Design Concept and Prototype Development of a Flexible Integrated Vision System

Kok-Meng Lee*
Robert Blenis
The George W. Woodruff School of Mechanical Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0405

Received August 13, 1993; revised July 21, 1993
Accepted September 20, 1993

Many industrial tasks require sophisticated vision interpretation, yet demand low cost, high speed, accuracy, and flexibility. To be fully effective, future machine vision systems must be able to handle complex industrial parts. This includes verifying or recognizing incoming parts, and determining the location and orientation of the part within a short cycle time. It is generally required that the cycle time required by the part-feeding system be less than that of the subsequent processes where parts are fed. To achieve this goal, a Flexible Integrated Vision System (FIVS) has been developed. FIVS offers performance and cost advantages by integrating the imaging sensor, control, illumination, direct digitization, computation, and data communication in a single unit. *© 1994 John Wiley & Sons, Inc.*

多くの工業用のタスクでは、低コスト、高速高精度、柔軟性をもつ洗練された視覚変換が要求される。この要求の全てが満たされれば、将来の機械視覚システムは、複雑な工業用の部品を扱うことができる。これには、短いサイクル・タイムでの、対象部品の確認または認識、そして、部品の位置と座標の決定が含まれる。一般に、部品供給システムに要求されるサイクル・タイムは、その後にある部品が供給されるシステムのサイクル・タイムよりも短くなければならない。これを実現するために、Flexible Integrated Vision System (FIVS) を開発した。FIVS では、イメージ・センサ、コントロール、照明、直接デジタル化、演算、データ通信を1つのユニットにまとめることで、性能とコストを改善している。

*To whom all correspondence should be addressed.

CCC 0741-2223/94/050387-012

1. INTRODUCTION

Although it has been well recognized in the past decade that vision can add considerably to flexible part feeding by simplifying grippers, component feeders, and location tooling, and can reduce the engineering time required to implement it, the capabilities of commercial vision systems for use in part verification, kitting, and presentation for robotic assembly are still very limited. Many vision systems today normally employ a camera that outputs a video signal limited by the traditional TV standard and an object-dependent structured illumination. For use in a robot vision system, the video camera must be accompanied by a frame grabber board and a high-performance host computer. The conventional vision approach generally discards color information and requires a substantial amount of memory, data Communication time, and sophisticated vision interpretation.

Conventional vision systems conform to the RS-170 standard[1] established in the 1950s, which defines the composite video and synchronizing signal that the television industry uses. It specifies a standard frame rate for visual interpretation. The RS-170 standard limits the read out of a complete frame at a rate of 30 frames per second (fps). Several problems associated with these conventional systems are as follows: (1) These vision systems require a host computer and a frame-grabber to digitize and process the acquired images. The frame-grabber converts the RS-170 video signal into a series of n-bit brightness values (gray levels) by using a flash analog-to-digital (A/D) converter. The digitized image is stored in the frame-grabber memory buffer. An image of m rows by n columns has $m \times n$ pixels and so requires a substantial amount of memory and loading time. Often only a few of these $m \times n$ pixels carry the information on which a machine vision system will base a decision, generally making "frame grabbing" inherently wasteful.[2,3,4] (2) The timing of the RS-170 standard video signal and the frame grabber must be synchronized. The sync pulse that signals the start of the video frame may be supplied by the camera or by the frame grabber board. In either case, any timing errors cause mismatches in the synchronization and horizontal scaling. These errors, which may be insignificant for visual interpretation, would cause errors in the geometric measurement or coordinate computation of the parts. (3) After the image has been digitized, it must be transferred from the frame grabber over the data bus of the host computer. The data bus of micro-computers has often restricted the through-put of the image processing.

Digital Signal Processors (DSPs) are increasingly being used as add-on boards to the host computer. DSP's permit high-powered, mathematically-intensive algorithms to manipulate and enhance images before the location and orientation of the part are computed. Despite the fact that many frame-grabbers and DSPs operate at a rate typically faster than that of the host bus, the frame grabber and DSP must continually wait on data from the host computer. This wait represents crucial overhead in a real-time environment. Although external connections to allow direct transfer between the frame-grabber and add-on DSP board have recently been developed, these add-on DSP boards are generally designed to perform pre-processing after the frame grabber has captured an entire image. As a result, they do not completely eliminate the need to transfer pixel data over the host-bus for further analysis beyond pre-processing.

An early attempt at solving some problems associated with conventional vision systems was the development of the Landmark Tracking System (LTS), which was designed for tracking well-defined retro-reflective landmarks in navigation of an automated guided vehicle (AGV).[5] Unlike the landmark tracking problem where the targets are two-dimensional and well-defined, industrial parts are often three-dimensional and have a wide spectrum of shapes, sizes, and surface reflectances. Presenting these industrial parts to robots often requires sophisticated vision algorithms that may utilize color and/or stereo information and the hardware to support it. To meet some of these requirements, several modifications were made to the LTS electronics for generic part-presentation, resulting in the Integrated Vision System (IVS).[6] A comparison between an early version of the integrated vision system and the off-the-shelf RS170-based video system has been conducted by Lee[7] on part presentation.

Dickerson and Lee[8] presented a particular design of an integrated vision system to eliminate an intermediate step converting the charge-coupled device (CCD) output to a TV video signal (such as RS170). Their design has rather simple microprocessor hardware to generate CCD control signals, the video memory addresses, and the control signals for the pixel data storage. The use of special hardware allows a relatively slow processor for data acquisition. As do most of the image processing systems, their design requires storage of the digitized image in a video buffer before performing any image processing. Target applications for the original IVS are those requiring simple image processing, where color and stereo information is not needed, and the cycle time is less than 15 fps. Although the original IVS is somewhat limited, it has been found very successful in low-cost applications incorporating retroreflective materials as a background or as engineered landmarks.

Many industrial tasks require sophisticated vision interpretation, yet demand low cost, high speed, accuracy, and flexibility.[9] To be fully effective, future machine vision systems must be able to verify complex industrial parts and determine their location and orientation within a short cycle time generally less than that of the subsequent processes where parts are fed. Thus, the part presentation must be computationally efficient. Requirements of machine vision systems for practical part-presentation and vision-based motion control applications in terms of flexibility, reliability, maintainability, fast execution time, real-time communication, and cost-effectiveness are demanding a compact system consisting of a few elements where the CPU, peripheral circuitry, and sensor are integrated.

This article presents the design concept and prototype development of a flexible integrated vision system. The contributions of this study are briefly summarized as follows: (1) The system offers performance and cost advantages by integrating the imaging sensor, control, illumination, direct digitization, computation, and data communication in a single unit. As a result, it allows relatively simple camera/illumination hardware without the limitations of a TV video standard, efficient computation, and low packaging cost. (2) The design concept offers a means to directly process pixel data computationally without having to store the image in a video buffer prior to processing. Thus, a significant fraction of image acquisition and processing time can be saved, which is often essential in real-time object tracking. (3) The design concept has significant potential in real-time motion-control system applications in which a vision system is used as an on-line sensor. By providing direct computation of controlled variables, the need to transfer pixel data through the host bus can be eliminated. Thus, the design concept offers a potentially useful solution to the problem of excessively long sampling time delay due to the vision feedback, which has often plagued the real-time vision-based motion-control applications.

The remainder of this article is organized as follows: Section 2 presents the design concept and issues associated with the implementation of FIVS for robot vision applications. Section 3 illustrates the basic operation of the FIVS with two examples, along with some experimental results. Section 4 suggests some applications of FIVS. The conclusions are summarized in Section 5.

2. DESIGN CONCEPT

To overcome the limitations associated with conventional RS-170 based vision systems and the LTS/IVS systems, a FIVS design as shown in Figure 1 was conceptualized.

The objectives of the FIVS design concept are three-fold: The first objective is to maintain low cost by integrating the processor and imaging sensor, thereby eliminating the need for the host computer, frame grabber, and dedicated computation hardware. By eliminating the host computer and frame grabber, the camera is no longer restricted by the RS-170 standard and thus frame rates higher than 30 fps can be achieved. The second objective is to offer the flexibility to process and/or store the digitized pixel data without having to store the image in a video buffer, thereby minimizing the image acquisition and processing time. The third objective is to maintain flexibility. Because a wide variety of manufacturing processes, product designs, and material handling systems are commonly found in a flexible manufacturing system, it is desirable to have a unique vision system that can be applied flexibly to any system configuration with little or no modification. Also, the system must be flexible enough to allow the user to reprogram the system freely and quickly, implementing a variety of image processing algorithms. From these objectives, the design concept of FIVS was developed.

2.1. FIVS Hardware

As shown in Figure 2, the central control unit of the flexible integrated vision system is a microprocessor-based control board. The design goal is to have all of the real-time processing performed by using the microprocessor control board without relying on any other system or computer. Thus, it is desired that the microprocessor have the following features: (1) It should have an on-chip program memory and independent on-chip data memories. These memories must be externally expandable and accessible with zero wait states. (2) It should have independent execution units that are connected by independent buses to the on-chip memory blocks. This feature is essential to provide the parallelism needed for high-performance digital signal processing and high-powered computation of mathematically intensive algo-

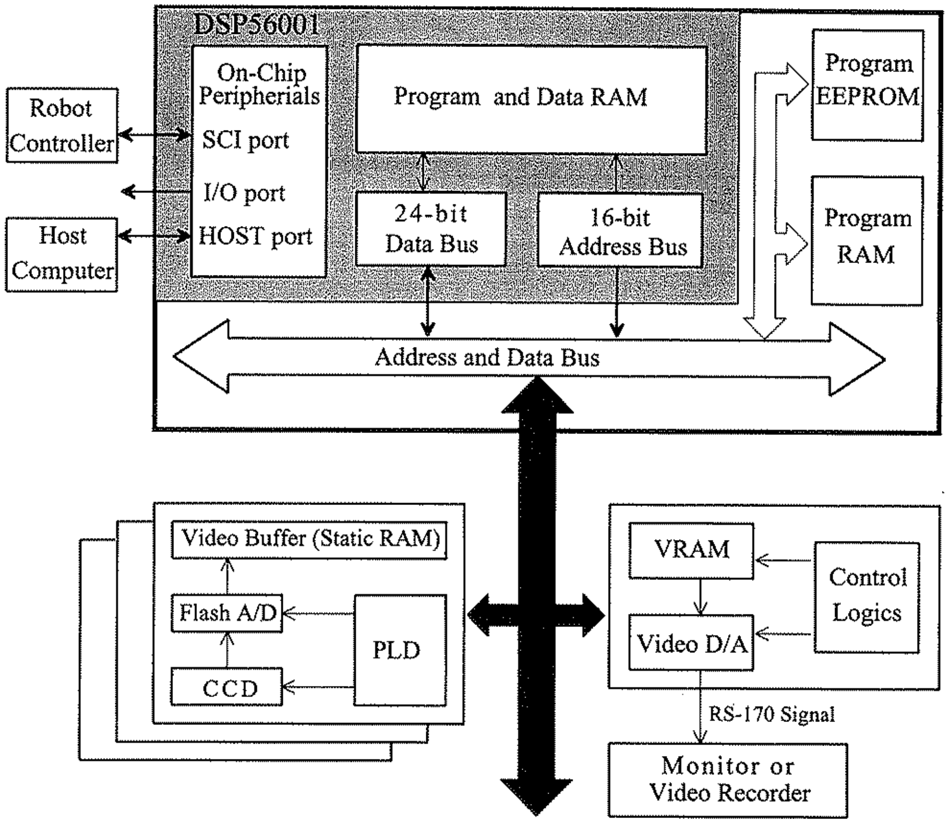

Figure 1. Schematics of a FIVS with three CCDs.

rithms. For these reasons, a digital signal processor (DSP) chip has been chosen.

The DSP-based control board is designed to communicate with several option boards in parallel to tailor the system for a number of applications. Each option board is controlled independently by a programmable logic device (PLD) that receives a peripheral select signal, a read/write signal, and an address signal from the microprocessor control board. Typical examples of the option boards for the FIVS are one or more digital video heads, a real-time video record/display/playback board, an expandable memory board, and a second co-DSP processing board.

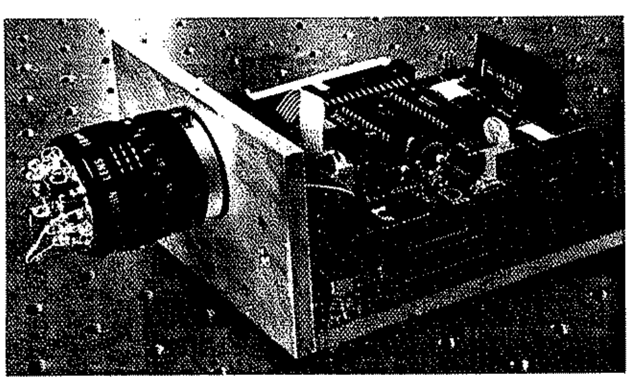

Figure 2. Prototype FIVS.

The video head consists of an $m \times n$ CCD array the output of which is conditioned by a high-bandwidth amplification circuitry. The amplified output is then sampled by a "flash" analog-to-digital converter (ADC). The DSP-based control board provides a direct software control of CCD array scanning and integration time, the intensity of the collected illumination, and the real-time execution of an user-selectable vision algorithm imbedded in the electrically erasable programmable read-only memory (EEPROM). In operation, the PLD decodes the control signals to initiate row shifts and column shifts in response to commands from the DSP-based control board. Particular row shifts and column shifts enable retrieving only a specific relevant area from an image. The PLD also provides control signals to the ADC for performing the analog-to-digital converison synchronized with row shifts, and it enables the video buffer when the DSP reads or writes data to the video RAM (VRAM).

Unlike the off-the-shelf conventional RS170-based systems or the LTS/IVS,[8] which strictly require pixel data to be stored in a video buffer before processing of the pixel data can commence, the FIVS design provides an option to completely bypass the video buffer. Thus, the design offers a means to process and/or to store the digitized pixel data by directly transferring the ADC output to the DSP. For real-time vision-based object-tracking and motion-control system applications, the scheme represents a significant saving in time and video buffer size required for processing an image. As an illustration, consider an image array of $m \times n$ pixels. The time needed to store the entire image (with no computation) in a memory at K MHz is $(m \times n)/\mathrm{K}$ μs and requires $(m \times n)$ bytes of memory. Typical array size of a CCD ranges from 200×160 to 4096×4096 of pixels. The corresponding video buffer and time required simply to store the entire image at a clock rate of 10 MHz would range from 32K bytes to 16 Mbytes and 3.2 ms to 1600 ms, respectively! Clearly, the option to completely bypass the video buffer offers a potentially useful solution to eliminate the prerequisite of frame storage that is often required in off-the-shelf systems. Furthermore, this scheme completely eliminates the special hardware needed in acquiring the digitized pixel data for storage.

A digital-to-RS170 signal conversion board, referred to here as a RS-170 board, is used for converting the digitized image data to the standard National Television Standards Committee (NTSC) TV signal. In the LTS/IVS, which has no digital-to-RS170 conversion, the digitized image data was sent to the computer through the RS232C serial communication channel. The transfer rate is quite slow and thus a considerable amount of time will be wasted during the initial setup, calibration process, and test of a particular configuration. With this interface, the CCD camera can display the real-time image on the TV screen. Furthermore, the signal can be recorded by a video tape recorder, so that upon some failure in a task-performing routine, the events/scenes that lead up to the failure omission are available for off-line analysis.

A memory expansion board can be added to the FIVS to increase its total available data memory. Several memory banks can be placed on the memory expansion board. By using the address bus and the chip select lines of the DSP, several banks of memory can be enabled at any one time. User software then can write a command to enable a particular bank of memory. These banks can be selected in sets, allowing several images to be accessed at any one time.

2.2. FIVS Software

The software has several functions. The first function is to give users the flexibility to control the CCD array scanning, integration time, and the intensity of the illumination. With the CCD under software control, partial frames can be "captured" instead of the customary full frame, reducing the cycle time required to capture and process an image. The ability to shift out partial frames is ideal for high-speed tracking applications where the approximate location is known from a prior image. By reducing the time to capture an image, the effective frame rate is increased. For example, shifting out $\frac{1}{4}$ of an image can increase the frame rate up to 480 fps, not including the time required for illumination and image processing. This frame rate is 16 times the rate achievable from the RS-170 standard.

The second function is to offer an option to process the pixel data from the ADC directly without having to store the pixel data prior to processing. Although windowing process methods have been suggested to perform object tracking under software control, these methods require that a partial window be stored before scanning can begin. The differences between the direct computation and the windowing process for object tracking are as follows: (1) In the windowing process, the entire image must be stored and analyzed at least once before any subsequent windowing process can be performed to provide a reasonable estimate of the object location. Furthermore, if the initial field of view does not contain the object, this estimate must be repeated until an approximate area containing the object can be reasonably found. This prerequisite of storing the image is not necessary if the pixel data from the ADC can be processed directly. (2) After the initial estimate, a fixed window, which must be sufficiently large to include the object in the field of view, must be specified in the windowing process. In most conventional systems, as well as the IVS,[8] which has the output of the ADC connected to the video buffer directly, any partial frame of the image as specified by the window must be stored prior to processing. By directly transferring the ADC output to the DSP, the windowing storing process is eliminated, and a significant fraction of time can be saved. This function provides an attractive feature to vision-based motion-control applications.

The third function allows image processing to be performed in real time without a host computer. The algorithm that allows the user to customize the system for a specified task is preprogrammed in the EEPROM. Because it is impractical to pre-program every possible vision processing algorithm into the FIVS camera, it is desirable that the system can be reprogrammed easily. The main kernel provides a user interface whereby the user can customize the real-time processing for a particular task, from a library of algorithms. This function also provides an effective means to resolve software implementation issues prior to an on-line application. By previewing images of a sample part, the user may select an appropriate vision algorithm for an accurate computation of the location and orientation in real time. Once the algorithms and data are downloaded into the on-board EEPROM, the FIVS can function as an intelligent sensor and communicate directly with the robot controller without a host computer.

The fourth function, which incorporates a real-time display, allows the process controller to setup, to calibrate the vision system, or to analyze a failure mode (if any). The flexible integrated vision system is calibrated by using the technique described by Tsai[10] to correct for the radial distortion of the lens. The coordinate transformation between the camera and the gripper is determined by using the eye-on-hand calibration approach outlined by Lenz and Tsai.[11,12]

3. FIVS PROTOTYPE AND EXPERIMENTAL RESULTS

A prototype FIVS that was developed at Georgia Tech is shown in Figure 2. The goals of the laboratory prototype were to demonstrate proof of concept feasibility, to serve as a test bed for the development of system functionalities, and to assist in achieving an optimal design. Thus, emphasis has been placed on the design of the appropriate circuitry that would provide maximum flexibility in implementing a variety of application software for a given hardware configuration.

3.1. Prototype FIVS

The prototype FIVS has five basic functional modules: (1) An on-board computer consisting of a microprocessor and its associated EEPROM, scratch RAM, and communication hardware. (2) A video head that includes an image sensor, a high-bandwidth signal-conditioning amplifier, an analog-to-digital (A/D) converter, and a video RAM (VRAM). (3) An optic system that houses the lens (or simply a pin-hole) and its associated illumination. (4) An off-line host-interface with a RS17O-board that allows the user to carry out calibration, perform image analysis, and implement application-specific software through a host computer. (5) A memory expansion board that allows multiple digital images to be stored if needed.

A Motorola 56-bit General Digital Signal Processor DSP56001, which has internal program and data memories to allow the processor to execute instructions and fetch data without the overhead of going through the external data bus, was chosen to illustrate the design concept. The DSP56001 has several versions that can run at different speeds ranging from 20 to 33 MHz. The address bus is 16 bits and the data bus is 24 bits. The internal accumulators are 56 bits. It has 64K words (24 bits) of program memory space and two separate 64K words of data memory. With two banks of 64K words of RAM, the DSP is well suited for stereo and 24-bit color sensing. It also has 24 I/O pins that can be configured for various functions such as serial communications and/or parallel I/O. The microprocessor also has 512 bytes of internal program RAM and 2 banks of 256 bytes of internal RAM, which allow code and data to be fetched in parallel without the overhead of going through the external data bus. Up to 15 software-programmable wait states can be set for each memory bank to accommodate slow memory such as program ROM. Running with a 20 MHz clock, the DSP56001 executes instructions at 10 million instructions per second (MIPS), and calculates a 1024-point complex FFT in 3.39 ms. With this computing power, the processor can perform quite complex tasks in a short period, which is crucial for real-time applications.

The video head has been designed around the Texas Instruments TC211 charge-coupled-device (CCD). The TC211 is a small area image sensor with 192×165 active pixels. The output of the CCD is shifted out at 4 MHz and is conditioned by a high-bandwidth amplification circuitry. The amplified signal is then sampled by a "flash" ADC. A Motorola MC10319 ADC capable of sampling at up to 25 MHz is used in this design. This board also features a driver for controlling the illumination, such as LEDs or a strobe. The ADC and CCD are primarily chosen for ease of comparison between the two design concepts, namely LTS/IVS and FIVS.

The optics consist of a 16 mm C-mount lens and 12 HP4101 AlGaAs LED lamps (100 mW each), which are evenly spaced at a 25.4 mm diameter from the center of lens. The duration of the LED illumination and the time over which the CCD integrates the irradiance on its surface can be individually controlled.

A memory expansion board can be added to FIVS to increase its total available data memory. The DSP56001 only has a 16-bit address bus and three chip-select lines, enabling it to address up to three banks of 64 kwords (kw) at any one time. A programmable address decoding scheme is implemented that enables banks of 32 or 64K words (kw) to be individually enabled or disabled. The programmable address decoder/bank enable (PAD/BE) is a memory-mapped device that is "programmed" by the CPU to remap addresses to a particular memory bank. Several memory banks can be placed on an option board with a PAD/BE. The PAD/BE is given a set of two addresses in the DSP56001 memory map, located in the upper 64 words of the Y data memory. User software then can write a command to one of these two addresses to enable a particular bank of memory. Writing to the first address selects two 32kw memory banks to be remapped to X memory, while writing to the second address selects two 32kw memory banks to be remapped to the Y memory bank. These banks can be selected in sets of 32kw, allowing up to 4 different small images (192×165) to be accessed at any one time. Images can be captured directly to these remapped memory banks as if they were normal VRAM.

To provide visual output in a laboratory environment to assist the initial setup and calibration process, a RS-170 board was built. This RS-170 board uses an EPROM incorporated with the counter circuits to generate those timings required for the RS-170 video standard. Because the RS-170 standard uses an interlaced scheme to display video images, the timing of the synchronization is quite complex. To accommodate this requirement, a 32 Kbyte EPROM and two sets of counters, one for horizontal timing and the other for vertical timing, are used to address individual pixels and each horizontal scan line. The horizontal counter is responsible for addressing each pixel and generating the required horizontal synchronization signal. The vertical counter is used to address each vertical scan line. There are a total of 525 vertical lines in the NTSC TV system. Therefore, a 10-bit counter is required for the vertical counter. The display of a real-time image requires rapid updating of the image data from the processor and fast retrieving from the display circuit. To satisfy this requirement, a VRAM is used as the image buffer. The advantage of using VRAM is that its contents can be accessed from two ports, one parallel and one serial, simultaneously, without getting into the bus contention.

3.2. Experimental Results

Several experiments were conducted on a prototype FIVS with a 20 MHz Motorola DSP56001. An Intel-386 33 MHz personal computer was used as a host, which limits the serial communication rate to 115.2 Kbaud non-conventionally. The FIVS has been designed to communicate with a host computer through an RS-232 communication port.

An example algorithm has been written to illustrate the design concept as follows:

1. In general, the host requests the FIVS to perform a task by sending a character.
2. The FIVS acknowledges the request by returning the character. It then performs the task following the instructions pre-programmed in EEPROM. A typical embedded software program consists of the following steps: (1) Once the FIVS acknowledges the request from the host, the imaging sensor integrates the irradiance over the imaging surface for a specified time. (2) The data shift-out of the CCD are digitized. The FIVS performs direct processing or stores a partial frame of the digitized image before processing, depending on the specific application example. (3) The location and orientation of the part are then computed using a user-specified algorithm.
3. Once done, the FIVS signals the task completion to the host by responding with a character.
4. The computed data are then transferred to the host serially.

3.2.1. Example 1

Figure 3 (Image 1) shows an image of two through-holes of an electrical part, which characterize the location and the orientation of the electrical part. The sizes of the two blobs (in Image 1), which correspond to the two through-holes, are 250 and 70 pixels. An embedded fiducial locating algorithm has been written to scan for any isolated blobs as fiducials, identify the targeted fiducials based on the size of the blobs, compute the centroid of the fiducials, and locate the coordinates of the part. The FIVS outputs the follow-

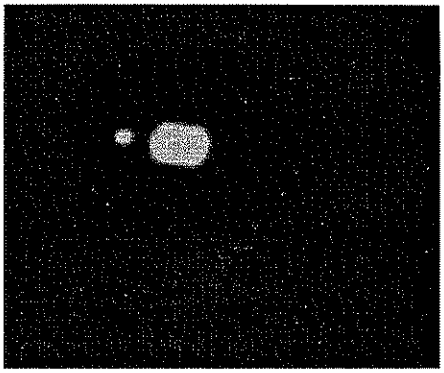

Figure 3. (Image 1) Fiducials characterizing an electrical part.

ing data to the host: the area (1 byte or 2 characters), the peak grey value (1 byte), the centroid (4 bytes), and the second moment (4 bytes) of the two blobs.

The benchmark estimates the cycle time required for the FIVS to "capture" and store a partial image frame of 169 × 135 pixels (68% of a full frame), locate the two blobs, compute the output data, and communicate the computed data to a host computer at 115.2 kbaud. The total cycle time required to perform the task was 31.4 ms, as shown in Table I. In other words, the specific task was performed at a rate on an order similar to that specified for the conventional RS-170 standard, which simply defines the read-out at a rate of 30 fps without frame grabbing and computation.

The following factors contribute to the short cycle time: (1) The ability to capture a partial image frame and store it directly into the VRAM results in a significantly higher rate of image frame transfer. (2) Because the computation of the centroid/orientation of the image is performed by the on-board DSP, no image transfer between the FIVS and the host is necessary. (3) As shown in Table I, a significant portion of the total cycle time is taken by the computation of the fiducial location, which depends significantly on the image size and the number and size of individual blobs, and thus the application. The DSP enables the fiducial location to be rapidly computed without relying on a host. (4) Only the computed data are required to be transmitted to the host through the high-speed serial communication of the DSP. It is worth noting that the serial communication rate of 115.2 kbaud was limited by the host computer used.

The FIVS performance is compared with the published performance of LTS/IVS[7] for a similar task. Both systems used the same CCD and ADC for ease of scientific comparison. The LTS/IVS communicates through a RS232 at a baud rate of 9600 and requires a full frame transfer of pixel data to the video buffer prior to any image processing. The published data for the LTS/IVS[7] has been based on locating a single landmark of 240 pixels. The comparison is given in Table 1.

3.2.2. Example 2

In some applications, it may be desirable to perform an image segmentation and preserve the data so that other information (edges, inertia, etc.) can be obtained. The segmentation data then can be used for subsequent image processing such as object verification and computation of the part location/orientation. An algorithm based on region-growing segmentation via thresholding[11] that requires binarization has been written for illustrating the FIVS operation.

Figure 4 (Image 2) displays the silhouette of a machine component along with an unknown foreign particle. The sizes of the two blobs are 3260 and 320 pixels. The task is to discriminate the machine component from the foreign particle and to compute the location of the machine component. Table II summarizes the cycle time required to complete the part-locating task (binarization, segmentation, and centroid computation) by using region growing segmentation for Image 1 (Fig. 4) and Image 2 (Fig. 5). The binarization and data transfer between the

Table I. Experimental cycle time of fiducial locating.

Process	Time (ms) FIVS	Time (ms) LTS/IVS[7]
Serial communication	0.8	9.4
Illumination/integration time	10.00	21.5
Frame transfer from CCD to video RAM	5.80	8.25
Computation time	13.00	38.00
Data transfer to the host	1.80	22.00
Total	31.40	99.15

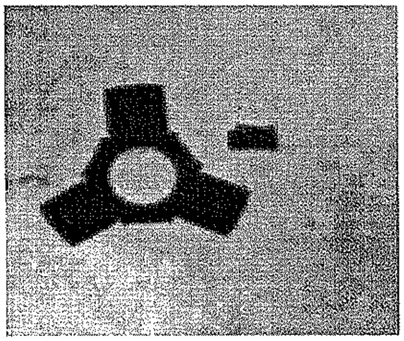

Figure 4. Silouette of a machine component.

processor and the RAM depend significantly on the image size. As shown in Table II, the time required to process the full frame of Image 2 (192 × 165 pixels) is 242 ms as compared to 70 ms of partial frame of Image 1 (169 × 135 pixels).

For the same captured Image 1 of 169 × 135 pixels, the region-growing segmentation via thresholding requires a cycle time approximately five times higher than that computed using a fiducial-locating algorithm. The significant increase in time required for the image processing and computation is as follows: The fiducial locating algorithm simply scans for any isolated blobs and stores only limited data about the isolated blob. Thus, for an image that has relatively few isolated blobs, the data transfer between the processor and the RAM is minimal. In contrast, the region-growing segmentation algorithm preserves all the segmentation data in RAM for subsequent image processing needs and requires binarization.

In both examples, it is expected that the computation and data transfer rate can be increased significantly by means of a DSP chip with higher processing speed. The choice of the algorithms, clearly, depends on the applications. The ease and flexibility to program the embedded software so as to customize the FIVS for a specific application without costly hardware redesign is essential.

3.2.3. Example 3

This example concerns motion-control applications where the typical sampling time is on the order of 1 ms. Simply storing the image of 200 × 165 pixels (with no computation) in a memory operated at 4 MHz requires 8.25 ms and 32 KBytes of memory space. Clearly, the time required to store the digitized data in the video RAM becomes a significant fraction of the sampling time.

Unlike Examples 1 and 2, where a partial frame of the digitized image from the A/D converter is stored before any image processing is performed, the following example shows the experimental results by using direct computation of pixel data from the A/D converter without having to store the digitized image in a video buffer.

Figure 5 shows a typical image of a grid pattern, which was captured by a CCD viewing through a pin-hole of 35 μm. The image is a portion of a grid pattern consisting of concentric circles with 1 mm spacing and radial lines spaced at one degree apart. For the viewing area of 4 mm × 4 mm, the resolution is on the order of 25 μm. The image analyzing task is to locate the intersection between the optical axes and the grid pattern (as shown in Fig. 5). The cycle time required to complete the image analyzing task is summarized in Table III. Because no data is transmitted serially, the cycle time required for analyzing an image (96 × 82 pixels) was found to be on the order of 5 ms.

4. APPLICATIONS

The flexible integrated vision system has potential applications where sophisticated vision interpretation is required, yet low cost, high speed, accuracy,

Table II. Experimental cycle-time for image segmentation.

Process	*Image 1 (Time in ms)*	*Image 2 (Time in ms)*
Serial communication	0.80	0.80
Illumination/integration time	10.00	10.00
Frame transfer from CCD to video RAM	5.80	8.25
Binarization	16.00	22.00
Segmentation and centroid computation	54.00	220.00
Data transfer to the host	1.80	2.00
Total	88.40	263.05

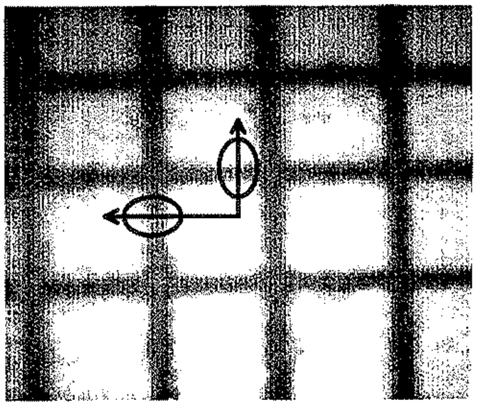

Figure 5. Typical image of a grid pattern.

Table III. Experimental cycle-time for analyzing image of a grid pattern.

Process	*Time (ms)*
Integration time	2.00
Frame transfer	2.05
Computation	0.84
Total	4.89

and flexibility is demanded. The following illustrates two application areas where FIVS can be effectively employed as an integrated as well as an intelligent component for robotic application or in an opto-mechatronic design.

4.1. Robotic Part-Feeding

The flexible integrated vision system can be effectively incorporated as a part of the part-feeding system that identifies and locates the part, and directs the robot to feed parts from egg-crate-style pallets to subsequent processes such as machining or assembly processes. The application of FIVS for flexible part-feeding is illustrated in Figure 6. The function of the robot controller is to serve as a host and thus control all system activities. The robot is driven by on-line sensory information from the flexible integrated vision system, which is mounted on the arm of the robot. Based on the eye-on-hand calibration, the camera-gripper relationship can be determined off-line, and the pallet-robot relationship can be determined each time a new pallet is used.

The part-feeding can be addressed here as a model-based vision problem because parts to be handled are often known a priori. The goal of the model-based vision algorithm is to use a precompiled description (template) of the part to verify or to recognize an object in the field of view. The part-presentation algorithm[14] begins by identifying the pre-specified features, such as fiducial marks on the part or simply the sharp curvature on the edge of

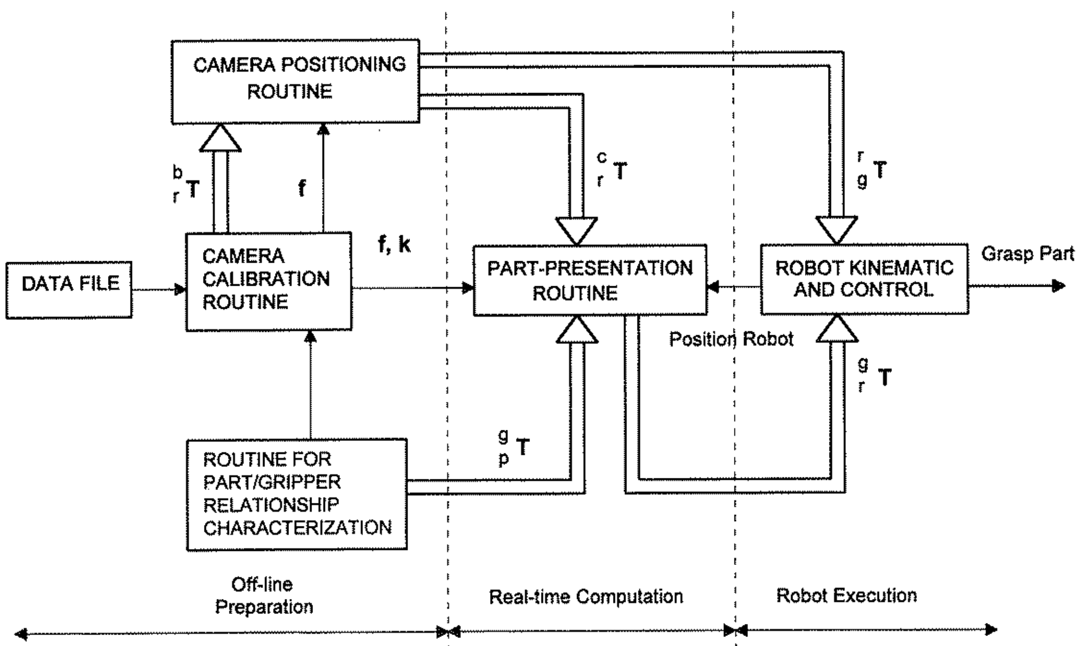

Figure 6. Example system organization.

the part. Once the part has been identified, the part location is determined using the user-specified algorithm, and sent to the robot controller.

For flexible manufacturing, where a large variety of product sizes and component types are encountered, the robot part-feeding system[7] must have the ability to adapt to a changing product design without costly hardware redesign or time-consuming software re-engineering. To achieve system flexibility and to resolve software implementation before on-line applications, the off-line image analysis package can be used. It allows the process planner to preview images of a sample part. A suitable method then can be selected from a vision library for generating templates and for an accurate computation of the location and orientation of the part. The templates, calibration data, and geometric relationship between the part and the gripper can be tested before implementation. It is worth noting that once the algorithm and data are down-loaded into the on-board EEPROM, the FIVS can function as an intelligent sensor and communicate directly with the robot controller without a host computer.

4.2. Opto-Mechatronics

Because the central control unit of the FIVS can perform real-time computation without relying on a host, the concept of FIVS allows the compact optoelectronic hardware to be used as an integrated part of an intelligent mechanical and/or electro-mechanical device. The following illustrates an interesting application where FIVS is used as an optical wrist sensor for measuring the combined roll, yaw, and pitch motion.

For multi-dof motion, single-axis shaft encoders are often used with mechanical linkages in measuring the orientation. For a ball-joint-like device such as a spherical motor,[13] which is characterized by its isotropic and smooth motion and has no singularity within its workspace, conventional shaft encoders are undesirable due to the friction and inertia effects introduced by the mechanical linkages. One proposed design for an optical wrist sensor for measuring three dof orientation uses two imaging sensors for tracking a grid pattern on the motor.[16] To be effective as an encoder, the system must be small, low cost, fast, and accurate. The FIVS has the unique capabilities to satisfy these requirements.

The FIVS that serves as an optical wrist sensor of the ball-joint-like spherical motor allows the following functions to be effectively achieved: (1) It directly controls the operation of the two CCDs. With the ability to eliminate the need to store the digitized image, the FIVS can track the orientation of the spherical motor at speeds much higher than would be possible with any conventional vision approach. (2) It performs the image processing, computation of the wrist orientation, and control of the spherical motor using a single on-board DSP processor. (3) It handles the data communication between the integrated spherical motor and an external system controller.

The FIVS design architecture offers a means to perform direct computation of the controlled variables and controlled input without having to transfer pixel data through the host bus, and thus has potential contributions to real-time vision-based motion-control applications.

5. CONCLUSIONS

Unlike the video camera, which outputs an RS-170 video signal and requires a separate frame-grabber, a DSP board, and a host computer, the FIVS performs all processing directly. By combining the imaging sensor control, collocated illumination, direct digitization, computation, and data communication in a single unit physically no larger than a typical video camera, the cost of the FIVS is significantly less than that of the conventional vision system, which has significant overhead to support redundant system hardware such as memory, clock synchronization, system bus, data communication, and power supply. More significantly, the considerable bottleneck of transferring large amounts of data through the data bus of the host computer is avoided.

This article presents the design concept and system architecture to provide a means to perform direct computation of pixel data without having to transfer pixel data through the host bus. A prototype flexible integrated vision system, which offers an option to completely bypass the video buffer, has been presented and demonstrated experimentally. By providing an effective means to provide a direct communication between the ADC and the DSP, the need to store pixel data prior to image processing is eliminated, thus demonstrating the significant potential of FIVS for real-time motion-control and object-tracking applications.

This research was supported by the National Science Foundation (NSF) through the NSF Presidential Young Investigator Grant No. 8958383. The donation of the DSP 56001 Development Kit by Motorola Inc. in supporting this work is greatly appreciated.

REFERENCES

1. EIA Standard, "Electrical performance standards—Monochrome television studio facilities," Electronic Industries Association, Engineering Dept., 1957.
2. G. R. Archer, "Vision in programmable assembly systems—Results," *Proc. of 5th International Conference on Assembly Automation*, May 22–24, 1984, Paris, France.
3. P. Dunbar, "Machine vision," *Byte*, Jan. 1986, pp. 161–173.
4. K-M. Lee, "A low-cost flexible part-feeding system for assembly automation," *Proc. of the ICARCV International Conference on Automation, Robotics, and Computer Vision*, Singapore, 1990.
5. S. Dickerson, E.-H. Lee, D.-R. Li, and T. Single, "Landmark tracking system for AV/RS," MHRC Final Report, 1990.
6. K.-M. Lee et al. "A low-cost flexible part-feeding system for machine loading and assembly," MHRC Final Report, 1992.
7. K.-M. Lee, "Flexible part-feeding system for machine loading and assembly, Part I: A state-of-the-art survey; Part II: A cost-effective solution," *International Journal of Production Economics*, **25**, 141–168, 1991.
8. S. L. Dickerson and K.-M. Lee, Image Reading System, U.S. Patent No. 5,146,340, September 8, 1992.
9. R. T. Chin and C. R. Dyer, "Model-based recognition in robot vision," *Computing Surveys*, **18**(1), 67–108, 1986.
10. R. Y. Tsai, "A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf TV camera and lenses," *IEEE Robotics and Automation*, **RA-3**(4), 323, 1987.
11. R. K. Lenz and R. Y. Tsai, "Calibrating a Cartesian robot with eye-on-hand configuration independent of eye-to-hand relationship," *Proc. of Computer Society Conference on Vision and Pattern Recognition*, Ann Arbor, MI, 1988, pp. 67–75.
12. R. Y. Tsai and R. K. Lenz, "A new technique for fully autonomous and efficient 3D robotics hand/eye calibration," *IEEE Transactions on Robotics and Automation*, **5**(3), 345–358, 1989.
13. D. H. Ballard and C. M. Brown, *Computer Vision*, Prentice Hall, Englewood Cliffs, NJ, 1982.
14. K.-M. Lee and S. Janakiraman, "A real-time part-presentation algorithm using retroreflective vision sensing," Presented at the SME Applied Vision Conference, Atlanta, GA, 1992.
15. K.-M. Lee and C. Kwan, "Design concept development of a spherical stepper motor for robotic applications," *IEEE Journal of Robotics and Automation*, **7**(1), 175–181, 1991.
16. K.-M. Lee, G. Meyouhas, and R. Blenis, "A machine-vision-based wrist sensor for direct measurement of three degrees-of-freedom orientation," *Mechatronics*, **3**(5), 571–577, 1993.