The IBM 3495 robotics and vision system

by G. T. Kishi

The IBM 3495 Tape Library Dataserver is an automated tape library that consists of one to four IBM 3490 Magnetic Tape Drive Subsystems, a Library Manager computer, a storage enclosure, and a tape cartridge accessor. The accessor incorporates a combination of IBM and OEM robotics and vision systems with application programs and special fixtures that provide automatic teaching and updating for up to 18 000 cell locations in the library. Command queueing allows continuous robotic motion, removing the performance penalty of communications with the Library Manager computer. The application program written for the vision system compensates for label tilt and spatial translation. A neural network with special filters ensures the quality of the labels read.

Introduction

The IBM 3495 Tape Library Dataserver (**Figure 1**) is an automated tape library for users who want full-service attachment to IBM host computers. The 3495 provides storage from 6000 to 18 000 cartridges and attaches to IBM 3490 tape drives. These cartridges represent up to 45 terabytes of storage when enhanced-capacity cartridges and data compression are used.

The 3495 consists of one to four 3490 tape drive subsystems, a Library Manager computer (a modified IBM PS/2® Model 95), a storage enclosure, and a tape cartridge accessor. Commands are issued by the host computer to the 3490 control unit, which sends them to the Library Manager. The Library Manager converts and optimizes the host commands into cartridge accessor commands. These accessor commands are prioritized, queued, and sent to the cartridge accessor.

The cartridge accessor consists of an OEM robot mounted on an IBM-designed vehicle and uses an OEM vision system. This paper discusses the adaptations and improvements of the OEM systems that have made the 3495 a successful tape library system.

Hardware design

The IBM 3495 earned a Gold 1993 Industrial Design Excellence Award (IDEA) from the Industrial Designers Society of America (ISDA) [1]. The primary design goal of the 3495 Dataserver was to provide a reliable, automated tape library system for up to 18000 cartridges. The reliability of the 3495 was guaranteed by selecting an OEM robot with proven field performance. This design objective ensured long product life.

This requirement, coupled with other design requirements such as maximum library height and number of vertical cartridge storage rows, led to the selection of a six-axis anthropomorphic robot. The OEM robot chosen had proven field performance in mechanical assembly, machine loading, material removal, and other robot tasks.

A custom vehicle was designed to transport the robot along the length of the library, and storage racks were designed to hold the cartridges. Special storage rack sections were designed to allow IBM 3490 tape drive units to connect to the library. A transparent roof was provided to enclose the library, to provide a dust-free environment, and to allow ambient light to enter the aisle.

0018-8646/96/$5.00 © 1996 IBM

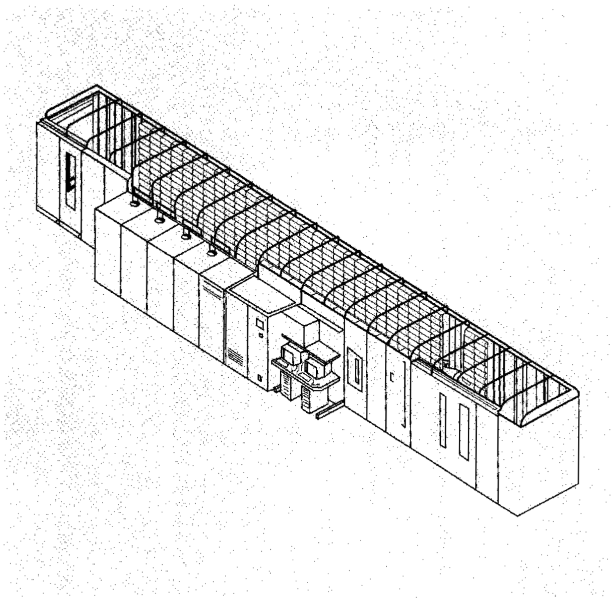

Figure 1

The IBM 3495 Dataserver.

The robot provides a rotating, end-of-arm tooling mount. Two cartridge grippers and up to two cameras are located on a turret plate attached to this mount. This design provides a cartridge-swapping operation for efficient accessor motion. Three-dimensional space requirements constrained the lamps for each camera to be coaxial with the camera (**Figure 2**). Using a six-axis robot allowed cartridge picking within the minimum vertical clearance required by the library design.

Controller hierarchy

There are three interconnected main controllers in the IBM 3495, as shown in **Figure 3**:

- *Library Manager computer* This computer receives commands from the host computer through an IBM 3490 control unit. The Library Manager prioritizes and optimizes these commands, thus creating the appropriate accessor commands. A typical accessor command includes
  - A three-character command acronym.
  - A command identification number.
  - Accessor coordinates.
  - A cartridge label.
  - A gripper number.
  - A tool orientation reference.
  - A command-type identifier.

  The physical coordinate database for all of the cell locations in the tape library is kept in the Library Manager computer, which reduces the memory requirements on the accessor controller.
- *Accessor controller* This controller receives accessor commands from the Library Manager, performs the requested action, and returns the status of the completed command to the Library Manager. If necessary, the controller commands the vision controller to read the cartridge label, or to take any other appropriate action.
- *Vision controller* This controller receives vision commands from the accessor controller, performs the requested vision function, and returns the appropriate data.

Accessor control

The accessor microcode was designed to optimize cartridge handling. Each command from the Library Manager computer follows this sequence:

1. Current command sent from Library Manager.
2. Current command accepted (or rejected) by accessor.
3. Accessor starts moving to the commanded position (if necessary).
4. Next command sent by Library Manager.
5. Current command performed by accessor (after it is in position).
6. Next command "prefetched" (if available) and motion started to next command position.
7. Current command results sent to Library Manager.
8. Next command becomes current command.
9. Steps 2–8 are repeated until the last command is received.
10. The process is repeated starting at step 1.

This sequence allows the accessor to queue one command from the Library Manager while it is performing a cartridge-handling command. The accessor also communicates command results while the accessor is moving to the next command position. These features eliminate any pauses in motion due to communications overhead with the Library Manager, contributing to the "surprisingly fluid motion" [1] of the system.

Performance is also maximized by overlapping the operation of the vision system with that of the robot. When a cartridge is picked, the label is read by the vision system. After the robot moves the camera in front of the label, the vision system "grabs" the image. The image is processed to read the label while the cartridge is being picked.

The Library Manager enhances 3495 performance by deferring dismounts from the 3490. This process allows a deferred dismount to be matched with the next mount request to create a "swap" operation, reducing accessor motions. If the customer selects "floating" home cell operation rather than "fixed" home cell operation, the Library Manager also swaps cartridges at the rack, which further reduces accessor motions.

If the robot has to retry an operation, its successive attempts are at slightly different Cartesian coordinates. When it determines the correct coordinates for the operation, information is included in the response to the Library Manager to update the library coordinate database, a technique referred to as "programmed recalibration" [2].

Efficient accessor motion had to overcome two problems:

1. *A round robot in a square hole* A six-axis robot has natural motions that are convex arcs about its center. The 3495 design places the robot inside a rectangular parallelepiped. We placed a round robot in a square hole. Because it is critical to keep the robot's natural arcs from hitting the walls, we separated each of the different general types of motions that the robot had to make and tailored a routine to handle each motion (for example, left-side rack to left-side drive, and right-side rack to left-side drive).
2. *Interdependent robot and vehicle motions* The robot's operating system allows either simultaneous robot and vehicle operation or independent motion, but not both at the same time. This means that the vehicle and the robot motions must be coupled while the vehicle is moving along the track. The vehicle motion is divided into ten parts, which allows the coupling to happen at any multiple of 10% of the vehicle motion.

   When the vehicle and the robot motion are coupled, the slower of the two mechanical devices limits the speed of the coupled motions. Ideally, all motions are limited by the vehicle, which indicates that the accessor is moving at its maximum speed down the aisle.

   Each type of motion has its coupling points set empirically; for example, the right-side rack to left-side drive motions are coupled as follows:
   - *0% of vehicle motion* The robot arm is withdrawn a safe distance from the rack to allow swinging the arm in convex arcs.
   - *20% of vehicle motion* If the robot must be either raised or lowered to clear the rail or the roof before being free to spin its grippers, it is moved to this position during the first 20% of the vehicle motion.
   - *50% of vehicle motion* The robot arm is moved to a safe position (center of track).
   - *80% of vehicle motion* If a cartridge is going to be inserted into the drive, the robot is moved to align the cartridge for insertion when the vehicle motion is 80% complete. If a label is going to be read or a cartridge is going to be picked from the drive, the robot is moved to align the camera to read the label when the vehicle motion is 80% complete.

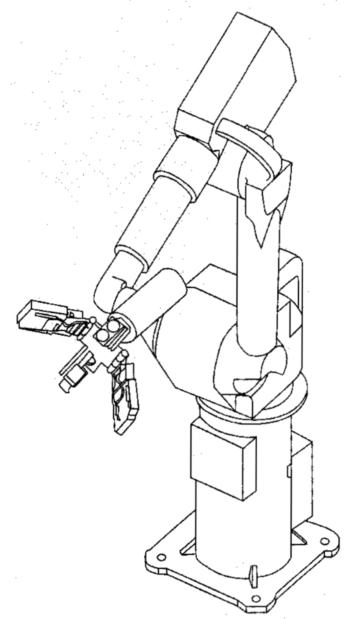

Figure 2

Robot and tooling.

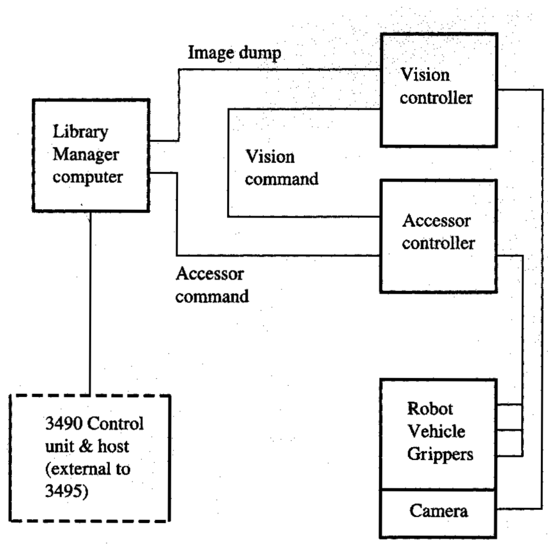

Figure 3

Controller interconnection.

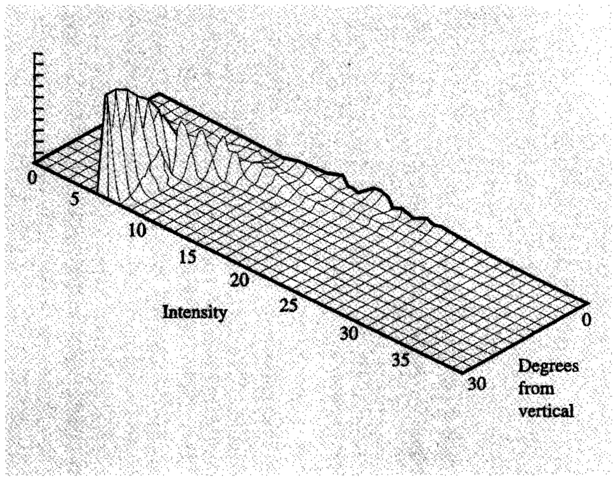

Figure 4

Histograms of rack surface light intensity at camera angles from 0° to 30°.

- *100% of vehicle motion* The vehicle completes its motion with the robot already in position, which reduces the settling time required for the robot motion.

This coupling allows the vehicle to move at full speed the majority of the time, which maximizes the cartridge-handling throughput of the accessor.

Vision system viewing angle

Because the camera and the lamps are colinear, flat objects that are normal to the camera's viewing axis cannot be viewed without excessive glare. To determine the minimum off-axis viewing angle for the camera and the lamp assembly, an experiment was performed.

A camera and a light were mounted on a test stand. A sample of the material used in the injection-molded rack magazines was placed at the viewing distance in front of the camera. The sample was initially normal to the camera. An intensity histogram was taken of this surface, which was chosen because it provided a large number of pixels for the histogram and was representative of the rack surface that the camera would be viewing during teaching and label-reading operations.

The rack was rotated 2.5° away from the camera. Another histogram was taken. This process was repeated until the rack was rotated 30° from the camera. This series of histograms is shown in **Figure 4**. The average intensity is reduced as the rack target surface is rotated away from the camera.

The average values of each of these histograms were analyzed to determine the minimum angle at which the glare from the target surface was minimized. Different sets of measurements were made to ensure that there were no significant errors in measurement. We determined that 15° was the minimum angle at which glare was reduced to a level where it would not affect label reading or teaching operations.

The angle chosen for use in the 3495 design was 20° off the camera axis, which included the 15° minimum glare reduction angle and an extra 5° of error for design tolerance.

Teaching the robot

The 3495 must be able to direct its accessor accurately to any of up to 18000 cells in the system. These cells are grouped by their logical locations and can be accessed at random by the host computer, which makes the accessor program function as if it were programmed off-line. The ideal robot to use for off-line programming is highly accurate [3]. The 3495 robot, like other industrial robots, is highly repeatable, but not as accurate as it is repeatable. A method of compensating for this inaccuracy was required.

Much research is currently underway on methods for improving a robot's accuracy to the same tolerance as its repeatability. Most of the research to improve robot accuracy either requires access to the internal robot kinematics, or is computation-intensive [4, 5].

The language supported by the 3495 OEM robot is considered by some to be the best available [6], although it has limitations that are typical for robot control languages. First, user programs are translated; computation-intensive operations are slow relative to the internal machine capability and may cause reduced performance. Second, user access to robot kinematics is prevented. These limitations made it impractical to use kinematics-based solutions on the 3495 to improve robot accuracy; therefore, other solutions had to be developed.

Empirical testing determined that the robot is accurate enough to linearly interpolate cell locations along a single vertical line within its workspace to pick cartridges once the endpoints are determined. Two such lines are defined, one representing the left wall and the other the right wall. Each line is a fixed distance forward from the base of the robot and at the appropriate lateral offset for its wall. All cartridge-handling operations are performed using these lines as a reference position.

Thus, all cartridge-picking operations are performed at a fixed distance from the base of the robot. These picking operations are characterized in order to ensure that the lines chosen avoid both points of singularity (associated with a six-axis robot of this general design) and joint limit errors while performing linear motions.

Using these vertical lines as reference for all cartridge-handling operations and fully characterizing these operations ensured that no motion would fail, regardless of the differences from system to system. The only disadvantage is in those rare operations where an adjacent cartridge is accessed and the vehicle must move prior to picking it, even though the robot has adequate motion to reach it.

Using these references, the robot must be taught the relative locations along the vertical lines and the vehicle locations for every cell in the library. This teaching process has three major steps: automated mastering, automated teach mastering, and the automated teaching process.

Automated mastering Whenever one of the robots used in the IBM 3495 Tape Library is set up in the factory or is altered, it must be mastered. Mastering is the process of physically moving the robot to a known position relative to its base and aligning all six coordinates (x, y, z, roll, pitch, and yaw). This initializes the starting reference position on which all of the robot kinematics are based.

Before the release of the 3495, mastering was done using a fixture with six dial indicator gauges that were zeroed by a service person, who moved the robot tooling head with a control pendant. On the 3495, performing this procedure was not acceptable because the person could not see all of the gauges through the safety enclosure door. Also, the mastering process could have human-induced errors that could damage the 3495.

All of the six manual indicators were changed to digital indicators, a data-gathering unit was added, and a special robot control program was written to automate this mastering process.

During the rare occasions when the robot needs mastering, the fixture is placed on the robot and gauge blocks are used to zero the indicators. The mastering program is invoked, and the robot is moved until all of the gauges are within the expected tolerance. The new mastering values are saved and are used by the robot controller to calculate the kinematics for the robot.

Automated "teach mastering" After the robot is mastered, the teach-mastering process is performed. This process calibrates the camera to the grippers and must be performed whenever their relative alignments may have changed.

The camera is set up to view an object 18 inches in front of the camera. This distance allows the camera to fully view five cartridges in its field of view, but at the same time causes small angular robot errors to be magnified during the teaching process. This camera-to-gripper error must be measured before teaching the system.

Two special teach-mastering fixtures are located in the service bay of the 3495. One fixture is mounted on the left wall and the other is mounted on the right wall. At a height corresponding with the top and bottom of the racks, a teach-mastering point is located. When teach mastering is performed, one of the fingers on the gripper is replaced with a tool that has three proximity sensors.

Each teach-mastering point has two features:

- *Teach targets* The teach target consists of three white 5-mm squares on a dark background. These squares are placed 10 mm apart. This spacing was chosen because no labels appear to have characters or features spaced at 10-mm intervals. The three white squares at 10 mm create a unique teach target that cannot be confused with a cartridge label. The teach targets are designed with a 20° tilt from a plane parallel with a rack wall, eliminating glare, on the basis of the viewing-angle tests described in this paper.

  During the teaching of a teach target, the camera views the target perpendicular to the rack walls. The robot is moved until the teach target is centered in the camera's field of view. This ensures that target parallax is eliminated and the final taught position along the axes parallel to the rack rows and columns is the same no matter how far the robot is from the target.

  Positioning the camera perpendicular to the rack wall eliminates any error due to inaccurate positioning of the camera away from the wall. Variations in the camera-to-rack distance do not affect the accuracy of the teaching process.
- *Gripper calibration target* Two perpendicular slots are machined into the fixture to allow the proximity sensors to find the intersection of the slots.

For each teach-mastering point, the following process is performed:

1. The tool is touched to the fixture.
2. The three sensors are used to flatten the tool on the fixture.
3. A sensor is used to locate both edges of the horizontal slot and determine its vertical centerline.
4. A sensor is used to locate both edges of the vertical slot and to determine its horizontal centerline.
5. Using the intersection of these centerlines, the location of the vision target is predicted.
6. The vision target is viewed in its predicted location.
7. The vision target is centered in the camera's field of view.
8. The vision offset from its predicted location is recorded for this point.

This process is repeated for both rack points on both fixtures. Additional points are used for 3490 drive teach

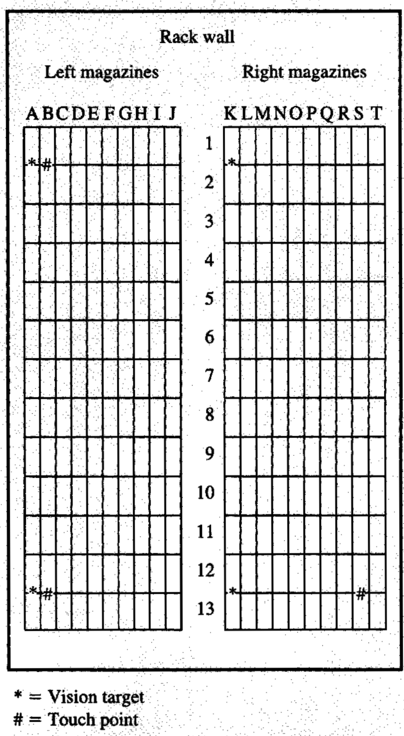

Figure 5

Rack teach locations.

locations and for other types of special hardware on the 3495.

This teach-mastering process calibrates the camera-to-gripper offsets, but only at key points, such as the top and the bottom points for the racks on each side of the 3495. No attempt is made to calculate the correction factors for the robot kinematics—after the endpoints are located, the robot is accurate enough to interpolate the gripper positions along the vertical cartridge-handling line.

Automated teaching process After the teach mastering is complete, the 3495 can be taught. Each of the components in the library is taught individually:

- *Rack walls* Each two-foot section of rack wall is taught individually. Three touch points and four vision targets are used, as shown in **Figure 5**. Each wall section has two vertical columns of magazines (A–J and K–T), and each column has 13 rows of magazines, each holding 10 cartridges.

  Two vision targets are used to determine the location and tilt of each column of magazines. When each target is located by the vision system, the teach-mastering camera-to-gripper offsets for the appropriate point are applied and the adjusted location is returned to the Library Manager.

  Three touch points are used to determine the distance to the rack and to adjust the distance to each cell, assuming that the entire rack face is a plane in space. The gripper is used to touch the edge of a magazine and to determine its distance from the robot.

  The Library Manager computes column tilt and touch plane calculations independently, assuming a quasilinear relationship, but this approximation has been empirically determined to be adequate. The calculated positions of all 260 cartridge cells are stored in the Library Manager database.
- *3490 drives* Every 3490 consists of an A-box (controller) and up to four B-boxes (drives). Each B-box has four drives. The 3495 teaches each drive individually.

  A "fat" cartridge was designed to fit snugly into the integrated cartridge loader (ICL) of the 3490 drive, in the slot just above the feed slot. This fat cartridge has a vision target located in the center of the slot.

  Each drive is taught by finding the vision target to locate its position. The fat cartridge is touched by the gripper to determine its distance from the robot. Because each drive is a relatively small device, it was empirically determined that all of the other cells could be assumed to be vertically in line with the fat cartridge.

  The Library Manager uses the vision and touch data to calculate the positions of all of the cells on the 3490 ICL. These additional cell positions are also stored in the Library Manager database.

Reading labels

Whenever a cartridge is picked, the label must be read by the vision system. Labels are also read during tape library inventory operations. When the accessor is in position to read a label, it sends a label read command to the vision system. Typically, a large accessor motion has just been completed. At the end of this motion, the arm must be allowed to settle and the image must be stabilized prior to "snapping" a picture of the label with the vision system.

During a large accessor motion, the accessor decelerates rapidly to a stop and then indicates that it has finished moving when it has reached a predetermined distance from its goal position. The camera, located at the end of the robot arm, is still oscillating slightly. The vision system is given control at this time.

The vision system waits a predetermined settling time to allow the camera oscillations to dampen and then checks for image stabilization by comparing a specified window in successively snapped frames. When the images in this window of two successive frames appear nearly the same, the last image snapped is kept and the label read operation begins.

The coaxial mounting of the camera and its lamps prevents labels from being read perpendicular to their surface. Because the perpendicular view would cause enough glare from the labels to wash out the center of the label image, the labels are viewed, as determined by the viewing-angle experiments, at 20° off the perpendicular axis.

Five labels are in the camera's field of view. Whenever a single-cartridge vision operation is performed, the middle label is read. During tape library inventory operations, however, five labels at a time are read to increase throughput by reducing robot motion.

The image size, camera focal length, and off-axis view distort the label image and require the vision system software to compensate for this distortion, as shown in **Figure 6**. The labels in this field of view are being viewed from above at a 20° angle. The labels to the right of the center label are rotated at increasing clockwise angles, while the labels to the left are rotated counterclockwise. Because this viewing angle makes the top of the labels closer to the camera than the bottom of the labels, the tops appear larger to the camera.

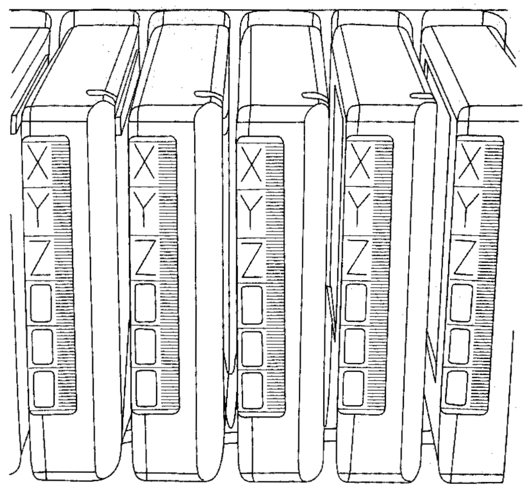

Figure 6

Vision system field of view.

This distortion is handled by the tilt compensation code. Whenever a read is performed, the appropriate label image window is selected. The horizontal center of this window is searched to find the bar code, using a function that locates the most rapid black and white pixel transitions in the vertical direction. When the bar code is located, the tilt information from the most recent successful read in the selected window is used to select the pixels used to analyze the bar code. This analysis is attempted starting at the scan line in the horizontal center of the bar code and alternating outward until the bar code is successfully read.

If the bar code cannot be successfully read using the stored tilt information, two more searches are performed at the top and at the bottom of the bar code to calculate new tilt information for the label, and the image is analyzed. If this search fails, models of the label top and bottom are searched for, and new tilt information is calculated. Other model pairs (left top and left bottom) and (right top and right bottom) can also be used.

When a scan line is selected, analyzing the bar code requires finding the widths of the white and the black bars in the scan line and comparing the sequence and bar widths to the bar-code standard. Three different techniques are used to read the bar code. Each successive technique is used only if the previous one has failed and has indicated that enough bars have been found to warrant using the next method. The techniques are ordered so that the fastest method is performed first and the slowest is performed last. The techniques, in the order performed, are

- *Fixed threshold* A value halfway between the minimum and the maximum values found in the center 50% of the bar code is used as a fixed threshold. Using the center of the bar code avoids edge effects and spans enough characters to avoid the effects of special characters. Only every other pixel is scanned to minimize computation time.
- *First derivative* The first derivative of the bar-code image is calculated. Any first-derivative value that is greater than a previously determined noise threshold is accepted as an edge transition.
- *Adaptive threshold* This threshold is determined for each pixel on the basis of its environment. An adaptive threshold can read a bar code that has varying light intensities, while a fixed threshold may fail to read the same bar code. This method proceeds as follows:
  1. Start at the beginning. Use the next n pixels to determine the threshold for each starting pixel. This threshold is halfway between the minimum and the maximum values found in the n pixels. Only every other pixel is scanned to minimize computation time. Using the next n pixels eliminates edge effects as soon as the starting pixel is off the edge. This process continues for 33% of the bar-code length.

2. Start at the end. Use the previous n pixels to determine the threshold for each starting pixel. This threshold is halfway between the minimum and the maximum values found in the n pixels. Only every other pixel is scanned to minimize computation time. Using the previous n pixels eliminates edge effects as soon as the starting pixel is off the end. This process continues for 33% of the bar-code length.
3. Linearly interpolate the remaining thresholds for the middle 33% of the bar code.

The raw gray-scale data in the scan line are thresholded into a series of bar widths and colors (black and white) in sequence. This sequence is used to decipher the bar code. In practice, a camera-based system can have difficulty applying bar-code specifications to decipher bar codes such as the following:

- Narrow bar width $= n$ pixels. The camera may not be able to discern narrow white bars and narrow black bars as being the same width.
- Wide white bar width $= m \times n$ pixels, where lower threshold $< m <$ upper threshold. The camera may not be able to discern wide bars as being a specific multiple of the narrow bar width.

These difficulties can be caused by a low number of pixels per bar in the camera image, differing responses of the camera to black and white transitions (varying saturation of the pixels), aliasing problems, or thresholding variations.

Methods that reject bad characters must be selected without comparing white bar widths with black bar widths to ensure the quality of the label deciphered. This quality is ensured by using a three-pronged approach. Gross-quality filters are used when deciphering a character. Multiple reads are performed until at least two attempts match to ensure that more than one scan line deciphered the same. After a bar code has been read multiple times, a neural network is used to detect bad characters that were not caught by the filters.

- *Gross-quality filters*

These filters are used to screen out bad characters and never directly compare white bar widths with black bar widths:

- *Relative size of wide bars* The wide bars in the bar code must be wider than the narrow bars.
- *Glitch detection* Single-pixel glitches indicate an imaging problem and force rejection of the character being analyzed.
- *Wide bar detection* Any excessively wide bar is failed as being too wide.
- *Quiet-zone relative size test* Both quiet zones surrounding any internal (not the start or stop) character must pass the tests for narrow white bars.

- *Multiple reads*

Successive scan lines are deciphered until two results match. The scan lines start at the horizontal center of the bar code and alternate outward along the horizontal axis. This scan sequence ensures that a minimum number of scans are outside the bar code.

- *Neural network*

After the label is read and verified, the data for the characters in the last label read are tested using a neural network. Thus, the neural network is called only once per label, reducing the time penalty for this computation-intensive task.

A modified back-propagation neural network is used for this analysis. The training set for this network was created by selecting good characters for the good training set and only those bad characters that escaped the filters for the defect training set. The inputs to the neural network were the scaled widths of the bars in the character. These inputs are ordered from the widest black bar to the narrowest black bar, then from the narrowest white bar to the widest white bar. This ordering reduces the number of samples required in the training set because it makes all characters look identical. The neural network output indicates either a good or a bad character. The back-propagation network was modified to run in fixed-decimal-point integer arithmetic and uses a truncation function instead of the sigmoid function.

Various sizes of networks were tested. All had nine inputs (one for each bar) and two outputs (good and bad). These networks had two to eight nodes in a single middle layer. The network that converged well with the minimum number of middle-layer nodes had five nodes in its middle layer.

The training results on this network are shown in **Figure 7**. This network converged with both the maximum error and the RMS error below 0.01 (full scale is 0.00 to 1.00). Having both the maximum error and the RMS error converge means that the network will correctly select the bad characters without accidentally rejecting a good character.

Occasionally, the vision system cannot read a label or reads a different label than expected. When this occurs, the actual digitized image is copied into one of five rolling buffers. Whenever the vision system is idle and at least one image is buffered, a packet is generated by converting the image data to text and then compressing several image scan lines into a data packet that is sent to the Library Manager, where it is stored in one of fifty files. The vision code then checks for a command from the accessor controller. If the vision system is still idle, another set of lines is processed. Because only one packet is processed at a time and only while the vision system is idle, the only delay added to the vision process is the overhead associated with compressing and sending one packet, which takes a few tenths of a second.

The automated vision dump and its associated display tool, which allows actual captured images to be viewed on an IBM PC (such as an office workstation or the Library Manager), was used to diagnose problems (such as camera and cable failures) as the 3495 was being developed. The dumps have also been used to detect and verify the 3490 index and unload failures, either complete or partial (for example, the output stack did not move the full distance), and to help identify new drive error detection and recovery microcode requirements.

During 3495 development, those images that were determined to be vision microcode problems (actual read errors) were saved and sent to the vision microcode engineer, who used them to develop new code to eliminate these problems. Through the analysis of these images, the microcode filters and a neural network were developed to eliminate errors in the final product.

- *Adaptive gain and offset control*

The 3495 is designed to run for months without requiring any maintenance activities. Over this period of time, a lamp can exhibit a noticeable decrease in its intensity.

The vision system is able to compensate for this reduced output by adjusting the gain and the offset used in its analog-to-digital conversion process.

Because the shift in intensity on a lamp occurs over time and is not great enough to affect any single read, a feed-forward technique was chosen. The vision system spends much of its time idle, waiting for commands from the accessor controller. The optimum new settings for offset and gain are calculated by the vision controller after a label is read and reported to the accessor controller. Thus, only idle time is used for this operation, and vision performance is not affected.

A back-propagation neural network was chosen to perform this adaptive offset and gain adjustment. Whenever a bar code is read, the pixels used to read the last scan are saved in a buffer. After the response is sent to the accessor controller, the neural network is run to determine the new optimal gain and offset values.

Summary

The IBM 3495 Tape Library Dataserver successfully integrated IBM and OEM components. This library holds up to 18000 cartridges and is automatically taught. Innovative concepts have been incorporated into its accessor and vision control programs to provide efficient and accurate operation.

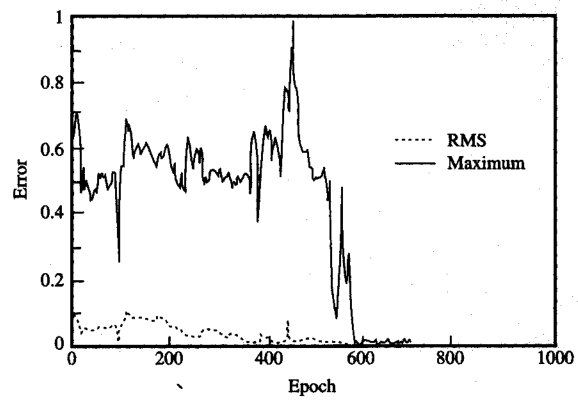

Figure 7

Neural network training error by epoch.

PS/2 is a registered trademark of International Business Machines Corporation.

References

1. B. Nussbaum, "1993 Industrial Design Excellence Awards—Winners: The Best Product Designs of the Year," *Business Week*, June 7, 1993, pp. 58–78.
2. J. Korein and J. Ish-Shalom, "Robotics," *IBM Syst. J.* **26,** 55–95 (1987).
3. W. Xu and K. Wurst, "Relative Calibration Method for Improving Robot Accuracy," *Robotersysteme* (Germany) **8,** No. 2, 107–113 (1992).
4. J. Berg, "Robot Accuracy: A Matter of Programming," *Int. J. Adv. Manuf. Technol.* (*UK*) **7,** No. 4, 193–197 (1992).
5. R. Lenz and R. Tsai, "Calibrating a Cartesian Robot with Eye-on-Hand Configuration Independent of Eye-to-Hand Relationship," *Research Report RC-13211* (*No. 59088*), IBM Thomas J. Watson Research Center, Yorktown Heights, NY, 1987.
6. R. Floyd, "Robotic Languages: A Comparison of Six Offerings," *Technical Report TR-54.0433*, IBM System Products Division, Boca Raton, FL, June 1988.

Received February 28, 1995; accepted for publication March 26, 1996

Gregory T. Kishi *IBM Storage Systems Division, 9000 South Rita Road, Tucson, Arizona 85744 (MOTORMAN at TUCVM2, gkishi@vnet.ibm.com).* Mr. Kishi received a B.S. degree in chemical engineering from the University of California at Davis; he joined IBM in 1978, working as a wet-process manufacturing engineer until 1981. Since then, he has worked primarily on robotic and vision systems in both manufacturing and development engineering. Mr. Kishi received an IBM ADSTAR Market-Driven Quality Award, an IBM Outstanding Innovation Award, and two IBM Outstanding Technical Achievement Awards for his work in this area. As a former member of Robotics International, Society of Manufacturing Engineers, he served on the advisory committee and as a session chair for the 1984 and 1996 Robots West conferences. Mr. Kishi currently works as a microcode development engineer on future tape controllers.