudhary and Janak H. Patel. Parallel Architectures and Parallel Algorithms for Integrated Vision Systems. *The Kluwer International Series In Engineering and Computer Science*, Kluwer Academic Publishers Boston, 1990.

[4] Robert Cypher and George L. C. Sanz. SIMD Architectures and Algorithms for Image Processing and Computer Vision. *IEEE Trans on ASSP*, 37(12), Dec. 1989.

[5] R. V. Dantu, N. J. Dimopoulos, K. R. Li, R. V. Patel, and A. J. Al–Khalili. Parallel Algorithms for Low Level Vision on the Homogeneous Multiprocessor. *Computers Elect. Eng*, 20(1): 51–60, 1994.

[6] Jack Dongarra, *PVM3.3 Users Guide*, Oak Ridge National Laboratories, 1994.

[7] C.C. Douglas, T.G. Mattson, and M.H. SchultzParallel Programming Systems For Workstation Clusters, *Yale University Department of Computer Science Research Report*, YALEU/DCS/TR–975, August, 1993.

[8] Bryan S. Everding, Adam R. Nolan, and William G. Wee. Generalization of an Automated Visual Inspection System(AVIS). *SAE Aerospace Atlantic Conference*, April 1994.

[9] Richard F. Freund and Howard Jay Siegel. Heterogeneous Processing. *Computer*, p. 13, June 1993.

[10] Richard F. Freund. The Challenges of Heterogeneous Computing. *Proceedings of the Parallel Systems Fair*, Cancun, April 1994.

[11] Chi–kin Lee and Mounir Hamdi. Efficient Parallel Image Processing Applications On a Network Of Distributed Workstations. *Proceedings of the Parallel Systems Fair*, Cancun, April 1994.

[12] David Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information.* W.H Freeman Press, San Francisco, 1982.

[13] Alfonso Matrone, Pasquale Schiano, and Vittorio Puoti. LINDA and PVM: A comparison between two environments for parallel programming. *Parallel Computing*, 19: 949–957, 1993.

[14] Bryan S. Everding, Adam R. Nolan, and William G. Wee. Performance Issues of an Automated Visual Inspection System(AVIS). *AIAA Jet Propulsion Conference*, June 1994

[15] A. V. Oppenheim and R. W. Schafer. *Digital Signal Processing.* Englewood Cliffs, NJ, Prentice–Hall, 1975.

[16] Quentin Stout. Mapping Vision Algorithms to Parallel Architectures. *Proceeding of the IEEE*, 76(8), August 1988.

[17] Charles Weems, Edward Riseman, Allen Hanson, and Azriel Rosenfeld. DARPA Image Understanding Benchmark for Parallel Computers. *Journal of Parallel and Distributed Computing*, 11(1): 1–24, Jan. 1991.

[18] Adam R. Nolan and Bryan S. Everding. Polynomial Time Scheduling of Low Level Vision Algorithms on Networks of Heterogeneous Machines, *EUROPAR*, August 1995.

[19] Athanasios Papoulis. *Probability, Random Variables, and Stochastic Processes.* McGraw–Hill, p. 135, 1984.