1986

The integration of a vision system with a CNC controller to perform closed loop machining /

David Samuel Hanan
Lehigh University

Recommended Citation

Hanan, David Samuel, "The integration of a vision system with a CNC controller to perform closed loop machining /" (1986). *Theses and Dissertations*. 4630.
https://preserve.lehigh.edu/etd/4630

The Integration of a Vision System with a CNC Controller to Perform Closed Loop Machining

by

David Samuel Hanan

A Thesis
Presented to the Graduate Committee
of Lehigh University
in Candidacy for the Degree of
Master of Science
in
Manufacturing Systems Engineering

Lehigh University
1986

Certificate of Approval

This thesis is accepted and approved in partial fulfillment of the requirements for the degree of Master of Science.

2/18/86
(date)

Emory W. Zimmers Jr.
Professor in Charge

Roger N. Nagel
Director of Manufacturing
Systems Engineering

E. Kane
Chairman of the Industrial
Engineering Department

Acknowlegdements

I would to thank and acknowledge certain people who aided me in the completion of this thesis. I would like to thank Hall Weaver for his advice and expertise in NC machining practices. I would like to thank Ken Wagner for his help in programming the GE 2000 CNC Controller. I would also like to thank Frank Bracken for his technical advise and interest in my research. Lastly, I would especially like to thank my advisor, Dr. Emory Zimmers, and the director of the Manufacturing Systems Engineering Program, Dr. Roger Nagel, for their support throughout the completion of my masters degree.

Table of Contents

List of Tables

List of Figures

I. Abstract

This thesis investigates the use of machine vision to provide a CNC controller with closed loop machining functions that would enable the machine tool to operate in an unattended mode. These closed loop machining functions compensate for tool wear, variable part location, and variable raw material size. This is accomplished by incorporating processed visual sensor data into the logic of the NC part program.

To demonstrate the feasibility of this system, a General Electric Optomation II (*) vision system was connected to a Mark Century 2000 CNC (*) controller in the Manufacturing Lab at Lehigh University. The software that enables vision system and CNC controller integration was designed and successfully tested.

In addition, this thesis discusses the required specifications for a vision system to operate in a manufacturing work station within a computer integrated manufacturing setting. This macroscopic view creates a full picture of a vision system performing closed loop machining functions for an unattended work station. From this design and a study of the front end components of a

(*)-Optomation II and Mark Century 2000 are registered trademarks of the General Electric Company.

vision system, necessary developments required for vision systems to provide complete automation of work station functions were identified. These included increased camera pixel array size, faster processing capabilities, the use of gray scale algorithms, and bimodal communication capabilities to allow for batch and real time communication.

II. Introduction

A. Automated Location and Inspection Strategies

An observable trend in manufacturing is the development of flexible, unattended machine cells. These cells usually consist of automated material handling devices feeding one or more CNC (Computer Numerical Control) machines, part cleaning and deburring devices, etc. The cell requires some form of intelligent control and metrology devices to monitor and control quality and variable factors inherent in the manufacturing process. Strategies for accommodating the automation of the control and metrology functions within the cell are in the late stages of development and early stages of availability. As new technology is further developed, new applications to solve the current information needs are under rapid development as application specifications become better defined.

The potential for machine vision systems to provide metrology automation and control functions for a machine cell is increasing due to the rapidly developing sophistication of the software required for system intelligence and the hardware needed for faster processing speeds. This thesis explores the use of machine vision to provide a CNC machine controller with non-real time feedback information for location, metrology, and tool offset control.

B. Closed Loop Machining

Closed loop machining (CLM) as defined here and by a controller manufacturer [1] is a non-real time feedback mechanism that allows the user to adjust for certain variables in the machining process. They are:

1) part location.
2) part orientation.
3) tool wear.
4) raw material size variability.

Non-real time control means that all adjustments are made while the machine is not in a processing mode. Contact probes and non-contact optical sensors are examples of sensors used in this type of control scheme. Real time closed loop control schemes fall into the category of adaptive control and involve making adjustments according to continuously monitored machining variables linked to the cutting process. Monitoring the horsepower of the tool as an indication of tool wear is an example of adaptive real time control.[2] Information from the measurement and adjustments should be used while the work piece is in the work station for maximum efficiency. Later use will require additional effort for the information to be useful.

Contact closed loop machining routines are currently available on CNC machine controllers using touch sensor probes. The probe resides either in the machine spindle, at a remote location from the table, or in its own locomotion device such as a gantry or other type robot. The probe contacts the surfaces to be measured enough times to establish the geometrical surface it is measuring (eg. three points to define a plane). The contact stylus can be of many different geometrical configurations to allow measurement of a variety of hard to get at locations. The stylus triggers an electrical switch that sends a pulse to the controller when the contacts are closed. The trigger device can also be stationary and used to set tool offsets. This is accomplished by allowing the tool in the spindle to trigger the stationary probe. [1]

Contact closed loop machining has provided a major step toward automated control of a machine cell but has these disadvantages :

1) Probing is time consuming because it requires:
    a. calibration prior to each probing cycle.
    b. multiple contacts of the same surface.
    c. caution when near the work piece so as not to damage the trigger mechanism.
2) Wired probes are difficult to set up for automated use for "on table" measurements.

3) Radio signal probes are subject to shop floor noise which can seriously destroy the integrity of the communicated data causing incorrect data to be transmitted. [3][4]

Non-contact closed loop machining uses visual sensors to provide locational and/or dimensional information to the CNC machine controller. These visual sensor systems rely on electrical and software systems to provide sensory data to the controller and therefore, are usually faster in processing speed than probes. They also allow for the measurement of surfaces that face contamination or destruction if touched by a probe or other measuring device.

C. Thesis Overview

Machine vision makes it feasible to automate the four manual functions of CNC machine tool work stations so that they can operate in an unattended mode. The four functions are:

1) alignment of machine and part coordinate systems.
2) part program adjustment for raw material size variance.
3) pre-finish cut tool offset adjustments.
4) tool wear offset adjustments.

The automation of these functions using machine vision requires the following:

1) a study of the current state of vision system components to determine:
    a. the specifications for close tolerance metrology.
    b. the specifications required to integrate the vision system into the manufacturing system.
2) the design of the vision system software to perform the closed loop machining functions.
3) the design of the controller software that would enable it to communicate with the vision system.
4) the design of the mode and method of using the visual sensor closed loop machining information in the part program to make the actual adjustments.

The manual operation of the closed loop machining tasks has been described in detail to provide background information on the functions to be automated. Because machine vision is still a developing technology, the technical operations of each component of a vision system are described in terms of how they function, the current limitations of available equipment, and what is being developed for the future. Because CNC machine control is a comparatively mature technology, the technical background on controllers will concentrate on providing information on how the Mark

Century 2000 CNC controller software operates.

The solution methodology to the work station location and inspection automation problem is presented in two sections. The first section describes the ideal system design specifications required if a vision system is to:

1) operate in an integrated factory computer architecture through hierarchical control.

2) perform the close tolerance metrology required in a machining center.

The second section describes the design of the software used to demonstrate the processing of closed loop machining information from a vision system to a CNC controller in the Manufacturing Lab at Lehigh University.

III. Problem Definition: Description of Current Manual Operations

The machining practices that will be described below normally require the use of:

1) fixed setups.

2) manually operated measuring tools such as micrometers, gages, and inspection fixtures.

3) manual calculations and adjustment setting.

In order for a CNC machine tool to produce parts of high quality with close tolerances in an unattended mode, the CLM system will have to replace these manual operations and make adjustments for variables introduced by the automated process. It is assumed that the vision system has the resolution capabilities required to perform the tasks. This may not be true in all cases and will be discussed in detail later.

The following NC machining practices are based on my past experiences as a manufacturing engineer. These application areas must be automated for the machine cell to operate in an unattended mode:

A. Aligning the Machine and Part Coordinate Systems

NC part programmers usually base their part programs on a local coordinate system whose origin is located on the work piece. This allows the part program to be independent of the placement of the part on the machining area. The machine controller must be taught where the part programmer has determined the part's origin to be which is also referred to as the program zero point or "program zero". All movement commands are relative to program zero when in the absolute positioning mode. Program zero is used as a starting point for programs when in the incremental positioning mode.

The task of locating program zero is usually performed by the machinist who switches the NC machine to a teaching mode and contacts the tool ("touches off") with the work piece at the program zero location. If the subsequent parts are not located at the identical spot on the machine table (no permanent fixture used), the machine controller must be taught the new location of program zero for each part. Care must also be taken to align the part with the machine's cutting axes. When permanent fixtures and setups are used, setup offsets and fixture offsets manually entered into the controller can be used to locate program zero.

B. Adjusting for Raw Material Size Variation

Because the initial dimensions of rough castings vary from part to part, roughing cuts must be programmed to account for the worst case of oversize of a dimension. The size of raw material stock delivered to a work station, especially in job shops, is often a function of the material sizes available rather than the material called for in the design specification. This may result in machining the work piece to the size of the raw material specified in the design prior to running the NC program.

Programmers, when possible, measure the rough castings and raw materials to determine where to start their initial machining cuts. They also can program into the part program optional roughing cut blocks whereby the machinist has the option, after measuring the raw material or casting, to skip blocks where appropriate. When this programming option is not used, the part programmer will usually program the tool to start its roughing cuts well outside of the part's anticipated worst case surface dimension and gradually move the tool closer to the surface. This avoids tool breakage and potential damage to the work piece and the machine. However, these programming practices often lead to inefficient machining (i.e. passes at the work piece where little or no metal is removed).

C. Pre-Final Cut Offset Adjustments

For close tolerance parts where variancing the work piece is extremely expensive and/or critical to a production schedule, it is desirable to machine with caution so as to avoid costly machining errors. Parts that would fall into this category have one or more of the following characteristics:

1) small lot sizes with little or no excess raw material stock available.

2) expensive raw material.

3) large WIP (work in process) costs already invested in the work piece.

Variancing and/or scrapping the part may result in:

1) missing important delivery dates.

2) the cost of additional setups and rework.

3) rescheduling the part which may interfere with other jobs in the shop depending on the manufacturer's workload.

For new programs, tool changes, new setups, and the machining of unusually hard materials at close tolerances, a common NC machining practice is to offset the tool from the work piece's final intended cutter path. After the trial offset cut has been made:

1) The machinist measures the difference between the programmed path and the actual cutter path to determine how the tool and material is cutting.

2) The machinist calculates the offset required to cut the piece to size.

3) The results are entered into the tool offset tables in the controller.

4) The finish cut program blocks in the part program are then rerun to add the offset information obtained

in the trial run.

Note that four manual steps are required.

D. Tool Wear Offset Adjustments

After the final machining steps have been performed on the work piece, it is inspected by the machinist for the following four possible conditions:

1) The piece is within tolerance, and no adjustments are required.
2) The piece is acceptable but at the edge of the tolerance band, and therefore, adjustments are required.
3) The piece is out of tolerance and can be reworked. Adjustments to the tool are required, and the appropriate finish cuts must be made while the work piece is in the same machine setup.
4) The piece is out of tolerance and must be varianced and/or scrapped. It cannot be reworked while in this setup. Adjustments to the tool are required.

Adjustments take the form of:

1) increasing or decreasing the tool offsets.
2) replacing the tool and resetting the offsets.

To summarize the problem of automating these machinist tasks using machine vision:

1) The vision system must replace the physical measurement inspections performed on the work piece.

2) The vision system and CNC controller software logic must then apply the logic that a machinist uses to analyze and adjust the process.

IV. Technical Background

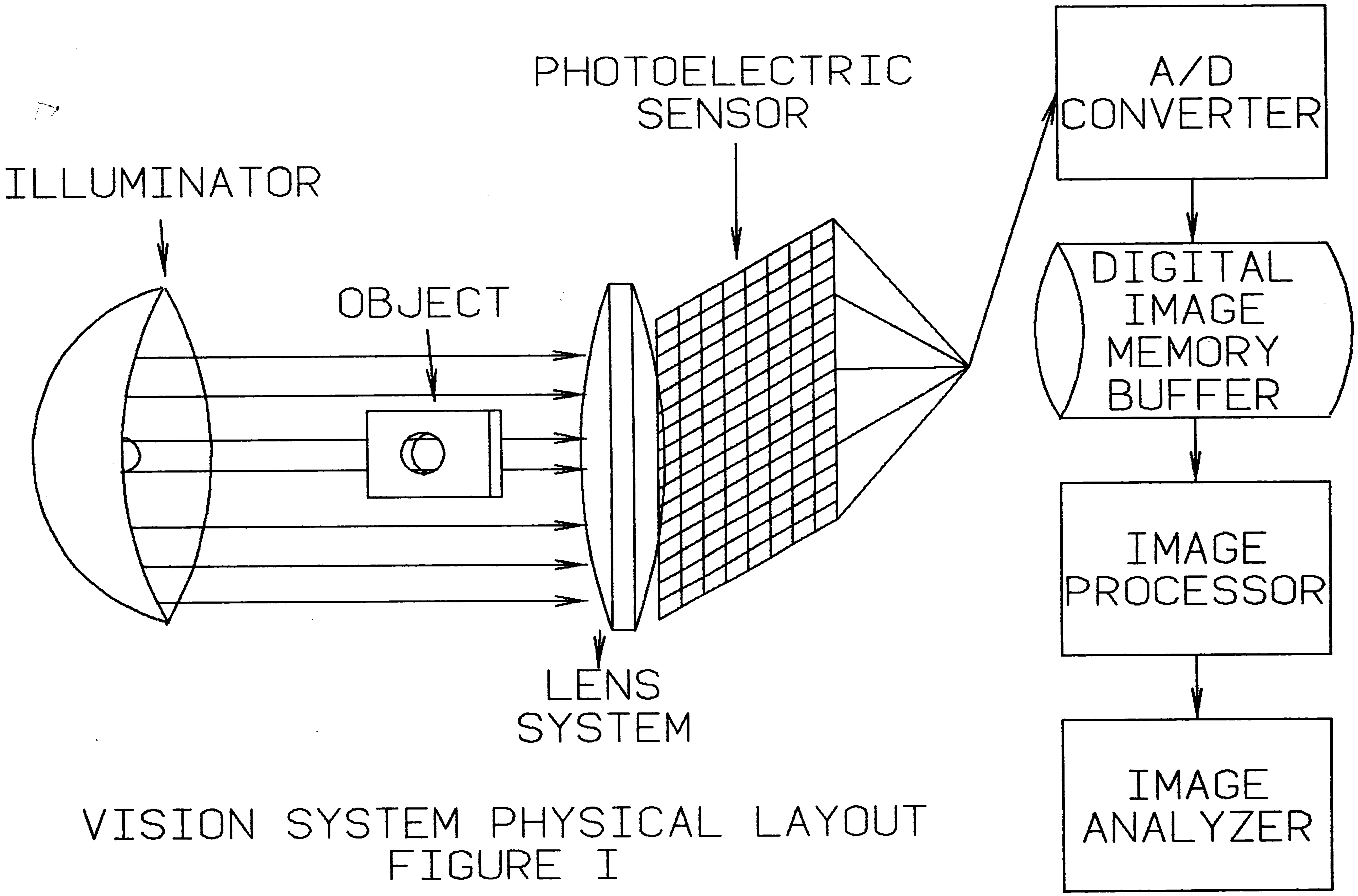
A. Vision Systems

By connecting a computer to an optical sensor, the sensor's usefulness increases. A vision system can be thought of as a grid of optical sensors connected to a computer so that the high level tasks of inspection, identification, metrology, and guidance can be performed.

1. Overall System Operation

As shown in Figure I, light radiating from an object is transformed through a lens system onto a photosensitive grid of the optical sensor. The light intensity radiated on each element (called pixel for picture element) of the grid is digitized (A/D or analog to digital signal conversion) according to a predetermined light intensity scale known as the gray scale. The two most prevalent types of vision system data processing schemes are binary thresholding and gray scale processing. [5][6][7]

In binary thresholding vision systems, the light intensity of each element in the grid is compared to a set light threshold intensity value. If the pixel value is above the threshold, it is represented by a one and below by a zero. The result is a binary representation of the scene where the background is separated from the object. The scene is further analyzed and the amount of data further reduced so that the data base representing the



VISION SYSTEM PHYSICAL LAYOUT
FIGURE I

scene contains mostly basic feature elements. These feature elements are the corner points, areas, and centroids of each item in the scene. A high level language can then transform and manipulate the feature data into usable information.

Gray scale vision systems operate on the original gray scale data as initially extracted which means using a six or eight bit value to represent each pixel instead of the one bit pixel representation of binary thresholding systems. These systems determine edges by identifying changes in light intensity rather than changes in state of a single bit. The disadvantage of gray scale systems is that they require much more data processing power to process the visual data at realistic speeds.

Other optical sensors used for metrology and location include laser micrometers and structured light. These focus only on a one dimensional slice across a scene to provide depth, width, or height information. Structured light can also provide shape information.

The ability to analyze images depends on the correct use of the front end components of a vision system. A brief description of the components that combine to form a vision system is provided in order to understand how the vision system will be used for closed loop machining applications and the limitations of using machine vision.

2. Field of View Illumination

The goal of illumination schemes is to make the task of interpreting the scene as easy as possible. Ideal lighting conditions would permit:

1) high light intensity contrast between the object and background.

2) minimum spectral or reflected light.

3) minimal shadows unless intended for contrast. [6]

The effects of ambient light source contributions, such as sunlight, should be eliminated. This reduces the variability of the primary light source.

Back lighting schemes where the light source is behind an object which rests on a translucent surface, provide a silhouetted image in which edges are well defined. This setup usually provides the best measurement conditions. The setup requirement, locating the part on a transparent or translucent surface, is usually not practical or possible for in-line industrial applications. Therefore, front lighting schemes are the most prevalent form of illumination in industrial settings. Front lighting sources range from incandescents and fluorescents to gas strobes and halogen lamps, each having different operating characteristics satisfying special lighting application requirements.[8] Light sensor detectors are usually matched to the light source wave length that produces the best response.

An additional crucial condition for successful interpretation of a scene is that the dispersion of light onto the viewing surface be uniform. This aids in edge detection. Light reflected back to the camera on the same surface should have equal intensity across the surface for uniform triggering of the photo sensor at each part feature. Since most machined surfaces have spectral surfaces, provisions should be made to reduce the effect of reflections on the object's surface. Because spectral light travels in parallel or plane polarized waves, spectral reflections can be removed by controlling the polarization of the light entering the camera lens so that only diffuse or randomly oriented light waves penetrate the lens.[9] A polarizer lens is required in front of the light source to orient the outgoing rays. An analyzer lens oriented perpendicular to the polarizer in front of the lens will screen out spectral rays allowing only the diffuse light through.

Front lighting schemes for binary thresholding vision systems are dependent on distinct, contrasting light intensities, and the user is limited to only two intensity levels. In most applications, the distinction between light intensities between edges is not very great. A system that could recognize edges of a low contrast image would be extremely powerful. Gray scale systems locate edges by monitoring changes in light intensities. They are less sensitive to:

1) overall lighting changes.
2) lack of light intensity contrast.
3) different light intensities per feature in the scene.
4) part discoloration.
5) the effects of oil and cutting fluid in creating different reflected intensities. [7]

Most gray scale systems use template matching techniques for non-metrological tasks such as part recognition and feature identification. The large data processing requirements that would enable the use of a gray scale system to perform close tolerance metrological work requires further development before becoming a cost effective alternative, given the current state of commercially available technology.

3. Lens Systems

The lens system transfers the light intensities from object space onto the image space consisting of light sensors. For close tolerance metrological applications, the aberrations and other distortions inherent in lens systems degrade the stated or nominal resolution of the image. Aberrations in the lens system occur when the light rays reflected from one point in the object space fail to converge to a single focus on the image plane of the camera. These degradations occur when:

1) Imperfections exist in the shape of lens surface.

2) Extension tubes (described below) are used, especially when used with lenses they were not designed for.

3) The image is out of focus (the feature end points are on two different planes or are simply out of focus).

4) The f-stop and/or aperture settings require adjustment. [5]

Extension tubes allow a camera with a standard lens to be placed further from the object plane while maintaining the same field of view. It has the effect of magnifying the scene so that the camera can be further from the process it is measuring, e.g. further from a machine tool casting off cutting fluid and heated metal chips.

Another factor that degrades resolution is that real lens transformations of light intensity decrease radially from the concentric axis of the lens. The gradual to dramatic drop off in light intensity from the center of the image field to the edge degrades the resolution of the binary thresholding edge detection system.[5][10] This can result from misfocusing, misapplication of extension tubes, stray light, and spherical and chromatic aberrations. It can be corrected by:

1) smaller aperture settings.
2) narrow band pass light filters.
3) light shields.[5]

Misfocusing has the potential to reduce the vision system's resolution by reducing the edge light intensity contrast required for binary thresholded edge detection.[5] When two points lie on different focal planes, the possibility that only one point will be in focus exists. There are numerous ways to reduce the effects of this phenomenon, the most obvious of which is to select an optimal depth of field setting for a well defined class of measurements.

Other systems have solid state auto focus circuitry that can be used to measure a third dimension for two dimensional vision system cameras. With advanced software, the auto focus function can be used to measure horizontal distances of points lying on different vertical focal planes.

For non-binary image processing systems where light intensity contrast of the edges is not crucial to maintaining system resolution, the consequences of misfocusing are more forgiving. For instance, if the directional derivative of the change in light intensity is taken across an item (spatial derivative technique), the derivative maxima would represent an accurate location of the edges.[5]

Geometric pin cushion and barrel distortions occur when parallel lines in the object scene appear bowed inward or outward in the image scene. This usually is associated with the misapplication of extension tubes or an incorrect f-stop setting on the camera. [10]

4. Photo Electric Sensors

Two types of sensors are used today to record light intensity information after passing through the lens system so that the scene can be analyzed. They are:

(1) vacuum tube cameras such as the Videcon camera.

(2) solid state cameras such as charged coupled devices (CCD) and charge injected devices (CID).

Vacuum tube cameras are the older of the two technologies. The image from a lens is projected onto a face plate consisting of two layers of material. The first layer consists of a transparent signal electrode film lying on the inner surface of the plate. The second layer consists of a dense array of photo sensitive cells which are called pixels (short for picture elements). The pixels generate a charge in response to light whose intensity is a function of the charges accumulated over the pixel. The pixels are scanned by an electron beam which deposits enough electrons on each pixel to neutralize the charge left by the light. This causes a current to flow which is proportional to the light intensity and the amount of time with which the area was scanned. This charge array information is then processed further and analyzed. The typical image frame generation rate of these vacuum tube systems is 30 frames per second.[6][10]

Both CCD and CID solid state cameras make use of arrays of photo sensitive semiconductor devices (usually silicon substrates) that are integrated into a single chip. The basic sensor mechanism is an integrating photon detector consisting of a MOS (metal oxide semiconductor) capacitor biased above threshold. The change in the charge on the capacitor is measured by integrating the current values. These values are transferred to output registers for conversion and processing.[6][11] As compared to vacuum tubes, CCD and CID systems:

1) capture 60 video frames per second (twice the speed of vacuum tube cameras).

2) are free from drift and other non-linear distortions inherent in vacuum tubes.

3) are insensitive to magnetic fields.

4) are unaffected by shock.

Also, the solid state camera's sensor sensitivity does not deteriorate over time. [11]

5. Vision Data Processing Scheme

After the visual image has passed through the lens system and has been processed by the camera, the visual data from the sensor must be processed and analyzed to provide useful information. The initial processing is accomplished with hardware for speed and routine data reduction. Further processing is completed with application software to handle more complex and varying analyses.

There are a variety of ways to process incoming vision data. New methods of processing such as pipeline processing schemes are under development and rapidly entering the market place. They often feature:

a) the capability to handle increasing amounts of data at increased processing speeds.

b) the capability to perform sophisticated enhancements and interpretations on the data, also at increased processing speeds.

The system that will be described is a microprocessor based binary thresholded image vision system (General Electric Optomation II). It uses a CID solid state camera for vision sensory input and a corner point algorithm for edge detection and item identification.

This vision processing scheme generally has three phases (see Figure II):

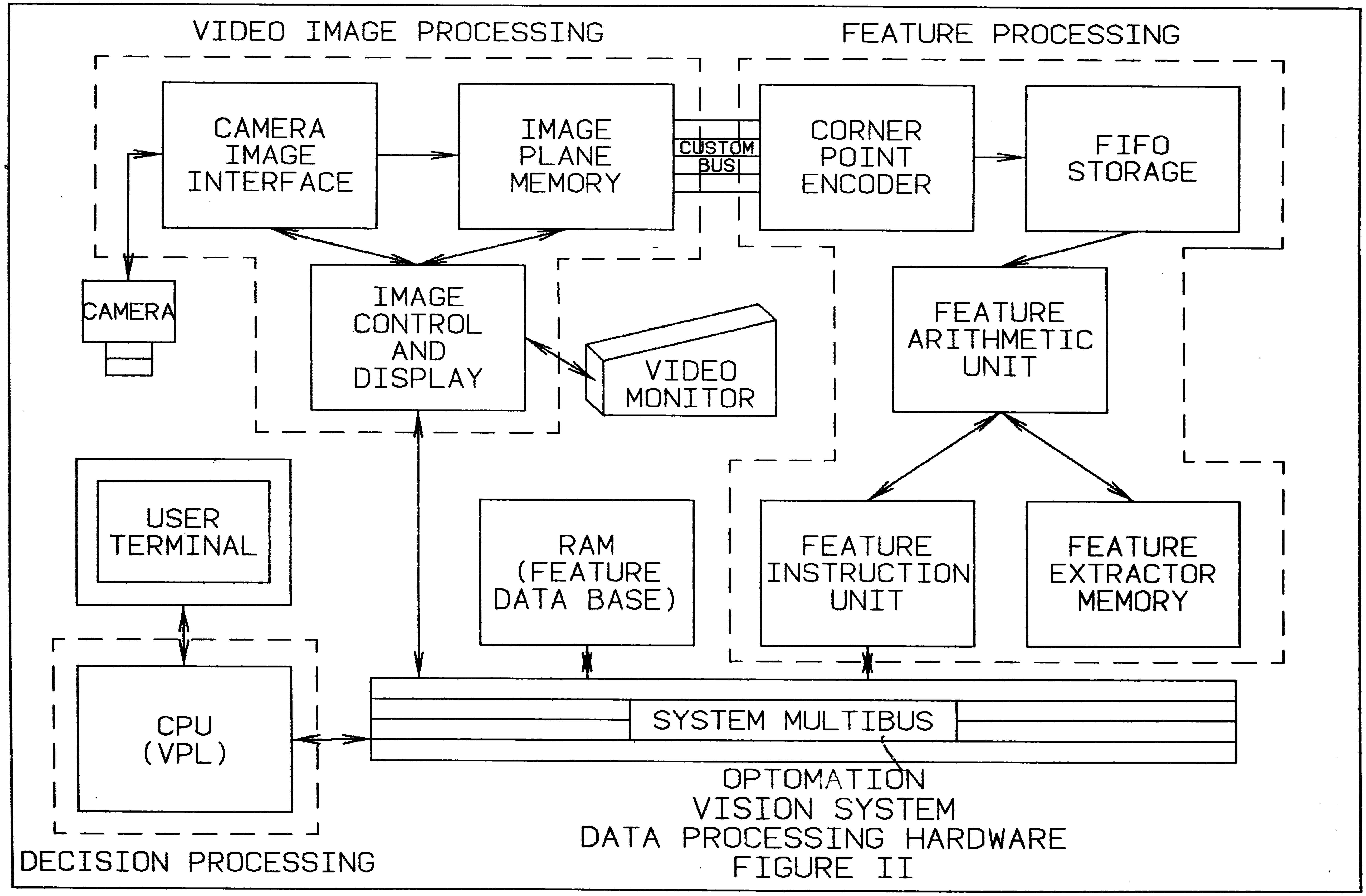
(1) Video Image Processing or Preprocessing which interfaces directly with the video source.

(2) Feature Processing or Primary Processing which transforms data into primary feature data.

(3) Decision Processing or Secondary Processing which is controlled by user application programs and uses primary feature data to create more complex feature data. [12]

The major goal of the hardware structure of the system is to continually reduce the amount of incoming data to a size that can be processed rapidly and still retain enough meaningful data to be used for analysis. The microprocessors in the system split these data refinement tasks and convert the incoming low level vision data into high level object feature data to be used in application software.

Preprocessing and primary processing functions are performed by hard-wired microprocessor boards. The software application programs are written by the user and are controlled by an Intel 8086 microprocessor central



OPTOMATION
VISION SYSTEM
DATA PROCESSING HARDWARE
FIGURE II

processing unit (CPU). The CPU controls the primary processing activities of the other microprocessors in the system.

Primary communications between the CPU and the microprocessor modules are transmitted on a system bus (Intel's Multibus). High speed data flow between the peripheral microprocessors is accomplished via custom high speed buses that take advantage of the special and highly structured nature of their communication tasks. Also attached to the system bus is a feature data base that operates with both the CPU application software functions and peripheral hardware.

Application programs are written in VPL (Vision Programming Language created by GE) which resembles BASIC programming language in logic, commands, and format. VPL is the user's interface with the vision system hardware.

The hardware modules that transform the sensory data to useful information are described below for GE's Optomation II vision system.

(1) Video Image Processing:

This module consists of three distinct units:

(a) Camera Image Interface (CII).

(b) Image Plane Memory (IPM).

(c) Image Control and Display (ICD).

Note that when vacuum tubes are used, the additional task of digitizing the analog signals from the sensor must be performed. The solid state CID camera used with the Optomation II generates a digital video signal with an eight bit resolution and a six bit dynamic range.

The camera interface unit receives the eight bit digitized video frame data and compares the value of each pixel with a user determined light intensity threshold value. If the incoming video signal is greater than or equal to the threshold value, a value of one is assigned to the pixel. Otherwise, a value of zero is assigned. Thresholding reduces the amount of data used to represent an image by a factor of eight and allows for faster processing in subsequent operations. This processed data is stored in the image plane memory.

The image plane memory's primary purpose is to provide a data buffer that allows incoming picture data to be processed and stored. Operations slower than the incoming data rates can finish processing their data at their own slower rates. The image plane memory consists of a four page 2x256x256 bit high speed dynamic RAM memory module. The first bit of the 256x256 array stores the thresholded bit value. The second stores a graphic overlay bit used to display window boundaries and other diagnostic overlay display features.

The image control and display module provides local control of picture taking timing and synchronization. It also controls data steering between:

a) the camera,

b) the camera image interface module, and

c) the image plane memory.

The data from the image plane memory is passed to the corner point encoder on a custom bus for analysis in the feature processing module.

(2) Feature Processing:

The tasks performed in the feature processing module are decomposed into five functions:

a) Corner Point Encoder (CPE).

b) FIFO data storage.

c) Feature Arithmetic Unit (FAU).

d) Feature Extractor Memory (FEM).

e) Feature Instruction Unit (FIU).

The corner point encoder has the task of recognizing corner points and boundary change directions (i.e., if the point passes from "0" to "1" or vice versa). It does this within a user determined window which limits the area which must be scanned for object features. Windowing and corner point encoding both reduce the amount of data used to represent the scene. This increases the processing speed of the ensuing modules. The corner point encoder begins its scan in the upper left corner of the window and proceeds across line by line down to the lower left corner of the window. Corners are identified by recognizing the bit patterns formed by a central pixel and its eight adjacent pixels. The coordinates of the corner points are stored in FIFO memory (first in- first out) until the corner points for the entire frame have been encoded. This data is sent to the feature arithmetic unit.

The feature arithmetic unit has several higher level functions to perform:

(a) Corner points are sorted and linked to form closed edge sets or items. If the item is smaller than a user determined minimum pixel area referred to as noise, it is discarded.

(b) Items are numbered sequentially as received from the FIFO module.

(c) The items are labeled as being either parts or holes in the parts. This is defined by the user who initially identifies parts as being white or black depending on the lighting and application.

(d) Standard geometric features such as the item's area and centroid are calculated.

(e) All item feature data is stored in the Feature Extractor Memory until the entire frame has been analyzed. This data will eventually be loaded into the feature data base.

The feature instruction unit controls the transfer of item data for each frame over the system bus to the feature data base. It is used again to handle VPL (Vision Programming Language) requests for secondary feature data calculations.

(4) Decision Processing:

Secondary processing commands are programmed by the user in VPL (Vision Programming Language). VPL allows the user to:

a) control the operation of the vision system's camera hardware (requesting pictures, setting threshold levels).

b) manipulate primary feature data to create higher level feature information.

c) analyze the higher level feature data by performing calculations and logic on it.

d) communicate results to other devices.

VPL camera control commands are translated in the CPU and sent via the system bus to the appropriate hardware controllers. VPL data analysis commands are sent to the feature instruction unit which controls the data transfer between the feature data base and the feature arithmetic unit. The feature instruction unit also sends the commands to the feature arithmetic unit for calculations. Results are sent back through the feature instruction unit to the feature data base.

B. CNC Controllers: The Mark Century 2000 CNC Controller

The General Electric Mark Century 2000 CNC Controller is a machine tool control system that replaces hard-wired machine controls with software. This creates a flexible control system that can be interfaced to virtually any machine tool. The 2000 controller's flexibility stems from the control software which is composed of modules that can be assembled by the user to suit the application required.

The controller has two types of software programs (see Figure III):

1) numerical control (NC) software,

2) machine control logic software. [13]

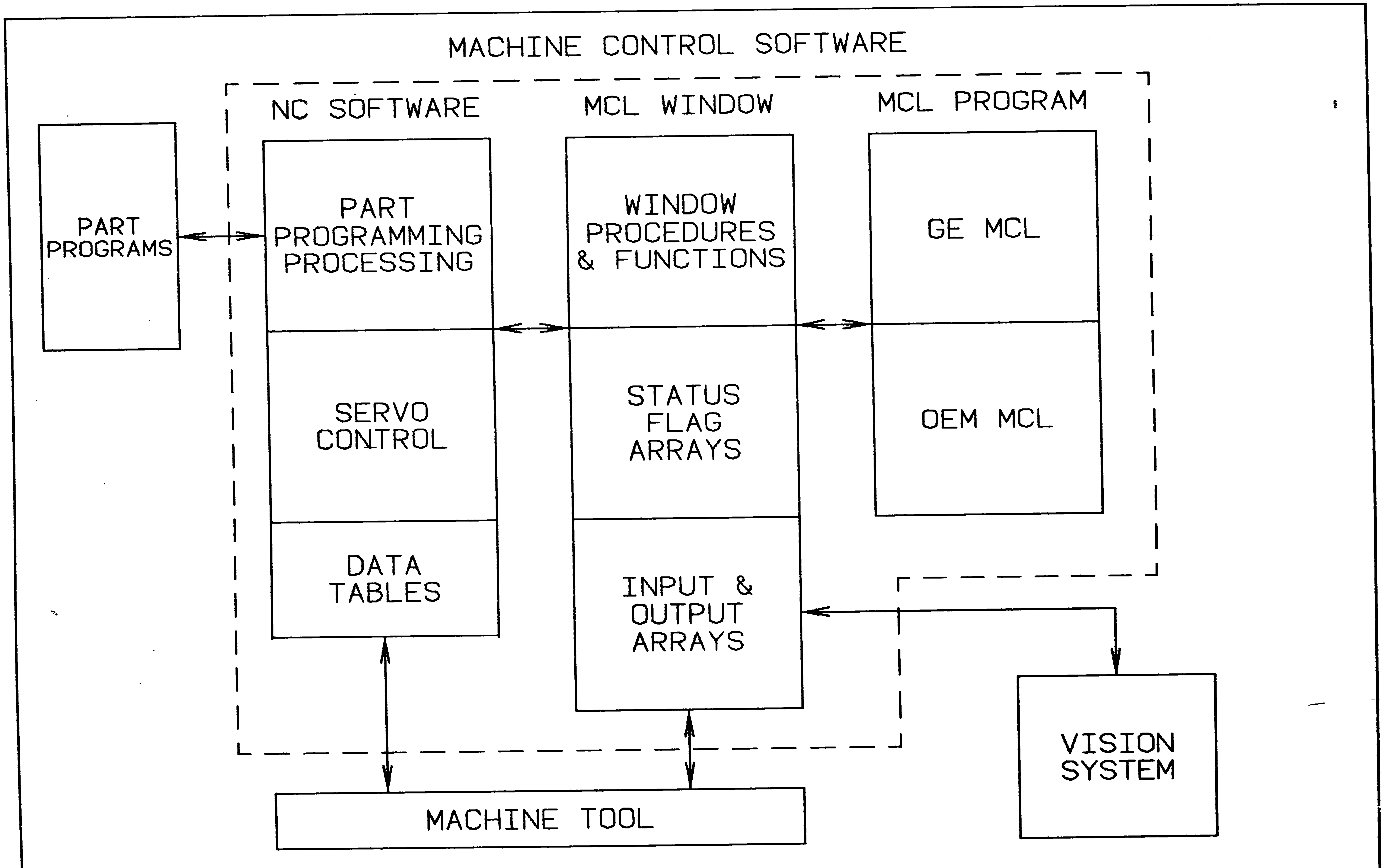
The numerical control software consists of programs which control:

1) the axial motions of the machine.

2) standard CRT displays.

3) part program processing and related data table modules.

The machine control logic (MCL) software controls the individual auxiliary functions of the machine. The MCL is



MC 2000 CNC MACHINE CONTROL OPERATIONAL OVERVIEW-FIGURE III split into subprograms that control the machine control station hardware of the controller (called GE MCL) and user created subprograms (called OEM MCL) that allow for the customization of the controller for user equipment and special needs. The special task of interfacing with the vision system from the part program was programmed in the OEM MCL.

The GE MCL controls tasks such as machine reference zero moves, manual table jogs, and other functions of the machine control station. The OEM MCL controls the operation of auxiliary machine equipment such as tool turrets, tool changers, gaging equipment, material handlers, and, in this case, a vision system. The OEM MCL is programmed by filling the framework of supplied packages with programming commands and variables written in a GE high level, structured language called PCL (Programmable Control Language). To create the entire OEM MCL package the user:

1. compiles all of the packages needed to operate the machine tool and peripheral devices from PCL to machine code.
2. links or combines all of the machine code files together into an MCL package.
3. locates or assigns all system variables in the file to RAM address locations.

After completing these steps, the MCL package performs with the MCL window and the NC software as shown in Figure III.

The MCL window provides a method of communication between the MCL and the NC software. The window contains three sections:

1. window procedures and functions including program related math functions, mode selection, etc.
2. status flag arrays which present the boolean condition of the operation of NC hardware, such as whether the "cycle start" has been activated.
3. input and output arrays which provide the status of hardware devices and present commands from the MCL to the hardware devices.

The following sequence of operations provides an example of how the controller functions described above were used to demonstrate closed loop machining using machine vision:

1. In the part program located in the NC software, a user designated code is encountered that changes a status flag in the MCL window.

2. A special package in the OEM MCL reads this change in status and runs its program which sends a signal to the vision system through the output array section of the MCL window.

3. The input from the vision system is received by the

OEM MCL program through the MCL window input array module and is processed.

4. The package in the MCL, through a special function in the MCL window, places the processed information in the data tables in the NC software section and changes the status flag in the status flag array section of the MCL window to indicate completion.

5. The part program uses the information in the data tables to process the part to complete the cycle.

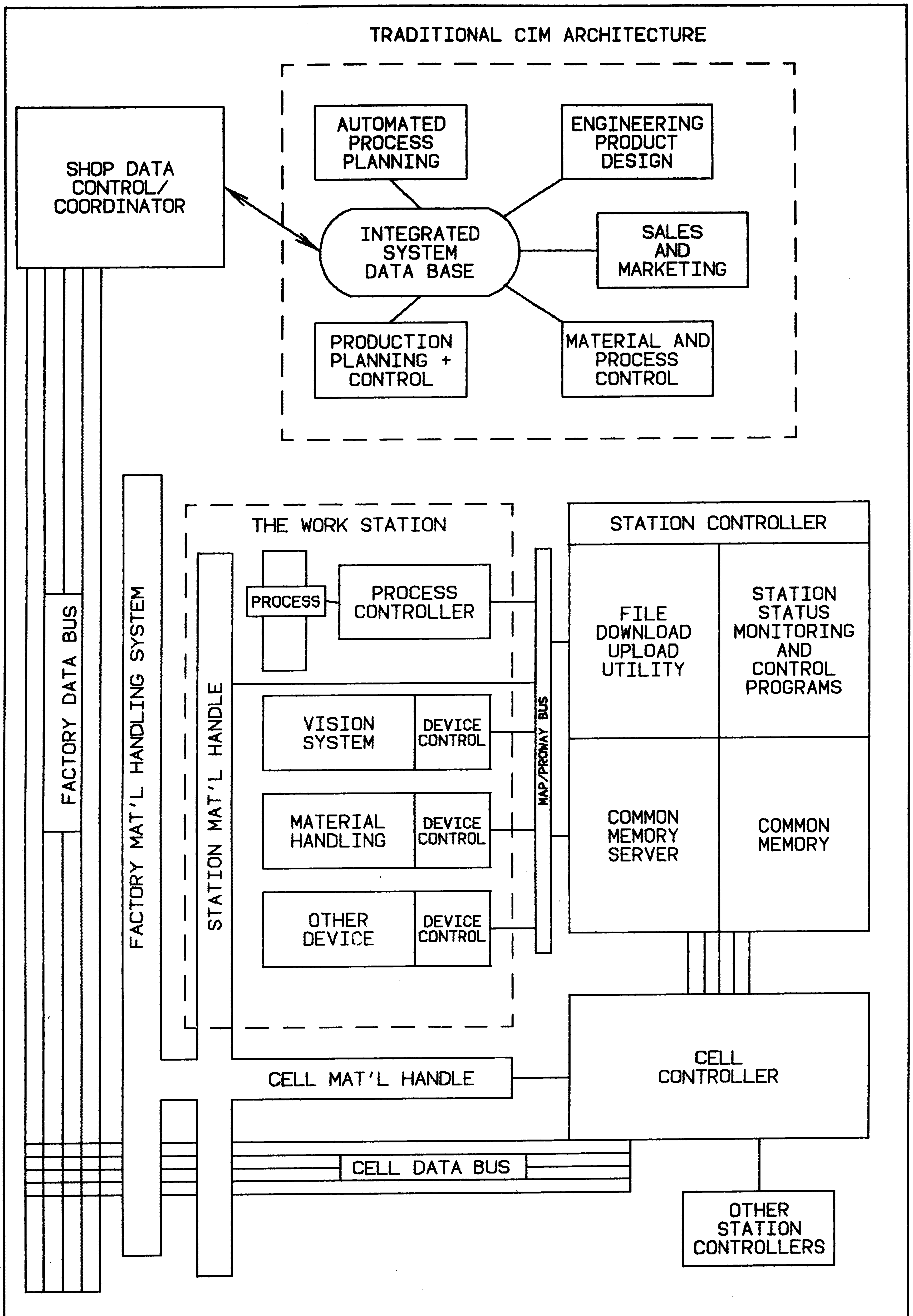
V. Ideal Design Specifications

A. Software System Specification.

In order to develop the ideal specification for the integration of a vision system with a CNC controller, a macroscopic view of the manufacturing system in which this system would operate is required. With a true CIM (Computer Integrated Manufacturing) system, all manufacturing computer information resources are available to be utilized. The ultimate goal of this system is to achieve a completely automated closed loop machining system from CAD (computer aided product design), through automated process planning, to the processing and inspection of acceptable final products.

An illustration of a version of a manufacturing system that could perform closed loop machining in a completely automated environment is shown in Figure IV. The data generation and storage base is centered around a typical CIM architecture similar to the CASA/SME (Computer and Automation Systems Association of the Society of Manufacturing Engineers) CIM architecture. [14] The need to use information from the common integrated data base is evident from the diagram and from the following description of the processing events.

After an order for a product has been received, the product design is created and/or retrieved from the



CIM SYSTEM DATA FLOW AND CONTROL DIAGRAM-FIGURE IV data base by the automated process planning module. In order to create inspection and processing programs automatically from the product design data base, the product design information residing in the data base will have to provide higher quality data than is currently available with IGES (Initial Graphics Interchange System), a standard format for storing CAD data. In the IGES format, tolerance data is listed in ASCII text notes of a dimension statement. This method does not provide a way of associating a tolerance band to a design feature. Current projects such as PDES (Product Data Exchange Specification) under way at NBS (National Bureau of Standards) are attempting to create a standard CAD data base format that will store a digital representation of the product design that completely defines a product for test and manufacturability.[14] In the future, a CAD created PDES file may be used to create the processing and inspection programs for the machine tool and, in this case, the vision system.

After production information has been coordinated with raw material procurement, the work can be scheduled. At this point, the job order enters a hierarchy of factory control analogous to NBS's automated factory control architecture. [15][16]

The highest level of the hierarchy in this application is the shop data control/coordinator. This control module coordinates the data flow to and from the machine cells and monitors the status of cell operations. It also coordinates the use and movement of resources, including the movement of material. Status information is continually fed back into the shop data control/coordinator from the cell controllers. This allows the master schedule module to update and rearrange work schedules according to the most current factory status information.

The cell controller receives the data files for the work stations within its cell. It coordinates the activities of the cell material handling devices with the station and factory material handling systems. The cell controller monitors the status of the cell devices and reports this information to the shop data control/coordinator. Information transfer occurs through a network on a system bus which operates using a standard network protocol such as MAP (Manufacturing Automation Protocol).

The work station controller performs more specific tasks such as processing, material handling, and CLM applications. The work station consists of processing modules, material handling modules, and sensor devices. Other devices not included in Figure IV, such as a cleaning station or a deburring station, would also be under the supervision of the station controller.

The work station controller communicates with its devices in two modes:

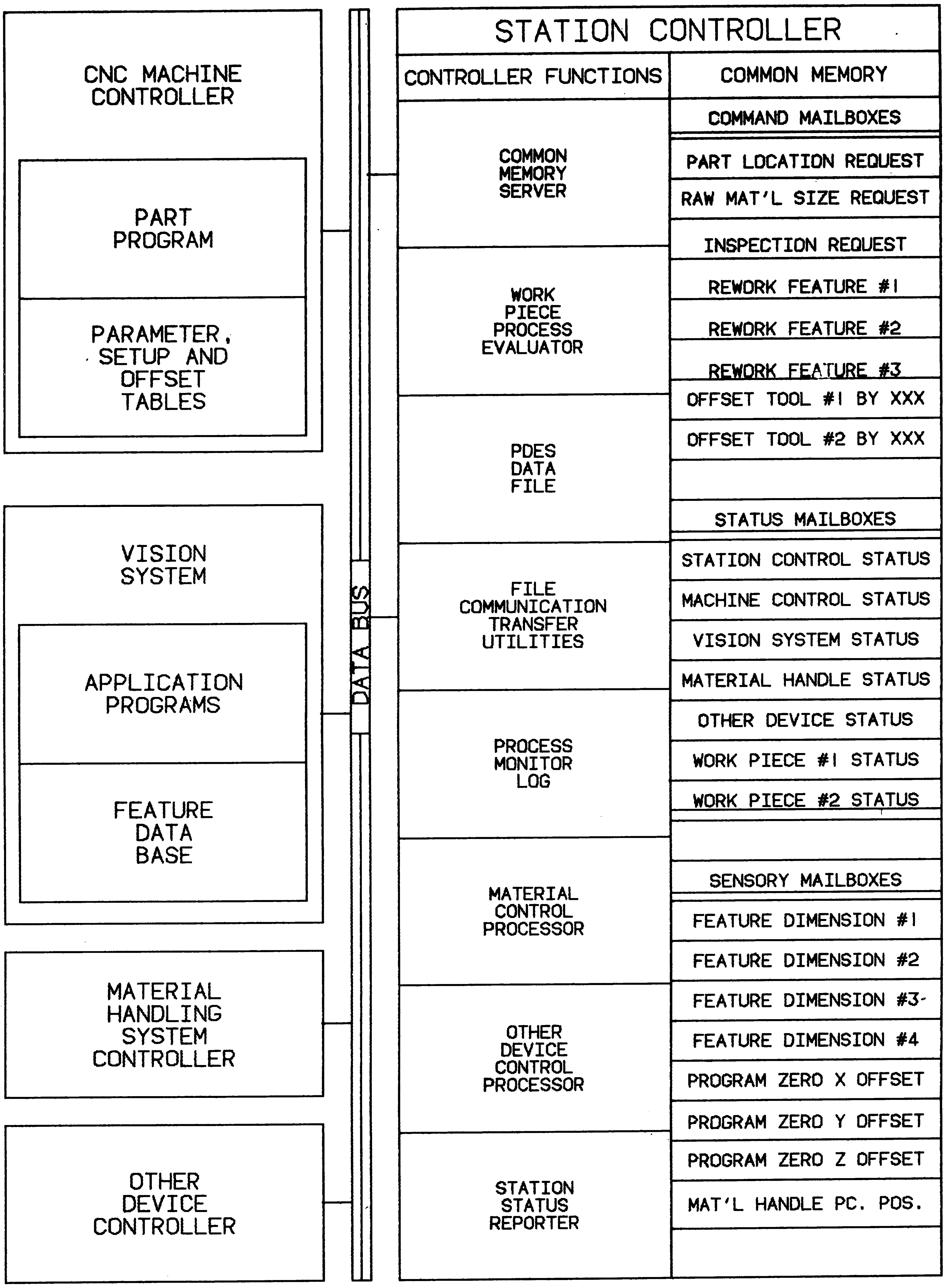
1) file downloading (uploading).

2) common memory communication.

The program files that run the work station devices are downloaded prior to the processing of each part. The CNC machines and other station devices should have this file loading capability.

Communication between station devices takes place via a common memory system similar to the NBS model.[16][17] All inter-device communication within the work station occurs through common memory which serves as a post office that contains mail boxes of work station information as pictured in Figure V. Using common memory for inter-device communication simplifies communication and protocol requirements because:

1. The only protocol required is the device to common memory protocol since the devices do not talk directly to each other.
2. Common memory is accessible from a variety of languages. Communication with common memory can be performed using data packets of ASCII code in a format prescribed by the common memory server.



SAMPLE STATION CONTROLLER COMMON MEMORY MAP-FIGURE V

An important characteristic of common memory is that devices can read mail from any mailbox but only write mail to their own mailboxes. These mailboxes are similar to user shared global variables in that any program can use the current values in the mailboxes or variables. They differ in that, with common memory, only one device can assign a value to the variables. [15][17]

There are three types of information in the mailboxes:

1. command mail.
2. status mail.
3. sensory mail. [15]

A map of the a common memory layout for a CLM application using a vision system and a CNC controller is presented in Figure V. Each component of the station writes status and operating data to the mailboxes that have been assigned to receive that data. The station controller uses the status information in common memory to coordinate the activities of all work station devices. The components also coordinate their own activities according to the information obtained from reading the other mailboxes. All programmable devices in the system must have the ability to communicate in real time (not just in a batch mode) in the common memory server's prescribed protocol.

The following sequence of events is an example of how common memory would be used to perform closed loop machining using machine vision:

1. To initiate a closed loop machining routine, a code in the process controller program is used to request information from the vision system by changing the command in the corresponding command mailbox.

2. The vision system then reads the command request mail and runs the appropriate application programs which were generated automatically from a PDES file. This file was also used to create the machine tool's part program. The dimension feature data extracted from the process scene are written to the corresponding sensory mailboxes, and the vision system completion status mailbox is changed to indicate task completion.

3. The station controller reads the change in status from the vision system and runs a "work piece evaluator" program. This program compares the dimensional feature data to the related feature tolerance band information in the PDES file. The station controller calculates and writes the appropriate changes to the machine tool's offset mailbox and indicates rework where required in the appropriate command mailbox. The status of the work piece is recorded, and the status of the station controller is changed to indicate that its tasks have been completed.

4. After reading the change of completion status in the station controller mailbox, the process controller updates

all tool offsets as programmed in the part program. If rework is indicated in the work piece status mailbox, those feature process blocks are rerun in the part program, and the inspection process is repeated. Otherwise, the material handling unit reads the work piece status mailbox and deposits the part either with the quality approved or varianced parts, according to the status read.

5. The process controller waits for the completion of the material handling process by monitoring the material handling device's status mail box before starting the process for the next piece. The station controller tracks the station's progress and reports efficiency and other statistics useful in job tracking, scheduling, and routing to the cell controller.

By using common memory, a more extensive array of various devices and sensors could easily be integrated into this system.

Both process controllers and station controllers should have bimodal network standard communication facilities. One mode should handle large data file communication using MAP-type protocols. The other should be used for communicating smaller amounts of data to and from common memory using a less demanding, faster protocol such as PROWAY.

B. Accuracy specifications.

Accuracy statements from manufacturers of vision system cameras present information regarding the precision or repeatable measurement within a certain tolerance assuming a unit of measure. Assuming no biases are introduced into the measuring system such as the lens distortions described in Section IV.A.3, the system is accurate to the stated resolution. [5]

Nominal camera resolution is essentially dependent on two factors:

1. the size of the field of view of the camera.
2. the size of the array of the photo sensitive grid used to capture the information.

$$\text{Nominal Resolution} = \frac{\text{length of a side of the field of view}}{\text{the number of pixels per side of the array}}$$

Sources vary as to the rule of thumb that should be applied to determine whether the camera resolution is high enough for the tolerance of the feature being measured. Two sources agree [10][18] that to obtain an accurate sample measurement, the feature being measured must cover at least four pixels. Another accuracy consideration is that binary systems may call a pixel close to the threshold black one time and white the next [19] as was observed in the Manufacturing Lab at Lehigh University. Therefore, a +/- 1 pixel accuracy per edge is realistic with a worst case error of two pixels.[10]

The overall rule of thumb of contact gaging is that the gage resolution be ten times the tolerance of the feature being measured. This rule of thumb has evolved from military contract specifications regarding the calibration of gage devices. Because there are less variations created by wear and positioning errors with non-contact gaging, the recommended camera resolution per tolerance is less stringent. Resolutions from three [18] to five [10] times the tolerance are recommended.

Using these resolution requirements, a chart was developed to illustrate the resolution and camera pixel array requirements necessary for various tolerance measuring requirements of industrial metrology. Table I shows:

1) the nominal resolution,
2) three times the nominal resolution, and
3) five times the nominal resolution

for varying pixel array sizes and field of view sizes. The data processing requirements in bits per frame and bits per second for different frame rates are listed at the bottom of the chart.

Table I: Vision System Camera Resolution Table

Field of View Size		Pixel Array Size		
		256x256	1024x1024	2048x2048
.5"x.5"	Nominal Res.	.0020"	.0005"	.0002"
	3 X's Resol.	.0060"	.0015"	.0006"
	5 X's Resol.	.0100"	.0025"	.0010"
1"x1"	Nominal Res.	.0039"	.0010"	.0005"
	3 X's Resol.	.0117"	.0030"	.0015"
	5 X's Resol.	.0195"	.0050"	.0025"
2"x2"	Nominal Res.	.0078"	.0020"	.0010"
	3 X's Resol.	.0234"	.0060"	.0030"
	5 X's Resol.	.0390"	.0100"	.0050"
4"x4"	Nominal Res.	.0156"	.0039"	.0020"
	3 X's Resol.	.0468"	.0117"	.0060"
	5 X's Resol.	.0780"	.0195"	.0100"
6"x6"	Nominal Res.	.0234"	.0059"	.0029"
	3 X's Resol.	.0702"	.0177"	.0087"
	5 X's Resol.	.0117"	.0295"	.0145"
12"x12"	Nominal Res.	.0469"	.0117"	.0059"
	3 X's Resol.	.1407"	.0351"	.0177"
	5 X's Resol.	.2345"	.0585"	.0295"
Frame Process Time	Bits per Frame	393,216	6,291,500	25,166,000
	30 fps*	11.8 mbps**	189 mbps	755 mbps
	60 fps	23.5 mbps	377 mbps	1.5 tbps***
	10 fps	3.9 mbps	63 mbps	252 mbps

* fps- frames per second
** mbps- millions of bits per second
*** tbps- trillions of bits per second

NOTE: 2048x2048 pixel array cameras are not commercially available at this time.

Some conclusions can be drawn from this chart:

1. Close tolerance metrology functions (ie. tolerances less than .005") using 256 by 256 pixel arrays are not possible within the given rules of thumb.

2. After the field of view exceeds one square inch, the corresponding resolutions are not adequate for close metrological work using commercially available cameras.

3. The extremely high volume of data processing for high pixel density cameras requires computer horsepower that does not appear to be cost effective based on the state of commercially available equipment.

To overcome these barriers of resolution and speed, the following system hardware designs would enable close tolerance metrology using vision:

1. Resolution.

a. Multiple cameras mounted adjacent to each other can increase the field of view while maintaining the same resolution. [Asea Robots]

b. By mounting a camera on a robot, the camera is able to measure large features while still maintaining the small field of view required for high resolution needed for close tolerance measurements. The special inspection robots used have extremely small position and repeatability errors (+/-

.0001") to reduce their effect on degrading resolution. The camera/robot combination would probably require a combination of sensors on the arm to allow for depth information processing along with the two-dimensional standard camera data. A cylindrical robot with four degrees of freedom and +/- .0001" stated accuracy has been employed using a CAD data base to drive the camera around part features.[3][20] Deviations from the CAD path were monitored by the vision system to provide the inspection information.

2. Processing Speed.

Recent advances in silicon hardware components have made possible more cost effective processing of large volumes of data in parallel and pipeline type modes.

a. SIMD (Simultaneous Instructions on Multiple Data streams) takes advantage of standard video image processing procedures that are performed on each frame of data. Because the same operation is performed on each pixel, hardware processors are dedicated to perform the task on individual pixels or sections of pixels processing the data in parallel as it passes through the system.[7] For example, thresholding processors could be set up to threshold an entire frame simultaneouly by passing each gray scale value for a pixel through its own processor that compares it to a threshold value. With one pass, the entire image is thresholded to a binary representation. Although SIMD increases processing speed significantly, the number of hardware processors required (per pixel, per function, etc.) is significantly large.

b. Pipeline hardware is structured so that image data passes through a "pipeline" or row of processors where each processor in the row performs the next logical processing step on the piece of data. The data from a vision system would consist of pixel data or more refined feature data. The normal processing scheme would perform an operation on each element of data of the frame before proceeding to the next operation.[7] The pipeline architecture performs a series of operations on each element of the frame of data in 1/Nth the normal processing time where N is the number operations in the processing scheme assuming the pipeline is fully loaded.

Currently available systems.

Vision systems commercially available have some of these capabilities:

-multitasking ie., running different application programs simultaneously.[21]

-enhanced parallel processing.[21]

-1024x1024x8 bit image processing capabilities at 7.5 frames per second.[Trapix 5500, Recognition Systems.]

-512x512 array cameras with maximum resolution of .2% of the field of view.[21]

-communication capabilities with MAP, Ethernet, DECnet and NET 488.[Trapix 5500]

-communication through shared memory on a 20 megabyte-per-second arbitrated bus.[S512, IRI Inc.]

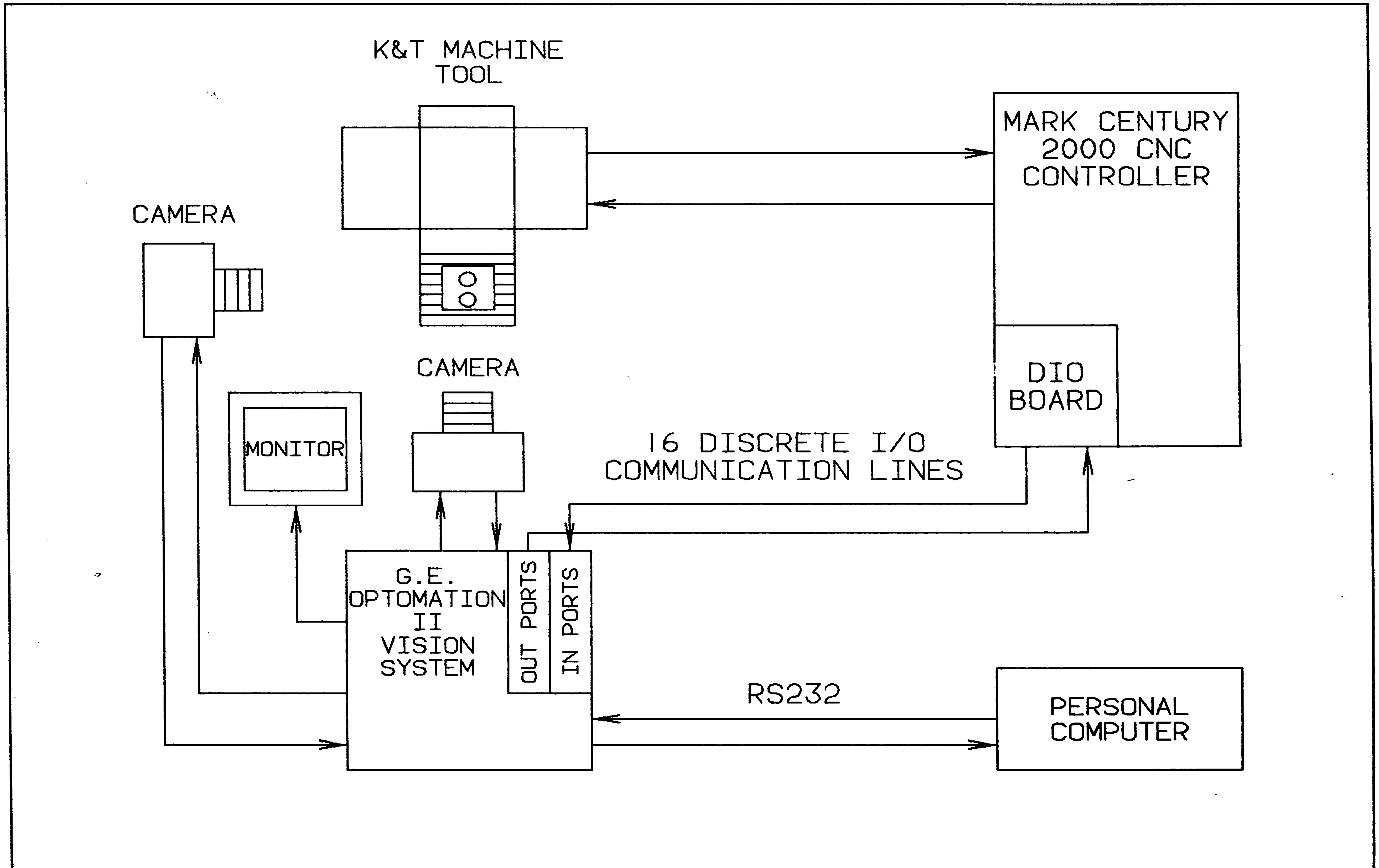
-capability of supporting up to 128 cameras with the same system controller.[S9][Trapix Ad]

VI. Demonstration of System Design

A. System Demonstration Design Constraints

The primary goal of the demonstration was to illustrate how a vision system could be integrated with a CNC controller to perform the closed loop machining functions described earlier. To achieve this, GE's (General Electric) Mark Century 2000 CNC Controller was connected to a GE Optomation II Vision System. The software to provide communication between systems and an illustration of the CLM applications were designed to encompass as many of the system design features presented in the specifications section as possible. Common memory was simulated with the use of a global variable data array in the vision system and with the use of a parameter table in the controller. As can be seen from the hardware configuration of the system shown in Figure VI, communication was performed using discrete input/output lines with a user developed protocol. MAP/PROWAY communication utilities or any non-batch mode data communication utilities were not available on either system.

The camera used was a solid state CID camera with a 244x248 pixel array.[11] The camera was placed over a CAD model of the table layout to accomplish the manual feature vision system teaching tasks. To increase the 16K memory



CLM HARDWARE DEMONSTRATION CONFIGURATION-FIGURE VI capacity for storing vision system programs, a personal computer (an IBM XT) was connected to enable the uploading and downloading of application programs from the personal computer's hard disc.

A major programming consideration in the design of system software of both systems was to keep the systems modular and flexible.

Modularity was achieved by keeping individual system functions as separate units or subroutines with generic inputs and outputs. By keeping the design modular, parts of the system or routines could be tested, debugged, and run independently of the rest of the program or other routines. For example, if a change in the user input to the setup of the inspection routine is required, only that routine is called and run as opposed to resetting the entire system. Also, programming changes made in one subroutine would not affect the operation of the other routines mainly because the information input to and output from all routines was retrieved from common memory, or in this case, a global memory array.

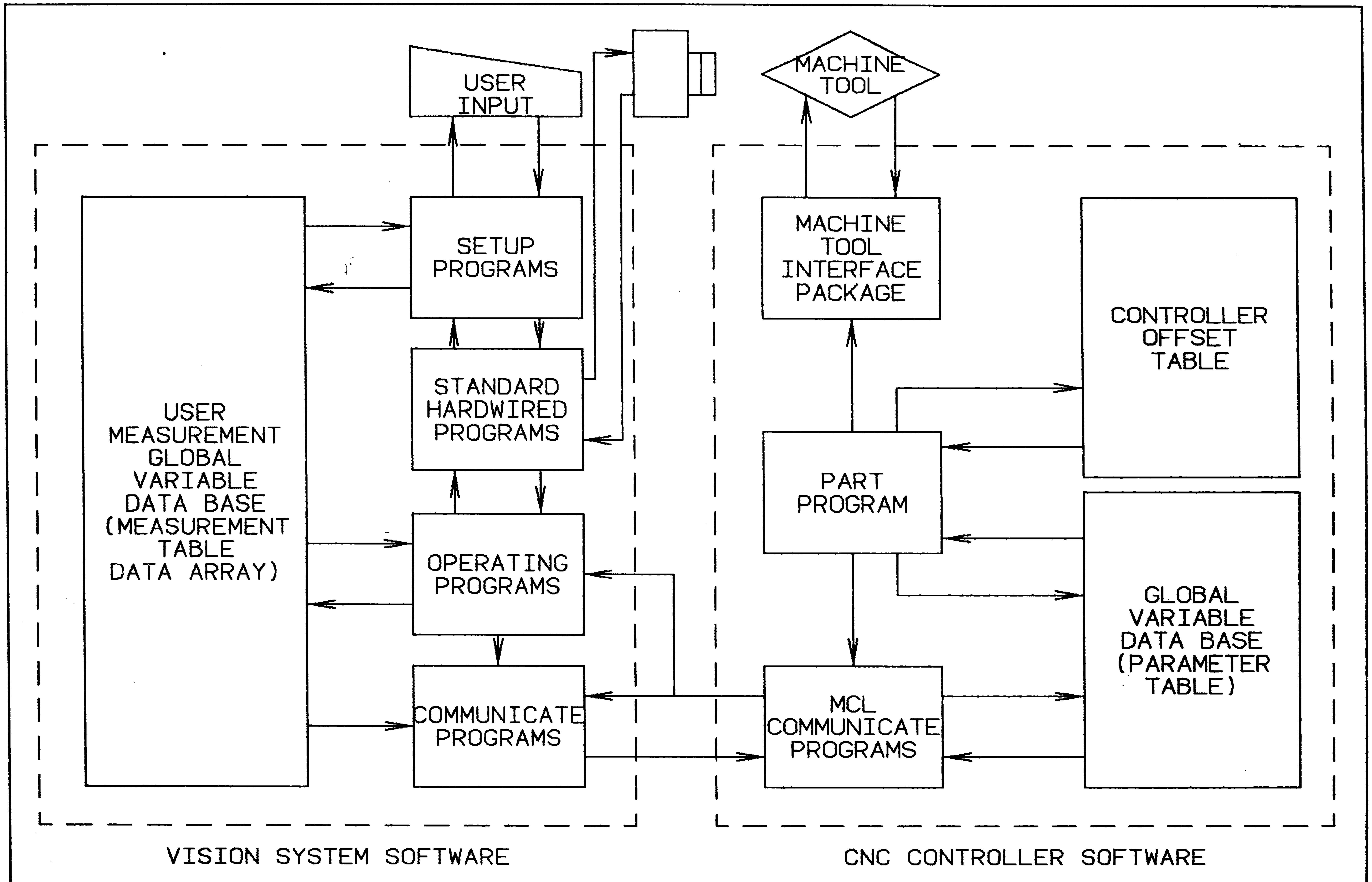
Because the systems were designed in modules, the user has the flexibility to put the modules together in many different ways suitable to the application. Also, the programs were designed to accommodate the expansion of the functions of the system, such as the addition of a second or third camera and other measurement analysis enhancements.

To maintain modularity across both systems and enable the integration of the two systems, common data bases and communication formats had to be standardized early so that both systems could be designed around the same information base.

B. Software System Overview

The goal of the integrated CNC-vision system is to automate the manual inspection, setup, and offsetting functions of a machinist so that a cell can operate in a completely automated mode. The software of the entire CLM system can be split into three categories (see Figure VII):

1. The process maintenance section is responsible for outputting quality product by monitoring and adjusting process variables such as tool wear and part positioning. These tasks are completed in the part program and data tables in the CNC controller.
2. The inspection section is primarily responsible for determining part location, inspecting the work piece against design, and determining the machine actions required resulting from the part inspection. These tasks are accomplished in the vision system.
3. The communication section transfers information



SYSTEM DEMONSTRATION SOFTWARE OVERVIEW-FIGURE VII requests and results between systems. These tasks are performed in specific communication routines in both the vision system and in the machine control logic (MCL) of the CNC controller. The inter-device communication tasks are always transparent to the user.

The vision system must be set up to operate in an automated mode. Setup entails:

1) teaching the system what and where the parts are in the scene.

2) calibrating the system.

3) teaching the system what dimensions should be inspected.

This is currently done with interactive routines that can be run independently from each other. Once the system setup has been completed, the system is placed in a fully automatic mode in which it scans all of its input lines waiting for an output signal from the controller.

After receiving an output signal, the vision system runs the corresponding routines which process the values requested by the controller. A communication routine then transmits the data from the global arrays in a predetermined protocol to the controller, and the vision system returns to reading the status of its input lines.

The programs in the MCL of the controller receive the data sent from the vision system and place it into its own variable data base for use in the part program. The variables are linked from the vision system to the controller via a pointer sent over in the vision system's communication packet. This pointer is actually the parameter number in the parameter table of the controller. The parameter table acts as the controller's variable data base for part programs.

In the part program, the software information requests and the logic required to perform the CLM task are programmed according to the user's needs for the particular application. The programming of CLM functions is flexible in that the part program structure for the application and logical use of the sensory data in the parameter table is left up to the programmer. An example program is provided in Table III to illustrate how the CLM data in the parameter table can be used. Standard subroutines can be written, as was done in the example program, to allow repetitive logical procedures to be performed with a subroutine call.

C. The Part Program and Controller Functions

The part program contains the instructions that enable the machine tool to process the work piece. To attain the goal of a completely automated machine cell, the machinist's tasks to be completed by the vision system and the controller must be incorporated into the part program with special programming commands. These special programming commands that perform the logical decisions of a machinist require the part programming language and machine controller to be able to:

1) make use of variables that can be assigned values from another source (i.e., another computer).

2) be able to perform logic in response to variable situations in the program (the logic consists primarily of conditional branching).

3) use tables supplied in the controller to collect, adjust tooling and other setup information in the normal operation of a part program.

Closed loop machining is made possible through the use of a parameter table, a group of tool offset tables, and special commands associated with a special communication package in the MCL.

1. The Parameter Table

The parameter table, shown in Table II, resides in the CNC controller and, as mentioned previously, allows for the assignment of variables within a part program.[1] Variables, in this case, are assigned from the vision system through a special interface package in the MCL. Parameters 10 through 99 (90 variables) are available for the user to assign values. In order to standardize the use of these variables so that the variables have the same meaning from program to program and machine to machine, the parameters were given permanent definitions as is shown in Table II. Appendix I contains the parameter assignment descriptions.

2. Controller Offset Tables

The CNC controller has a variety of data tables in addition to the parameter table previously described. The tool offset tables used in CLM are the setup offset tables, the tool length offset tables, and the tool radius offset tables. In addition to providing structured data tables to store various machining data, the controller provides special commands and status checks to aid in accomplishing CLM.[1]

1. Setup offset tables. The values in the offset tables for each of the three machine table axes establish a starting location for processing commands in the part program called the program zero location. The coordinates of the setup offset represent the distance from machine zero which represents a permanent position on the machine where all axes originate.

2. Tool offset tables. Each dimension has a finishing tool associated with it. Each dimension of a processed work piece is examined by the vision system. A

determination is made on whether to adjust the tool offsets

Table II: Controller Parameter Table

This table resides in the controller and allows for the assignment of values to variables within a part program. In this case, variables are assigned values from an outside source (a vision system) through an interface package in the MCL.

Controller Parameter Table Definitions

P10	SETUP OFFSET ADJUSTMENT FOR X AXIS
P11	SETUP OFFSET ADJUSTMENT FOR Y AXIS
P12	SETUP OFFSET ADJUSTMENT FOR Z AXIS
P13	MACHINE REFERENCE LOCATION FOR X AXIS
P14	MACHINE REFERENCE LOCATION FOR Y AXIS
P15	MACHINE REFERENCE LOCATION FOR Z AXIS
P17	GLOBAL SUBROUTINE PARAMETER
P18	1ST PASS STATUS/REWORK FLAG
P19	WORK PIECE VARIANCE FLAG
P20	INITIAL RAW MATERIAL SIZE MEASUREMENT (1)
P21	INITIAL RAW MATERIAL SIZE MEASUREMENT (2)
:	

TOOL DATA

P30	TOOL NUMBER OR TOOL POSITION
P31	TOOL OFFSET ADJUSTMENT
P32	MACHINING ACTION CODE
P33	TOOL NUMBER OR TOOL POSITION
P34	TOOL OFFSET ADJUSTMENT
P35	MACHINING ACTION CODE
:	

to compensate for tool wear.

Each tool contains an offset limit referred to as a tool offset tolerance. Adjustments to the tool can be made within the limits prescribed by the tolerance. This can be used to set a total tool wear limit range. Each time a tool offset is updated to compensate for wear, an offset table function checks that the total wear limit has not been exceeded for that tool. For example, if the offset tolerance for a tool is .007" wide and the cumulative tool offsets have exceeded .007" , a flag is set to alert the user that the wear limit has been exceeded and that the tool is worn. This tool status system variable is also checked in the part program. By using this feature, cumulative tool wear can be monitored.

3. Special Custom Programming Codes

In order to coordinate the operation of the vision system with processing in the part program, certain codes were created to pause in the execution of the part program and allow a communication program in the MCL to signal the vision system to run a particular routine. The communication program receives the requested information sent by the vision system and assigns this information to variables in the parameter table. Once communication has been completed, the part program resumes its execution and uses the new information in the parameter table.

The codes used are called m-codes or miscellaneous codes. They signal the vision system to send the controller information corresponding to three CLM tasks:

M21: Program Zero Location.

Prior to any processing of the work piece, the part programmer inserts this m-code (M21) into the part program and uses the information deposited by it in the parameter table to update the setup offset table.

M22: Initial Raw Material Size.

Before initiating roughing cuts of raw material, the part programmer uses this m-code to request the initial size dimensions. The dimensions to be measured are taught to the vision system during the vision system setup procedure. The part programmer uses this new data in the parameter table to start the roughing cut at the most efficient position.

M23: Part Inspection.

After finish cuts are processed, the part programmer can insert this code to inspect the work piece. The vision system runs through the inspection routine and returns to the parameter table the status of the work piece and the tool offset distance. The part programmer uses this information to:

1) update tool offsets.

2) check the status of the work piece and the tool.

3) determine corrective actions.

Note that certain m-codes are reserved in standard EIA coding formats. These m-codes were arbitrarily set for demonstration purposes.

4. The Part Program

The MC 2000 CNC controller uses an enhanced version of standard numerical control EIA code (RS358 ASCII character set) which expands the capabilities of the part programming code of the particular machine used.[2] Only the sections of the code relevant to the use of CLM functions will be presented. Specific cutting and machine tool commands have been omitted. Table III is a listing of sections of a program that illustrate the use of the CLM functions described in Section IV using the standard part programming language.

The part program is divided into logical sections, each of which is described in detail below:

Initialization.

In this section, program variables and format modes are identified and initialized along with other standard setup procedures. Machine specific setup procedures are not shown here.

Absolute positioning (G90) is specified as the

Table III: The Part Program

```
N001  ! Initialization.
N050  (ID,PROG,GENERIC,SAMPLE PROGRAM)
:
N010  G90
N020  P18=0
N030  P19=0
N040  (FMT1,SOV(X),SOV(Y),SOV(Z))
:
N099  ! Set Program Zero.
N100  M21
N110  (STO,54,FMT1,(P10+P13),(P11+P14),(P12+P15))
N120  G54
:
N199  ! Raw Material Size Adjustment.
N200  M22
N250  (IF P20<5.0 GOTO N290)
N260  (IF P20<5.1 GOTO N280)
N270  X___   Y5.1  Z___
N280  X___   Y5.0  Z___
N290  X___   Y4.9  Z___
:
N299  !Roughing Cuts.
:
:
N399  !Finish Cuts.
N400  !BLOCK 1
N410  (IF ((P32<>2) AND (P18<>0)) GOTO N420
:
N420  !Block 2
N430  (IF P18<>0 GOTO N440
:
N440  !Block 3
N450  (IF ((P38<>2) AND (P18<>0)) GOTO 499)
:
N499  !Offset Adjustment.
N500  M23
N510  P17=P32
N520  (GSUB,OFFSET)
N530  P17=P38
N540  (GSUB,OFFSET)
N550  (IF ((P18=0) OR (P19=1)) GOTO N650)
N560  (GOTO N400)
:
N650  M30
:
N690  (END,PROG)
N699  !Offset Subroutine.
N700  (ID,GSUB,OFFSET)
```

Sample Part Program (continued)

```
N710  (IF P17=0 GOTO N900)
N720  (UPD,((P17)-2),TRO,((P17)-1))
N730  (IF STATUS=0 GOTO N750)
N740  (MSG,CHECK FOR WORN TOOL)!flashing # in offset tbl.
N750  (IF P17<>2 THEN GOTO N780)
N760  P18=1
N770  (MSG,REWORK REQUIRED)
N780  (IF P17<>3 GOTO N900)
N790  P19=1
N800  (MSG,SCRAP WORK PIECE-SEE PARAMETER TABLE)
N900  (END,GSUB)
```

programming mode as opposed to incremental positioning. With absolute positioning, the movement instructions are always relative to the programmed zero location. This accommodates the reworking of sections of the program without having to run through the entire program. Incremental positioning would be difficult to use to automatically rework only parts of a program because the movement instructions begin from the last position of the tool. In this case, the incremental position would not be the correct tool position.

The CLM variables used in the part program are initialized. The first pass status is set equal to zero indicating that this is the first pass through the program. The scrap flag is also initialized (reset) for the new work piece.

A standard format for storing the setup offsets for the X, Y, and Z axes in the Setup Offset Tables is preset using the FMTn function (N040). This enables the storing of the new offsets to be achieved using only one line of code instead of three.

Set Program Zero.

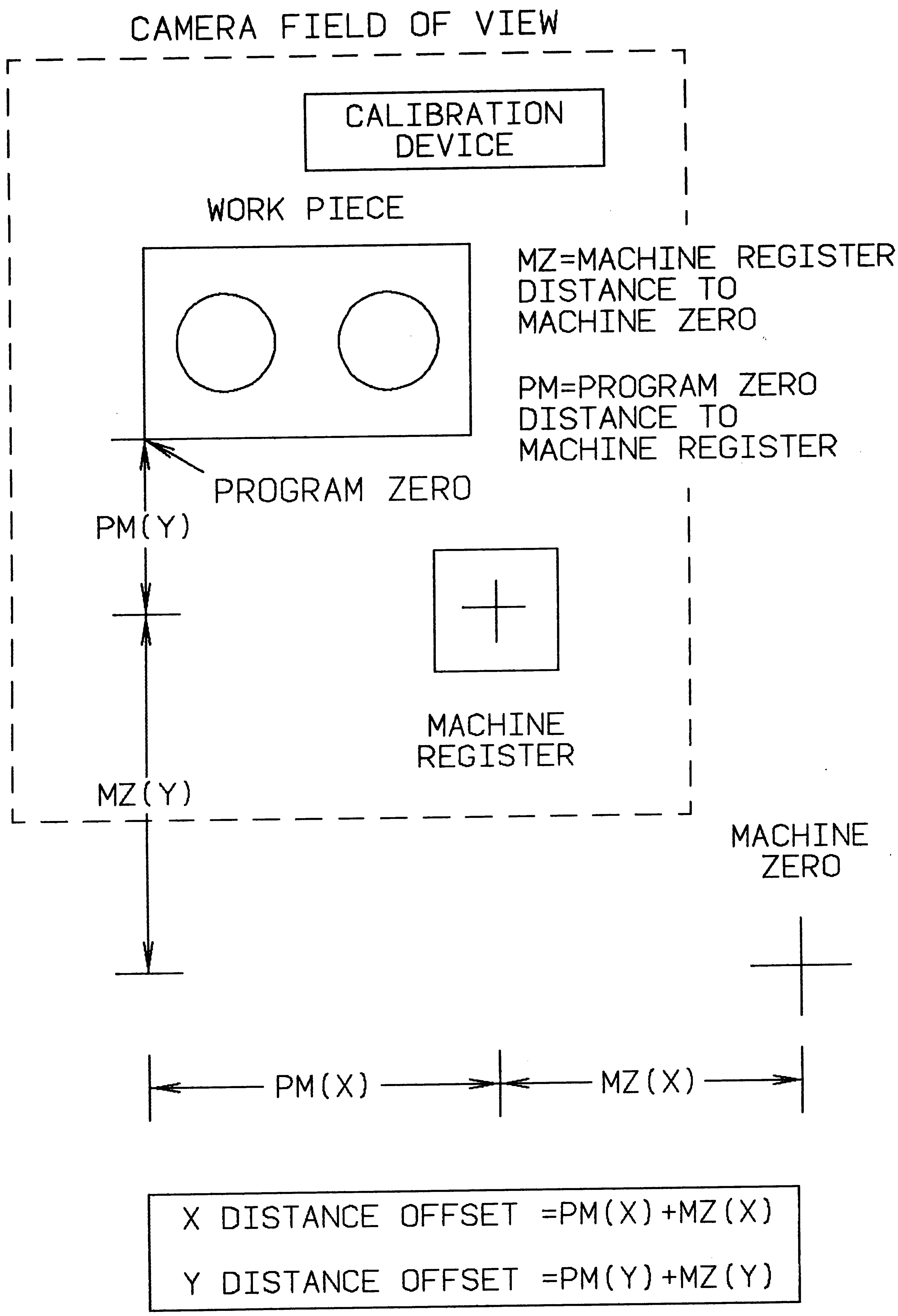
An m-code (M21) is called that signals the vision system to locate program zero and place the coordinates in the parameter table. The coordinate system of these values is with respect to the machine register and must be converted to the machine's coordinate system (see Figure VIII). This conversion is completed while storing the new setup offset values in the setup offset table (N110). The setup offsets are then activated (G54).

Raw Material Size Adjustments.

The part programmer uses this m-code (M22) when working with raw materials, such as castings, whose sizes vary enough to warrant special attention. The vision system is signaled to run the initial raw material size routine and returns the values of the initial size(s) to the parameter table. Based on the size of the raw material, the programmer can jump in the part program to a cutting location that is properly matched to the initial raw material size (N250-N290). This should result in a more efficient and safer roughing machining procedure.

Roughing Cuts.

The roughing cuts particular to the work piece being processed are performed in this section. Since this is a generic program concentrating only on required CLM code, this section is left blank.



MACHINE TABLE LAYOUT-FIGURE VIII

Finish Cuts.

Finish cuts are separated into blocks corresponding to the use of certain tools and cuts that can be rerun independently for rework. A logic statement that checks the rework status variables precedes each block in the finishing section. The rework status variable for the program (also called the first pass flag) is zero through the first pass of finish cuts, and therefore, all blocks are processed.

If rework is requested, the logic statement preceding that block will check to see that the rework status variable for the program (P18) is set equal to one and the machine action code for the tool associated with this block is set equal to two by the vision system to enable the reworking of that particular block. Otherwise, the part program will pass over that section.

If the programmer wants to ensure that a block is not reworked, the logic statement preceding that block would not check the machine action code and check that only the first pass status flag (P18) has been set equal to one (N430).

Offset Adjustment.

Use of the inspection m-code (M23) allows the user to inspect the work piece while it is still in a position to be reworked. Tool offsets can be adjusted for wear prior to rework or the processing of the next piece. The work piece also can be identified as having a variance.

When the M23 code is used:

1) all dimensions are measured.

2) the offsets of the tools are adjusted.

3) the status of the work piece is communicated back to the controller's parameter table from the vision system.

At this point the status of the machine action codes is checked for all measurements involved (see Table V for details on the possible machine action codes). An action code equal to zero indicates that the part was within tolerance on this dimension. If no adjustments are required for the tool, the next dimension action code can be examined (N710).

A code equal to one indicates that the part was within tolerance but requires an adjustment to the offset. The tool offset is updated using the value from the parameter table (N720). As a result of the update, the tool wear limit status is checked to see if the tool is worn and should be replaced (N730). If the status flag is set, a message is sent to the operator to replace the tool (N740).

A code equal to two means that the part was out of tolerance and requires rework. The rework flag is set equal to one (P18=1) and a message is sent to the controller screen indicating that rework is required.

An action code equal to three indicates that the dimension was out of tolerance and cannot be reworked on the machine with the current setup. The variance flag is set equal to one (P19=1) and a message is sent to the controller CRT indicating the part should be marked with a variance (N800). In a cell integrated with an automated material handling system, this part would be flagged to be removed from the normal work piece routing or floor holding space.

At the end of the part program, the program is rerouted to the beginning of the rework (finish cut) section only if the rework flag (P18) is set equal to one. If a part is marked with a variance or a tool needs to be changed, the program stops. If none of these cases apply, the program is started from the beginning after a new part is loaded.

D. Vision System Software

The user application programs for the vision system are written in a BASIC-like language. The system is structured by calling routines which perform standard tasks.[12] The system is built modularly by combining routines together in a pattern that suits the application.

In order for system variables to be passed from routine to routine, a global data array that contains all the information required by all the routines was created. This variable array must be declared as a global variable each time the system is powered up.

The programs written for the system fall into three categories (see Figure IX):

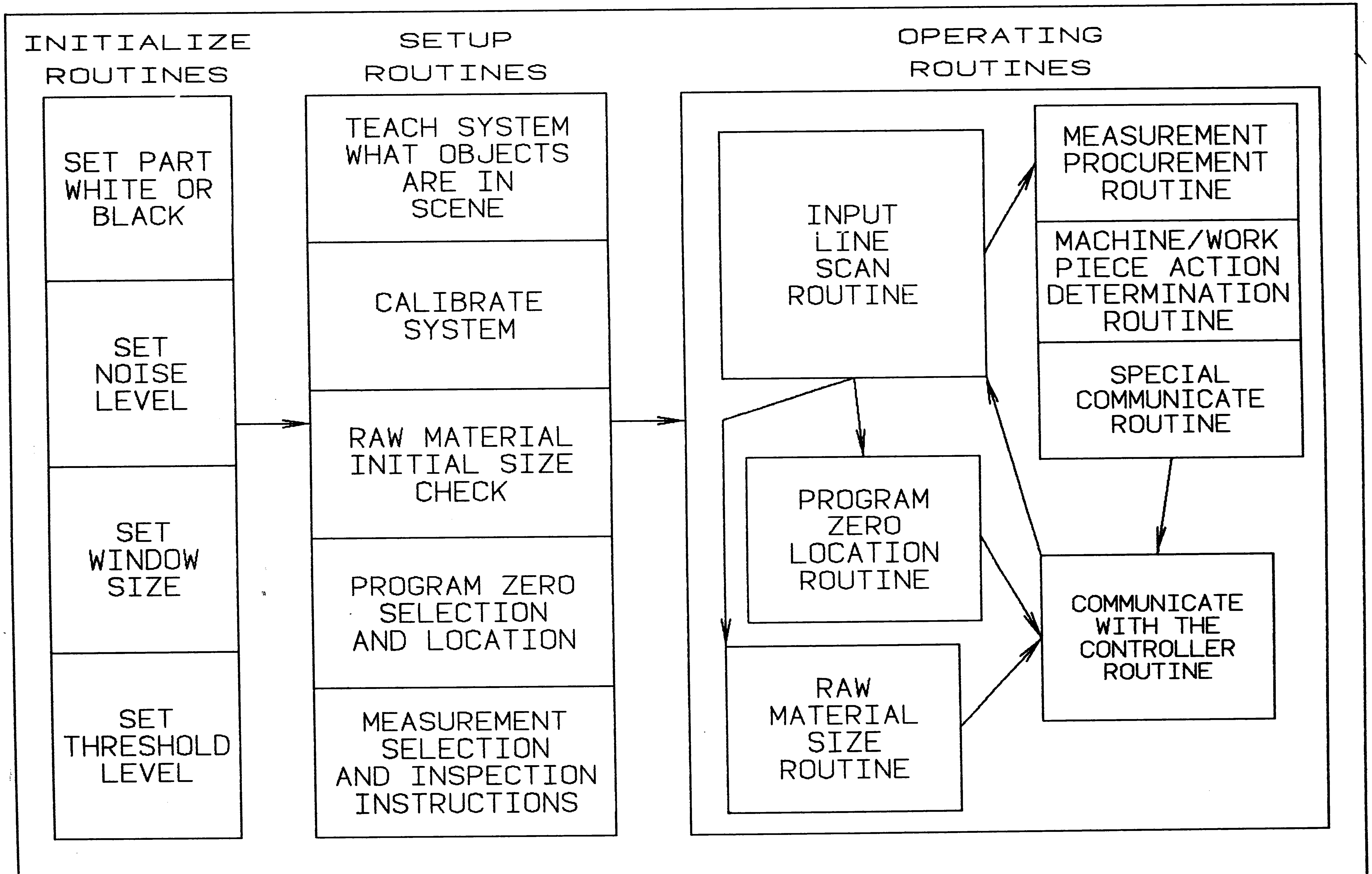
1. Hardware Initialization Procedures. These procedures are used when the system is powered up. They set operating hardware parameters such as threshold levels, noise levels, etc.

2. Software Inspection Setup Routines. The global measurement table data array is filled using these procedures. The system is taught what the items are in the scene and given measurement instructions to enable the completion of the measurements requested.

3. Inspection Operating Routines. This set of routines operates continuously scanning the input lines for requests from the controller. After servicing requests for program zero location, initial raw material size, or inspection and offset determination, the routine returns to scanning the input lines.

1. Hardware Initialization Procedures

Before processing visual data, the vision system



VISION SYSTEM APPLICATION SOFTWARE- FIGURE IX requires certain system parameters to be set. All of these initialization routines are in interactive programs.

Set Part Color. The color of the objects and the color of the background must be input which involves declaring the work piece in the scene to be white or black. This is needed to identify which feature items are objects and which are holes in the objects. Background areas are assigned negative values and object areas are positive areas.

Setting the light intensity threshold. Setting the correct light threshold intensity is one of the most critical and difficult initialization procedures since it determines the level at which pixels are declared as part of the background or part of the work piece to be measured. An incorrect threshold setting can severely affect the accuracy of the measurement results. Setting the correct threshold is hampered by the presence of glare, shadows, and non-uniform light dispersion across the scene. Threshold setting should be performed in an automatic routine (see Future Work) so that identical results from picture to picture can be obtained.

The current system depends on user judgment to determine the best threshold. The user inputs different threshold levels and examines the new picture on the video monitor to determine whether the threshold is at a stable, acceptable level based on a comparison between the true part representation and the binary picture.

Focusing. The camera is placed in a staging mode where a 64 gray level picture of the scene is displayed instead of the normal binary thresholded image. An item with contrasting lines, such as printed material, is placed in the field of view to set the focus, the aperture, and the field of view of the camera.

Setting the noise level. The user selects a noise level by choosing the minimum size area below any scene item size but large enough to eliminate scene noise. When a picture is processed by the vision system's hardware, items with areas smaller than the user determined noise level are ignored.

Setting the working window size. The vision system only processes information within the boundary of a user determined rectangular area in the monitor called a window. By creating a window around only relevant items in the scene, the time to process the scene data is reduced, and extraneous objects can be ignored.

2. Setup Routines

The setup routines are used primarily to fill the measurement table data array with the information that enables automatic picture taking and analysis. This table in the vision system is a three-dimensional variable data array (see Table IV). The first page of the array contains primary setup data identifying what the items are in the scene and which items are to be used for system calibration and program zero location. The second page contains the data required to complete the automated inspection and offset adjustments.

The setup routines consist of several different procedures described below:

The teach routine.

In order to teach the vision system what to expect in the scene and what measurements should be taken without the use of artificial intelligence, the actual work piece (raw material and finished product) should be used to teach the system. Future work with a CAD (Computer Aided Design) to vision system interface linked to an artificial intelligence system with sophisticated object recognition algorithms could enable the automation of the teaching process and other setup procedures (for more details, see Future Work).

The demonstration system uses a plot from a CAD system as a model of the work piece on the machine table to teach the system.

The teach routine identifies the items in the field of view of the camera (within the camera's window). This is

Table IV: The Measurement Data Table

The measurement table is a global variable array dimensioned in three dimensions: MT%(page #,a,b).

On page 1, the information pertains to primary setup data for part location, raw material size, and inspection routines.

Page 2 contains detailed instructions for inspection and offsetting routines.

page 1: Primary Setup Data

b=1: work piece item numbers
MT%(1,1-16,1): work piece feature item numbers extracted from the scene.

b=2: calibration data
MT%(1,1,2): Calibration item number.
MT%(1,2,2): Calibration units (calculated).

b=3: machine register data
MT%(1,1,3): Machine register item number.

b=4: program zero location data
MT%(1,1,4): Program zero location feature item #.
MT%(1,2,4): Measurement type (location of zero on item)
MT%(1,3,4): Distance to machine register (X axis)
MT%(1,4,4): Distance to machine register (Y axis)

b=5: Raw material size measurement data
MT%(1,1,5): Surface item number.
MT%(1,2,5): Measurement type.
MT%(1,3,5): Actual measurement.
MT%(1,4,5): Destination in parameter table of data.

page 2: Detailed Instructions for Inspection and Offsetting

The "b" array is used to identify the different inspection measurements, ie. each inspection has a different "b" value.

MT%(2, 1,b): Feature item number to be inspected.
MT%(2, 2,b): Feature item number of second item, if needed.
MT%(2, 3,b): Measurement type (height, width, etc.).
MT%(2, 4,b): Dimension of feature plus tolerance.
MT%(2, 5,b): Dimension of feature minus tolerance.
MT%(2, 6,b): Fixed offset zone band width.
MT%(2, 7,b): Outer dimension or inner dimension.
MT%(2, 8,b): Parameter number of final data destination.
MT%(2, 9,b): Actual measurement value.
MT%(2,10,b): Offset Adjustment.
MT%(2,11,b): Machine action code.

Optional for more than one camera:
MT%(2,12,b): Camera number.

Optional for views requiring detailed windowing:
MT%(2,13,b): Window coordinate (left x value)
MT%(2,14,b): Window coordinate (upper y value)
MT%(2,15,b): Window coordinate (right x value)
MT%(2,16,b): Window coordinate (lower y value)

accomplished through a user interactive query routine. The user is asked to identify what each item in the scene is according to the following choices:

1) the work piece.
2) the machine register.
3) the calibration device.
4) scene noise to be ignored.

The choice is entered after the vision system has drawn an overlay around the item on the system camera monitor.

The calibration routine.

The user selects the item to be used as the calibration device and inputs the device's dimensions. After taking a picture, the system calculates the number of system units equivalent to the user's units of measure. The calculated unit conversion value is stored in the measurement table data array (MT%(1,2,2)) and used every time a measurement is taken to convert vision system units to the units used on the machine controller.

The raw material initial size selection routine.

To enable the part programmer to adjust the path of roughing passes on the work piece according to the initial raw material size of the work piece, the variant surface is identified by the user. The user also selects the type of measurement required such as diameter, height, or width. These selections are recorded in the measurement table data array (MT%(1,1,5), MT%(1,2,5)).

The program zero location selection procedure.

To enable the vision system to communicate the program zero location of the work piece to the controller, the user identifies which feature on the work piece is to be used to locate program zero. The user is also asked which part of the identified feature is to be used such as the centroid. This data is stored in the measurement table data array (MT%(1,1,4), MT%(1,2,4)).

The measurement and inspection instruction routine.

This routine fills the measurement table data array with instructions and dimensional data that will allow the vision system to inspect and analyze the work piece.

The user selects the measurement type from a menu. The selections available are:

1) height.
2) width.
3) circular radius.
4) centroid to centroid distance.
5) feature boundary to item centroid distance.

The feature to be measured is selected and the dimensional requirements of the work piece for the feature are entered into the measurement table data array. This dimensional data is :

1) dimension size plus tolerance.

2) dimension size minus tolerance.

3) offset tolerance zone.

Items one and two create the dimensional tolerance band for the measurement of the feature. Item three, the offset tolerance zone, is used to determine when the dimension of the feature being examined is close enough to the edge of the tolerance band to warrant adjusting the tool's offset for wear. The measurement is also identified as being an outer dimension (ie. outer diameter) or inner dimension (ie. hole size). Offsets can then be applied in the correct direction.

The number of the tool offset corresponding to the tool associated with the processing of the measured feature is recorded in the measurement table data array. It is used as a pointer to the location in the controller parameter table for the information pertaining to that offset.

The remaining array spaces in the measurement table data array are for future enhancements of the measuring operation. If more than one camera is used to enable measurements in the X, Y, and Z planes or to increase the field of view of a single plane, the number of the camera used to obtain the particular measurement would be recorded so that the correct camera can be instructed to take the picture. This vision system has a four camera capacity but only one was available at the time of the development of this system.

Variable windowing would aid in the measurement of sections of complex and non-symmetrical objects. By varying the window size and location, specific sections of the object could be examined exclusively. An example of this would be the measurements of the diameter of a stepped cylinder on a shaft. Only the diameter of the shaft step to be measured would appear in the window at one time. More memory than was available was required to store and process the necessary information. Therefore, implementation of these features is left for future work.

At the completion of the setup routines, the vision system would be ready for automatic, integrated operation with the CNC controller.

3. Operating Routines

Each task routine (program zero location, initial size, inspection and offsetting) follows the same basic format. The pictures are taken and the measurements are extracted from the feature data base in RAM, reorganized, and analyzed. The type of measurements requested are listed in the measurement table data array. An average is taken of all measurements to smooth any measurement variations created by the inaccuracies involved with pixel thresholding. The measurements are then calibrated into the units used for machine control. After further analysis, the data is ready to be transmitted to the controller.

The data is prepared for communication with the controller in a special routine that separates a number into individual digits and places them in a special communication array. The communication routine transmits the prepared information in a predetermined protocol to the controller (see section on System Communication). After all communication has been completed, program control is returned to the line scan routine, and the vision system checks the lines for information requests from the controller.

Line scan routine.

This routine is a simple loop that scans all of the input lines connected to the controller. When the controller requests information by turning one of these lines high, the routine identifies and calls the corresponding task routine. When the called routine completes its tasks, control returns to the line scan routine . This routine allows the vision system to service the controller upon request.

Program zero and raw material initial size routine.

Both of these routines have the same logic format. After initializing routine variables, the routines:

1. take a series of pictures.
2. extract the measurements according to the instructions in the measurement table data array.
3. average and calibrate the results.
4. write the results to the measurement table data array.
5. prepare the data for communication using the communication preparation routines.
6. communicate the values to the waiting controller routine.
7. return program control to the scan line routine.

The program zero location routine always sends two pieces of data to the controller: x-axis and y-axis distance from the machine register. The raw material size routine is currently setup to send only one raw material size but could easily be reprogrammed to send more than one size.

Inspection and offset adjustment routines.

To keep the inspection and offsetting routines manageable in size and function, the tasks were divided logically into three routines to perform measurement procurement, machine/work piece action determination, and special communication preparation.

Measurement procurement routine.

After a picture is taken, the measurement is made according to the instructions in the measurement table data array. The measurements are averaged, calibrated, and stored in the measurement table data array.

Machine/work piece action determination routine.

Under ideal conditions, this routine, which determines the necessary adjustments to the machine tool offset and the status of the work piece, would be run in a cell controller or in the machine controller itself (see Figures IV and V). For the purposes of demonstration and logistical simplicity, the logic is performed in the vision system in this routine.

The Measurement State Table (see Table V) illustrates the possible actions resulting from various measurement results. The measurement state is determined by where the actual measurement lies in relation to the tolerance band and the offset tolerance zone within the tolerance band associated with that feature (see diagram below the table in Table V).

If the actual measurement is outside of the tolerance band, the part is undersized or oversized. Depending on whether the part is an outer dimension (O.D.) or inner dimension (I.D.), the part can be either reworked (oversized O.D. or undersized I.D.) after adjusting the

Table V: Measurement State Table

The measurement state table represents the possible scenarios resulting from inspection measurements. The inspection results as states and the resulting actions depend on the type of dimension being measured. The action codes are recorded in the controller parameter table as machine action variables and used in the part program to change offset values, rework cuts, or send a message to variance a work piece.

Measurement States		Machine/Work Piece Action Status Codes	
Within tolerance	O.D.*	0	No action.
	I.D.	0	No action.
Within tolerance, low end of tolerance band.	O.D.	0	No action, wear adjust.
	I.D.	1	Increase offset.
Within tolerance, high end of tolerance band.	O.D.	1	Increase offset.
	I.D.	0	No action, wear adjust.
Out of tolerance, undersized.	O.D.	3	Decrease offset, scrap
	I.D.	2	Increase offset, rework.
Out of tolerance, oversized.	O.D.	2	Increase offset, rework.
	I.D.	3	Decrease offset,scrap.

*O.D.=outer dimension; I.D.=inner dimension.

Dimension Diagram

```
--------------------------   Dimension plus tolerance
Offset Tolerance Zone
--------------------------

--------------------------   Designed dimension

--------------------------
Offset Tolerance Zone
--------------------------   Dimension minus tolerance
```

When the work piece measures in the offset tolerance zone, the offset should be changed to compensate for wear and avoid reworking parts. The change should equal the distance from the actual measure to the center of the tolerance band.

offset or indicated as a variance (undersized O.D. or oversized I.D.) and set aside from further processing until examined manually.

If the actual size is inside the tolerance band and not in the offset tolerance zone, no action is required and the part passes WIP inspection. If the part size falls within the offset tolerance zone, the part passes inspection but tool offset adjustment is required.

The tool offset adjustment, according to current NC machining practices, is to move the offset of the tool so that it would cut the work piece in the center of the tolerance band. Therefore, the adjustment would equal the distance of the difference between the current measured actual cutting path and the midway point of the tolerance band.

The offset adjustment (if required) and the appropriate machine/work piece action status code are recorded in the measurement table data array.

Inspection Communication routine.

This inspection communication routine differs from the communication sections of the other task routines in that a variable number of measurements need to be transmitted to the controller depending on the the number of measurements requested by the user. To accommodate this feature, the inspection communication routine sends the number of measurements it needs to transmit to the controller (see System Communication Section for details of this operation). This routine then prepares the data for communication and calls the digit conversion routine and communication routine to send the data to the controller.

Digit conversion routine and communication routine.

To facilitate communication with the controller in a standard module or routine, a global variable array was used to create a data packet to send to the controller. The first two positions of the array contain the controller pointer information regarding the destination of the data in the controller. These two positions are filled prior to calling the digit conversion routine.

The fourth through tenth array of the variable contains a six digit number and a sign bit (for details of the array content, see Section on Communication Data Format). The digit conversion routine converts the whole number to be communicated into six digits and places them into the communication array with the least significant digit in the four array position. The sign is bit placed in the tenth position of the array. The communication array is full at this point and ready to be transmitted to the controller.

The communication routine transmits the contents of the communication array in a data packet according to a predetermined protocol. The protocol used in this routine is explained in detail in the Systems Communication Section.

E. System Communications

1. Design Constraints

Although both the vision system and the CNC controller had standard communication features to facilitate intercomputer communication, the function of these facilities was primarily for file uploading and downloading in batch mode. This did not satisfy the need for real time communication generated with standard system software in the part program using standard communication protocols. The only real time communication available to the user that was common to both systems was through discrete input/output lines. This presented a significant limiting system design constraint in determining the type of data that would be transmitted between systems. It also created the need for a unique communication protocol scheme that would enable the synchronization of data transmission over discrete communication lines for two systems that operate at different processing speeds.

2. Physical Connections

Communication is performed on 16 discrete input/output lines- eight for the vision system to controller connection and eight for the reverse connection as shown in Table VI.

The vision system has eight discrete input and eight discrete output ports that are connected to the outside world through optically isolated relays. These relays transform the incoming signal to the correct system voltage (5 v.DC) and transform the outgoing signals from the vision system voltage to the 24 v.DC level of the receiving CNC controller. The highest numbered line in each set of eight ports represents the least significant bit of parallel transmission. These lines from the vision system are connected directly to pins on the digital input/output board (DIO board) of the controller. The controller pin assignments to the vision system ports are shown in Table VI.

3. Communication Data Format

The vision system to controller data communication format involves the parallel transmission of BCD (binary coded decimal) digits in packets of ten digits. Each digit in the data packet is an integer from zero to nine and has

Table VI: Communication Discrete Line Layout

Vision System Input Ports		Controller Output Pins (DIO board #1)	
0	<-------------	2	M21- request for part location
1	<-------------	3	M22- request for raw mat. size
2	<-------------	4	M23- request for inspection
3	<-------------	5	(null)
4	<-------------	6	
5	<-------------	7	Echo of
6	<-------------	13	Sequence
7	<-------------	14	Number

IDC24 Opto isolators used.

Output Ports		Input Pins	
8	Digit ------->	1	Seq_4
9	Sequence ---->	2	Seq_3
10	Number ------>	3	Seq_2
11	(1-10) ------>	4	Seq_1
12	-------------->	5	Dig_4
13	Integer ----->	6	Dig_3
14	(0-9) ------->	7	Dig_2
13	-------------->	8	Dig_1

ODC5 Opto isolators used.

a preassigned meaning:

Digit #	Description
1	The ten's digit of the parameter # of the final destination in the controller of the data.
2	The one's digit of the parameter number.
3	The machine/work piece action status code for the work piece (only used after the inspection routine).
4	The least significant digit of the 6 digit number.
5-9	Digits of increasing significance of the 6 digit number.
10	Sign bit: zero if the number in 4-9 is positive; one if the number is negative.

An example of a typical data transmission is shown below:

data packet sequence: 1234567890
data transmitted : 1003345001 means parameter #10,
action code 0, number -5.433.

The digit information is transmitted on the four least significant data lines and the data packet sequence number is sent on the four most significant lines.

Communication from the controller to the vision system is completed using two formats. The format of the first signal type is performed by producing a high signal on one of three lines connected to the vision system. Each line corresponds to a routine in the vision system and is controlled by the use of m-codes in the part program. The vision system scans those lines to determine which routine to run.

The second type of information format involves writing the digit sequence number in parallel on its output lines. This is used as an acknowledge signal by the controller to the vision system confirming the receipt and process completion of the latest data transmission as described in the protocol section.

4. Communication Protocol

This protocol is provided for the synchronization of data transfer between the controller and the vision system during parallel BCD data packet transmission.

Initialization. While the vision system is awaiting a service request signal from the controller, it outputs all high signals on its output port lines. After receiving and processing the controller information request, the vision system is ready for data transmission and clears its output lines by outputting low signals. It then waits for the controller to clear all of its output lines before beginning transmission.

The controller waits for all of its input lines from the vision system to be cleared before clearing its output lines. Once the controller clears all of its output lines and all data lines are clear of signals, the vision system is ready to start transmitting the data packet to the controller.

In the case of the inspection request, the number of data packets to be transmitted depends on the number of inspections requested. Therefore, an additional initialization step is required. The number of packets to be sent is transmitted to the controller over its 4 most significant output lines and is sent back by the controller as acknowledgment of receipt and process completion. The initialization procedure of line clearing is repeated and the system is ready for the normal data packet transmission.

Communication Maintenance. The vision system sends the digit sequence number on the four most significant output lines and an integer value on its four least significant output lines. It awaits the echo of the sequence number on its input lines before sending the next digit in the sequence.

The controller waits for a change in the sequence number sent from the vision system. When the sequence number has changed, it reads and processes the integer value received on the four least significant input lines. After processing, the controller then outputs the sequence number to the vision system and again, waits for the sequence number to change.

After the entire packet has been received the information is further processed and stored in the parameter table at the location determined by the parameter number sent in the first two digits of the data packet. If another data packet is expected, the controller initializes itself and receives the next packet. Otherwise, program control is returned to the part program.

VII. Summary

This thesis has explored the feasibility of using machine vision to provide a CNC controller with closed loop machining capabilities. The current manual methods for compensating for tool wear, variable raw material size, and variable work piece location were described to highlight the tasks that require automation.

A review of the functions of the components of vision systems and of the software structure of the CNC controller provided technical background information for subsequent explanations of system operations. Different illumination techniques, accuracy errors induced by lens systems, and the operation of vacuum tube and solid state cameras highlighted some of the technology available and problem areas of using machine vision in a machine work station. The operating systems of the vision system and the controller were also reviewed to illustrate how they would be interfaced with the application programs developed in later sections.

The design specifications of a machine vision system operating in a computer integrated manufacturing architecture were created to illustrate the future developments required to enable complete automation of the vision system closed loop machining functions. Using the data resources of the CIM data base and a hierarchical control scheme, the vision system was designed to operate in a work station using common memory to communicate between work station devices. The accuracy specifications were also analyzed to determine the required camera resolutions for various tolerance measurements.

The software used to integrate the data provided by the vision system with the logic of the part program was presented. By using special user written routines in the machine logic of the controller, the part program was able to request specific data from the vision system. The vision system results were transmitted to the controller's parameter table where they could be used in the logic of the part program. The logic that performs closed loop machining in the part program was shown in a sample generic program. The vision system hardware initialization routines, scene and measurement instruction routines, and actual measurement and offset calculation routines were described to illustrate how the visual sensory data is extracted and processed. A review of the communication protocols and data format was presented to illustrate how communication between systems was achieved.

VIII. Conclusions

It can be concluded that machine vision can be used to provide a CNC controller with automated compensation for tool wear, variable raw material size, and variable part location. The software required to perform these tasks, called closed loop machining functions, was demonstrated on a Mark Century 2000 CNC controller connected to an Optomation II vision system in the Manufacturing Lab at Lehigh University. It can also be concluded that the following developments would increase the usefulness and potential of vision systems to be used for close tolerance metrology in machining operations given the current state of vision systems:

A. The development of larger pixel array cameras would provide resolutions more suitable for close tolerance metrology, as was shown in Table I.

B. The use of gray scale vision processing for metrology applications would reduce the binary thresholding dependency of near uniform illumination of the object scene. This requirement is difficult to obtain in a manufacturing setting.

C. It can be speculated that more powerful vision sensor array processors will be needed to handle the increase of data to be processed with larger camera pixel arrays and gray scale processing vision systems.

Gray scale processing vision systems require six to eight times more processing than binary thresholding systems.

D. Based on the proven applications of robot based vision systems, further development of these systems would reduce the pixel array size requirements for close tolerance metrology by adjusting a camera's field of view to satisfy the resolution requirements of the tolerances being measured. Robot based vision systems also have the potential to increase the flexibility of the measurement capabilities of the camera by enabling the camera to make measurements in different planes.

IX. Future Work

A. Current Configuration and Equipment. After additional memory and cameras are connected to the vision system, the following areas of future work can be pursued:

1. Measurement enhancement. Using a variable window setting and window coordinate storing routine, the ability to view and measure only parts of the scene would increase the measurement capabilities of the vision system. Also, the addition of another camera would add a third dimension to the system's measurement capability. This task would involve coordinating the common points of view in the scene to establish a common base for the cameras. The calibration of a multiple camera setup also presents challenges.

2. Automatic threshold routine. To provide stability when using binary thresholding, a routine should be developed that automatically calibrates the threshold of the system based on a known object area in the scene or some other method. This routine should provide consistent measurements with varying system light intensities.

B. Vision Systems in Conjunction with Other Systems.

1. CAD based teach routines. By using CAD data to create the inspection routines, the manual task of teaching the vision system what and where the measurements are in the scene can be eliminated. This would require the

linkage of CAD dimensional and tolerance information with object recognition and inspection routines in the vision system.

2. Robot based vision systems. The connection of a camera to the end effector of a robot arm to perform work piece inspection presents numerous different research opportunities in robot control, inspection methodology, and device integration.

X. References

1. "Part Programming Manual 2000MC CNC." The General Electric Company-Industrial Controls Division, Charlottesville, Virginia, GEK-25384E, May 1984.

2. Groover, Mikell. Automation, Production Systems, and Computer Aided Manufacturing. Englewood Cliffs: Prentice-Hall, Inc., 1980, pp. 240-247.

3. Pastorius, Walt. "Gaging and Inspection in the Industrial Environment Through the Use of Machine Vision." Machine Vision Association of the Society of Manufacturing Engineers, Vision '85 Conference Proceedings. Detroit, MI: pp. 7-57 to 7-63.

4. Whitney, Donald. "Optical Gaging- Crankshaft." Machine Vision Association of the Society of Manufacturing Engineers, Vision '85 Conference Proceedings. Detroit, MI: pp. 7-1 to 7-12.

5. Stafford, Richard. "Induced Metrology Distortions Using Machine Vision Systems." Machine Vision Association of the Society of Manufacturing Engineers, Vision '85 Conference Proceedings. Detroit, MI: pp. 3-38 to 3-48.

6. Groover, Mikell; Nagel, Roger; Odrey, Nicholas; Weiss, Mitchell. Industrial Robotics: Technology, Programming, and Applications. McGraw-Hill, Inc., 1986.

7. Rutledge, Gary. "An Introduction to Gray Scale Machine Vision." Machine Vision Association of the Society of Manufacturing Engineers, Vision '85 Conference Proceedings. Detroit, MI: pp. 3-1 to 3-10.

8. Trepagnier, Pierre. "Recent Developments in Visual Gaging." Machine Vision Association of the Society of Manufacturing Engineers, Vision '85 Conference Proceedings. Detroit, MI: pp. 3-12 to 3-20.

9. Mersch, Steven. "Polarized Lighting for Machine Vision Applications." The Third Annual Machine Vision Conference Proceedings. Detroit, MI: pp. 4-40 to 4-54, 1984.

10. Englander, Arnold, and Lapidus, Stanley. "Understanding How Images are Digitized." The Third Annual Machine Vision Conference Proceedings. Detroit, MI: pp. 4-22 to 4-33, 1984.

11. "TN2500 Solid State Video/ Digital Camera Operating Manual." The General Electric Company- Intelligent Vision Systems Operations, Syracuse, New York, April 1980.

12. "PN2305 Optomation II VPL Language Reference Manual." The General Electric Company- Intelligent Vision Systems Operations, Syracuse, New York, February 1984.

13. "Introduction to PCL/MCL- A Self Teaching Manual." The General Electric Company- Industrial Controls Division, Charlottesville, Virginia, NEC-1214, November 1983.

14. Stauffer, Robert. "IGES and Beyond." *CIM Technology*, Summer 1985, pp. 20-21.

15. Libes, Don. "User-Level Shared Variables (in Hierarchical Control." *USENIS Summer Conference Proceedings*, 1985.

16. Albus, J.S.; Barbera, A.J.; and Nagel, R.N. "Theory and Practice of Hierarchical Control." *Proceedings of the 23rd IEEE Computer Society International Conference*, Washington, D.C., 1981.

17. McClean, C.R.; Bloom, H.M.; and Hopp, T.H. "The Virtual Manufacturing Cell." *Proceedings of the 4th IFAC/IFIP Symposium on Information Containing Problems in Manufacturing Technology*." Gaithersburg, MD, October 1982.

18. McCracken, William. "Integrating a Machine Vision System into the Manufacturing Process." *Machine Vision Association of the Society of Manufacturing Engineers, Vision '85 Conference Proceedings*. Detroit, MI: pp. 1-36 to 1-46.

19. Denker, Stephen. "Choosing a Vision System that Improves Testing Effectiveness." *Test and Measurement World*, October 1984, pp. 192-200.

20. Bahn, Michael and Harned, John. "Flexible Dimensional Gaging System." *The Third Annual Machine Vision Conference Proceedings*. Detroit, MI: pp. 3-23 to 3-35, 1984.

21. __________. "Machine Vision Reduces Costs and Improves Quality." *Sensor Review*, July 1985, pp. 158-160.

XI. Appendix I- Controller Parameter Table Definitions

Table II provides a listing of these parameters.

P10, P11, P12: Program Zero distance from the machine register for the X, Y, and Z directions, respectively.

The value in this parameter represents the distance between program zero and the machine register. The machine register is a known distance from machine zero. The value is input by the vision system.

P13, P14, P15: Machine register distance from machine zero in the X, Y, and Z directions, respectively.

When the machine zero location is not in the field of view of the camera, a machine register is placed in the field of field of view. Its position relative to machine zero is known so that it can be used as an intermediate location for setting program zero. This value is input by the user and should be permanent for the setup. By adding the distance from machine zero to the register location to the value of the distance of the register location to the program zero location, the machine setup offset for program zero can be determined for each setup. (See Figure VIII for an illustration of this.)

P18: First pass status/rework flag.

This parameter is set equal to zero during the first pass through the program and set equal to one if, after part inspection, any rework is required in any block of the part program. At the end of the program, if the flag has not been set, the program ends. If the flag has been set, the program reruns those sections flagged for rework. The use of this parameter as a flag allows the program to facilitate rework using logical branching.

P19: Work piece scrap flag.

This flag is set if the machine action code returned from the vision inspection indicates the part should be scrapped because a dimension was out of tolerance and was not reworkable in its present condition on the machine. If this flag is set, a message is sent to the CRT, and the operator is told to look at the parameter table to see which dimension is out of tolerance.

P20...P23: Raw material size measurements.

The vision system places the size of the raw material dimension requested into these parameters. The part program then logically determines where to begin its roughing cuts based on this information.

P30, P33,...,P96: Tool # or tool position #.

These are permanent tool assignments for a particular machine or group of machines. For instance, the 2" end mill tool position # will always reside in parameter 30, the center drill tool position # in parameter 36, etc.

P31, P34,...,P97: Tool Offset Adjustment.

This value is sent from the vision system after the inspection data for the dimension associated with the tool has been processed through the measurement state table residing in the vision system. The decision processing to offset the tool, the amount and the direction of the offset is performed in the vision system and explained in Table V. The value of this parameter is input into the tool offset table in the controller.

P32, P35,...,P98: Machine Action Code.

This parameter stores the instructions for the part program resulting from the inspection of the work piece. It has four possible values as illustrated in Table V:

0: Work piece is within tolerance, no offset adjustment required.

1: Work piece is within tolerance, offset is required because dimension is close enough to the edge of the tolerance band.

2: Work piece is out of tolerance and can be reworked after the tool is offset.

3: Work piece is out of tolerance and can not be reworked. Offset adjustment required.

The part program checks the status of these parameters in order to determine:

1) which tools require offsetting.
2) if any tools need to be changed.
3) which program blocks require rerunning.

4) if the work piece is to be varianced.

XII. Appendix II- Vita

David Samuel Hanan was born in Portchester, New York to parents Barbara and Solomon Hanan on May 12, 1960. After graduating from White Plains High School in 1978, he attended Lehigh University in Bethlehem, Pennsylvania where in 1982, he obtained a Bachelor of Science Degree in Industrial Engineering with a minor in Economics. He worked in industry for The New York Hospital as a Project Management Engineer and as a Manufacturing Engineer for Air Products and Chemicals before pursuing a Master of Science Degree in Manufacturing Systems Engineering at Lehigh University which he completed in 1986.