VISION

VISION highlights the latest developments in the machine vision industry, including applications, techniques and methods. Feature articles include updates on the latest manufacturing research, tutorials on a particular manufacturing technology and field reports on installed manufacturing technologies, all from a technical perspective. Each issue also includes additional brief articles, product announcements and event listings.

AUTHOR GUIDE

SUBSCRIBE

3rd Quarter 2006 Volume 22 No. 3

Content Online

THIRD QUARTER 2006 **www.sme.org/ama** VOL. 22, NO. 3

VISION

A Real-Time Hand Gesture System Based on Evolutionary Search

J. Wachs, H. Stern and **Y. Edan,**
Ben-Gurion University of the Negev (Be'er-Sheva, Israel)
M. Gillam, C. Feied, M. Smith and **J. Handler,**
Washington Hospital Center (Washington, D.C.)

Computer information technology is increasingly penetrating into the hospital domain. It is important that such technology be used in a safe manner to avoid serious mistakes leading to possible fatal incidents. Keyboards and mice are today's principle method of human-computer interaction. Unfortunately, it has been found that a common method of spreading infection from one person to another include computer keyboards and mice in intensive care units (ICUs) used by doctors and nurses [1]. Introducing a more natural

human-computer interaction (HCI) will have a positive impact in today's hospital environment. The basis of human-to-human communications is speech and gesture (including facial expression, hand and body gestures and eye gaze). With FAce MOUSe [2], a surgeon can control the motion of the laparoscope by simply making the appropriate face gesture, without hand or foot switches or voice input. In this article the authors explore only the use of hand gestures, which can (in the future) be further enhanced by other modalities.

A vision-based gesture capture system to manipulate windows and objects within a graphical user interface (GUI) is proffered. Current research to incorporate hand gestures into doctor-computer interface have appeared in Graetzel et al. [3]. They developed a computer vision system that enables surgeons to perform standard mouse functions (pointer movement and button presses) with hand gestures. According to Zeng et al. [4], by tracking the position of the fingers, they can gather quantitative data about the breast palpation process for further analysis. Much of the research on real-time gesture recognition has focused on exclusively dynamic or static gestures. In this work, hand motion and posture are considered simultaneously. This allows for much richer and more realistic gesture representations. This system is user independent, without the need of a large multiuser training set. In this experiment, a Fuzzy C-mean (FCM) discriminator is used along with Haar-type features. To obtain a more optimal system design, a neighborhood search method was employed for efficient feature selection and classifier parameter tuning. The real-time operation of the gesture interface was tested in a hospital environment. In this domain, noncontact aspect of the gesture interface avoids the problem of possible transfer of contagious diseases through traditional keyboard/mice user interfaces.

System Overview

A Web camera placed above the screen (*Figure 1a*) captures a sequence of images, such as those shown in *Figure 1b*. The hand is segmented using color, B/W threshold and various morphological image processing operations. The location of the hand in each image is represented by the 2-D coordinates of its centroid and mapped into one of eight possible navigation directions of the screen (see *Figure 2*) to position the cursor of a virtual mouse. The motion of the hand is interpreted by a tracking module. At certain points in the interaction it becomes necessary to classify the pose of the hand. Then the image is cropped tightly around the blob of the hand and a more accurate segmentation is performed. The postures are recognized by extracting symbolic features from the sequence of images. The sequence of features is interpreted by a supervised FCM discriminator that has been trained to discriminate various hand poses. The classification is used to bring up X-ray images, select a patient record from the database or move objects and windows in the screen. A two-layer architecture is used. The lower level provides tracking and recognition functions, while the higher level manages the user interface.

Figure 1. *Gesture capture.*

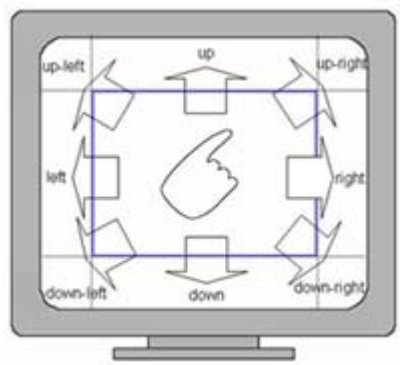


Figure 2. *Screen navigation map.*

Segmentation

To track and recognize gestures, the CAMSHIFT [5] algorithm is used together with an FCM algorithm [6]. For CAMSHIFT, a probability distribution image of the hand color is created using a 2-D hue-saturation color histogram [7]. This histogram is used as a lookup table to convert the acquired camera images into a corresponding skin color through a process known as back projection. Thresholding to black and white followed by morphological operations is used to obtain a single component for further processing to classify the gestures.

The initial 2-D histogram is generated in real time by the user in the "calibration" stage of the system. The interface preview window shows an outline of the palm of the hand gesture drawn on the screen. The user places his hand within the template while the color model histogram is built (*Figure 3*), after which the tracking module (Camshift) is triggered to follow the hand. The calibration process is initiated by the detection of motion of the hand within the region of the template. To avoid false motion clues originated by nonhand motion, a background maintenance operation is performed. A first image of the background is stored right after the application is launched, and then background differencing is used to isolate the moving object (hand) from the background. Because background pixels have small variations due changes in illumination over an extended period of time, the background image is dynamically maintained. Background variations are identified by threshold—the absolute difference between two consecutive frames. If the difference is under some threshold t_1, then the current images contain only a background; otherwise an upper-threshold level t_2 is checked to test whether the present object is a hand. In case the current image is a background, the background stored image is updated using a running smoothed average.

$$Bcc_k(i,j) = (1-\alpha) * Bcc_{k-1}(i,j) + \alpha * f(i,j) \qquad (1)$$

where B_{cc} is the updated stored background image at frame k, B_{cc-1} is the stored background image at frame k–1, α is the smoothing coefficient (regulates update speed) and $f(i,j)$ is the current background image at frame k. Small changes in illumination will only update the background, while large changes in intensity will trigger the tracking module. It is assumed that the

hand is the only skin-colored object moving on the area of the template.

Figure 3. *User's hand-skin color calibration.*

A low threshold and an open and a closed morphology operation followed by the selection of the biggest component are applied to obtain a single connected blob, see *Figure 4*.

Figure 4. *Image pre-processing of the pose.*

Feature Extraction and Pose Recognition

Gesture Vocabulary. There are currently three methods provided for generating mouse button clicks. The first two methods—click and double-click—consist of moving the cursor to the desired position and holding the hand stationary for a short time; performing the gesture similar to *Figures 5a* and *5b* will activate the command click/double-click of the virtual sterile mouse in the current position of the cursor. The third method, "drag" (*Figure 5c*), after being activated as the previous ones, will perform the drag command on the current view, while the hand moves to one of the eight directions. When the hand returns to the "neutral area," the command is terminated.

(a) (b) (c)

Figure. 5. *Gesture vocabulary.*

Hand Tracking and Pose Recognition. The authors classify hand gestures using a simple finite-state machine (*Figure 6*). When the doctor wishes to move the cursor over the screen, he moves his hand out of the "neutral area" to any of the eight directions.

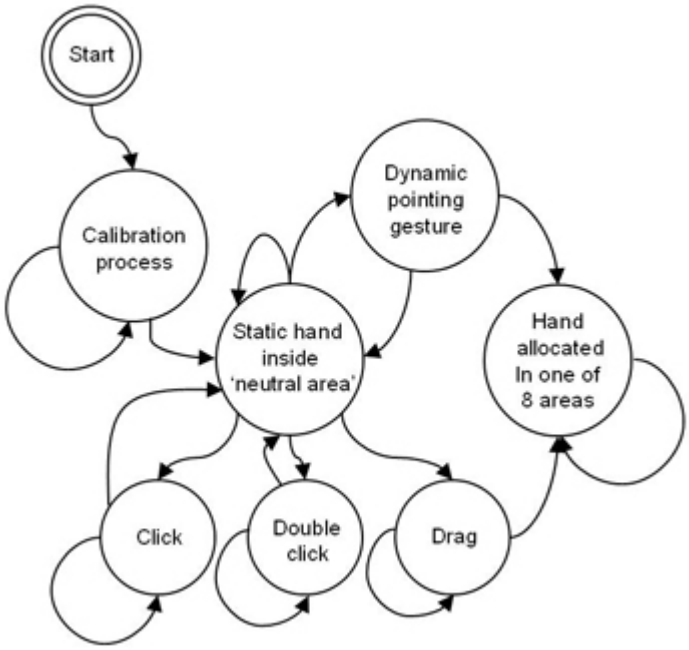


Figure 6. *State machine for the gesture-based medical browser.*

The interaction is designed in this way because the doctor will often have his hands in the "neutral area," without intending to control the cursor. While the hand is in one of the eight quadrants, the cursor moves in requested direction (*Figure 7*).

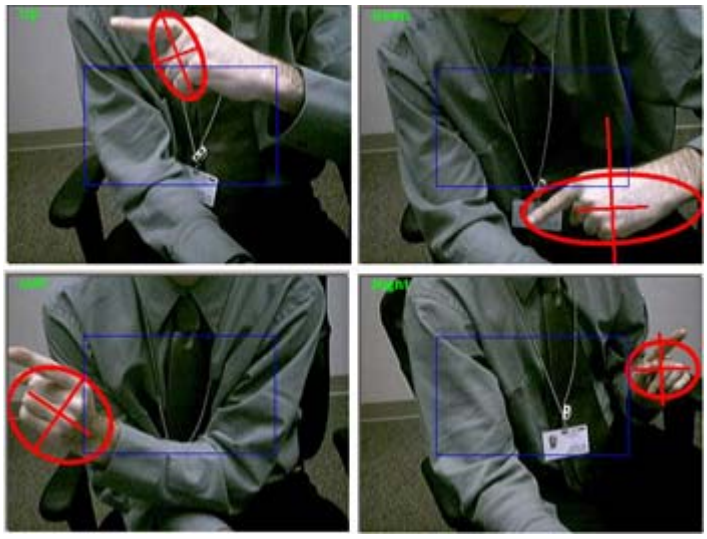


Figure 7. *Four quadrants mapped to cursor movement.*

To facilitate positioning, the hand motion is mapped to cursor movement. Small, slow hand (large fast) motion cause small (large) pointer position changes. In this manner, the user can precisely control pointer alignment. When a doctor decides to perform a click, double-click or drag with the virtual mouse, he/she places the hand in the "neutral area" momentarily. This method differentiates between navigation and precise commands.

Haar Features. Basically, the features of this detector are weighted differences of integrals over rectangular subregions. *Figures 8a*, *8b*, *8c* and *8d* visualizes the set of available feature types, where black and white rectangles correspond to positive and negative weights, respectively. The feature types consist of four different edge-line features. The learning algorithm automatically selects the most discriminate features, considering all possible feature types, sizes and locations. The feature types are reminiscent of Haar wavelets and early features of the human visual pathway, such as center surround and directional responses. Their main advantage is that they can be computed in constant time at any scale.

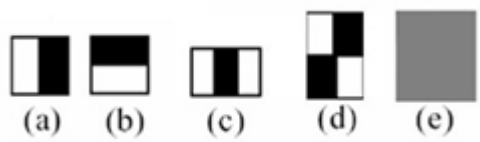


Figure 8. *Extended integral rectangle feature set.*

Each rectangle feature is computed by summing up pixels within smaller rectangles:

$$f_i = \sum_{i \in I=\{1,\ldots,N\}} \omega_i * \mathrm{Re}cSum(r_i) \qquad (2)$$

with weights $\omega_i \in \Re$, rectangles r_i and their number N. Only weighted combinations of pixel sums of two rectangles are considered. The weights have opposite signs (indicated as black and white in *Figure 8*) and are used to compensate between differences in area. Efficient computation is achieved by using summed area tables. Average block features have been added to the original feature set from Viola and Jones [8], for example, features f_1, f_2, f_3 and f_4 (*Figures 8a*, *8b*, *8c* and *8d*, respectively). The augmented rectangle feature f_5 (*Figure 8e*) has been shown to extend the expressiveness and versatility of the original features, leading to more accurate classification. Given that the basic resolution of the classifier is 100 x 100, the exhaustive set of rectangle features is quite large (>750,000). Even though computing each feature is efficient, the complete set is prohibitively expensive [8]. A rectangle *r* in the image can be defined by the (*x*,*y*) position of its upper-left corner and by its width *w* and its height *h*. The total set of rectangles is constrained in an image by using the relation: $x = w*n$ and $y = h*m$ where *n*, *m* are integer numbers. Hence, the total number of possible rectangles is <13,334. In general, the total number of possible rectangles that fix in a square image with size *L* is $\sum(L/n)^2$ for $n = 1$ to L.

Pose Recognition. In this system, the Haar rectangular positions are reduced severely to a set of "selected" rectangles *v*. These rectangles are limited to lie within a bounding box of the hand-tracking window and are obtained by dividing the window in *m* rows and *n* columns, and for each cell decide whether it is selected "1" or not "0." A more elaborated strategy enables one to define the type of feature for selected rectangles. Therefore, a set of rectangles in a window is defined by a tuple $\{n,m,t\}$, where n,m are columns and rows and $t = \{t_1,\ldots,t_i,\ldots,t_v\}$ represent the type of feature of rectangle *i* indexed row wise from left to right. The feature type *t* can take integer values from 0 to 5, where 0 means that the rectangle is not selected, and 1,2,3,4,5 represent features of type f_1, f_2, f_3, f_4 and f_5, respectively. The hypothesis expressed in Ref. [8] is that a very small number of these features can be combined to form an effective classifier. As opposed to Ref. [8], this learning algorithm is not designed to select a single rectangle feature that best separates the positive and negative for each stage of a cascade of classifiers. Instead, a set of rectangle features is evaluated simultaneously, which accelerates the process of feature selection. The Haar features selected are input into the hand-gesture FCM recognition system architecture. Note that the feature sizes are automatically adjusted to fit into the dynamically changing bounding box created by the tracking system.

Optimal Feature Selection Based on Evolutionary Search. The process of feature selection and finding the parameters of the Fuzzy C-mean algorithm for classifying hand gesture sets uses a probabilistic neighborhood search (or evolutionary search) (PNS) method [9]. The PNS selects samples in a small neighborhood around the current solution based on a special mixture type point distribution model:

$$\mathrm{P}_S(\mathrm{x}\,|\,\mathrm{h}) = \begin{cases} \mathrm{h}, & x = 0 \\ h((1-h)^{|x|})/2, & x = \pm 1, \pm 2, ,, , \pm(S-1) \\ ((1-h)^{|x|})/2, & x = \pm S \end{cases} \qquad (3)$$

where

S = maximum number of step increments
h = probability of no change
x_j = a random variable representing the signed (positive or negative coordinate direction) number of step size changes for parameter p_j.
$P_S(x|h) = P_r(x = s)$ the probability of step size s, given h.

Figure 9 shows an example of the convergence behavior of the PNS algorithm for five randomly generated starting solutions.

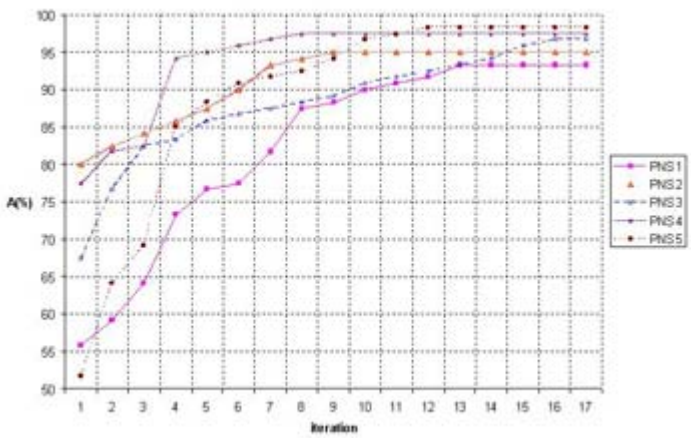


Figure 9. *Convergence curve for five solutions of PNS alg.*

Figure 10 shows the optimal set of features selected by this run. The feature f_4 and f_5 capture whether the posture is a kind of palm-based gesture using diagonal line features and average grayscale. Inner-hand regions (such as inside of palms) and normal size fingers are detected through f_1, while f_3 captures the ring finger based on edge properties. Hence, this is quite different from traditional gesture classifiers that rely on parametric models or statistical properties of the gesture.

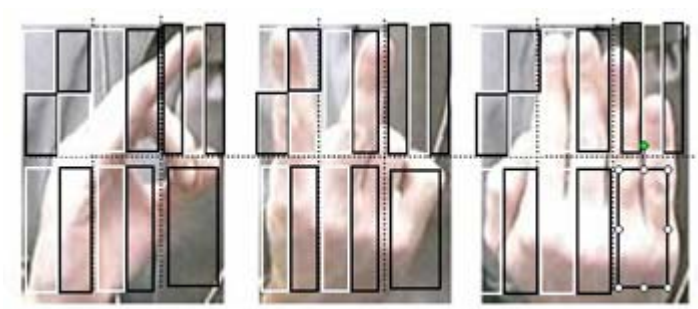


Figure 10. *Automatically selected features (f_4, f_1, f_3, f_1, f_1, f_5) for the 2 x 3 partition found by PNS.*

Note that the result is a set of common features for all three pose gestures. The optimal partition of the bounding box was 2 x 3, giving six feature rectangles. The parameter search routine found both the number of sub blocks and the type of Haar feature to assign to each.

Test of Hand Gesture Fuzzy C-Mean Classifier

To evaluate the overall performance of the hand gesture tracking and Fuzzy C-mean recognition system, the Azyxxi Real-time Repository™ was used, which was designed to accommodate multidata types. *Figure 11* shows an example of the user screen of this database.

Figure 11. *Screen shot of Azyxxi controlled by gestures.*

The data set consists of 20 trials of each of four tasks: select record of patient, browse X-ray collection, select specific X-ray and zoom-in damaged area. The user was asked to perform the tasks sequentially. The total results for one experienced user are shown in *Table 1*. The success task rate shows how many times an action (part of the task) was performed correctly without catastrophic errors. Minor errors are related to inaccurate position of the cursor due to fast movements or changes in direction, while catastrophic errors occurred as a result of misclassification of the supervised FCM algorithm.

Task	Steps	Trials	Success Task
Select record of patient	1	19	94.74%
Browse X-ray collection	2	20	100%
Select specific X-ray	1	20	100%
Zoom in damaged area	2	19	94.74%

Table 1. *Results of medical tasks using hand gestures.*

In general, the results of *Table 1* indicate both the ability of the system to successfully track dynamic postures and classify them with a high level of accuracy.

Test of the AdaBoost-Haar Classifier

Most of the work using Haar features is only for detection of a single object in an image and not as features embedded in a multiobject classification system, as was done here. Nevertheless, because of the recent popularity of using Haar features for object detection, it was decided to run a small exploratory test, using only one of our pose gestures, to assess its suitability for multihand gesture classification.

For this test, cascade of boosted classifiers was used based on an extended set of Haar-like features [10]. Only one gesture was adopted (*Figure 5a*) to asses the performance of this method. Positive samples were created using an automated process, which overlaid one instance of the gesture over several random backgrounds. The original instance was embedded in the background, twisted from –30 to 30°, in the *x*, *y* and *z*-axis a total of 1,350

positives samples. A set of 5,318 negative samples images were obtained from Sebastian Marcel's repository Web site [11]. The testing included 50 image sets, and it was created automatically using the same methodology (*Figure 12*).

Figure 12. *Positive testing set for different angles.*

Using a cascade of 13 stages of classifiers, the primary results show an average hit rate of 52%, 48% missed and 34 false alarms. These results are for testing, not training. The explanation for the low accuracy is due to the fact that the training and testing of positive samples were artificially generated. The classifiers could not learn light and geometry changes properly because the single instance used did not create enough variations. Lighting conditions in a rotated image are different than a hand in real conditions. Gestures that include shadows, occlusion and change in geometry must be obtained by true-life images. However, for the task of gesture detection, the rectangle features selected by AdaBoost are meaningful and easily interpreted. The features selected seem to focus on one area, for example, the region of the palms is often darker than the far side of the hand (*Figure 13*).

Figure 13. *Four features selected by AdaBoost in the last stage. Features are shown overlayed on a typical training gesture. The features measures the difference in intensity between the region of the palm and the far side face of the hand.*

In the hand-gesture FCM classifier, the tracking system uses color segmentation, which adaptively compensates for variable illumination, so there is no need to include varying illumination in the training samples. In the Fuzzy C-mean classifier, good results were obtained by using only 40 training samples per gesture. Because of the large training set required for the Haar Detector approach, it does not look promising. However, further testing is required.

Conclusion

In this article, the authors consider a vision-based system that can interpret a user's gestures in real time to manipulate windows and objects within a medical data visualization environment. A hand segmentation procedure first extracts binary hand blobs from each frame of an acquired image sequence. Dynamic navigation gestures are translated to commands based on their relative positions on the screen. Static gesture poses are identified to execute

nondirectional commands. This is accomplished by using Haar-like features to represent the shape of the hand. These features are then input to a FCM clustering algorithm for pose classification. A probabilistic neighborhood search algorithm is employed to automatically select a small number of visual features and to tune a FCM classification algorithm. Intelligent handling of features allows nondiscriminating regions of the image to be quickly discarded while spending more computation on promising discriminating regions. The gesture recognition system was implemented in a sterile medical data-browser environment. Test results on four interface tasks showed that the use of these simple features with the supervised Fuzzy C-mean classifier yielded successful performance rates of 95% to 100%. In addition, a small exploratory test of the AdaBoost Haar detector was made to detect a single hand gesture from the pose set. The results indicated that the required large training set did not warrant the use of this method for the authors' purposes. It should be noted that the hand-gesture Fuzzy C-means recognition system used here must find a bounding box around the gesture using color tracking and image processing, and then use the Haar features within the subboxes of a partition of the bounding box. The AdaBoost Haar system does not use a bounding box but searches over the entire image and then classifies the image as containing a hand or not. Although the initial test points out a number of difficulties in using the AdaBoost Haar detector methodology for hand gesture recognition, further tests are warranted.

Acknowledgment

This project was partially supported by the Paul Ivanier Center for Robotics Research & Production Management, Ben Gurion University of the Negev.

References

[1] Schultz, M.; Gill, J.; Zubairi, S.; Huber, R.; and Gordin, F. **"Bacterial Contamination of Computer Keyboards in a Teaching Hospital."** *Infection Control and Hospital Epidemiology* (v4, n24, 2003), pp302-303.

[2] Nishikawa, A.; Hosoi, T.; Koara, K.; Negoro, D.; Hikita, A.; Asano, S.; Kakutani, H.; Miyazaki, F.; Sekimoto, M.; Yasui, M.; Miyake, Y.; Takiguchi, S.; and Monden, M. "FAce MOUSe: A Novel Human-Machine Interface for Controlling the Position of a Laparoscope." *IEEE Trans. on Robotics and Automation* (v19, n5, 2003), pp825-841.

[3] Graetzel, C.; Fong, T.W.; Grange, S.; and Baur, C. "A Non-Contact Mouse for Surgeon-Computer Interaction." *Technology and Health Care* (v12, n3, 2004), pp245-257.

[4] Zeng, T J.; Wang, Y.; Freedman, M.T.; and Mun, S.K. "Finger Tracking for Breast Palpation Quantification Using Color Image Feature." *Optical Engg.* (v36, n12, Dec. 1997), pp3455-3461.

[5] Bradski, G.R. "Computer Vision Face Tracking for Use in a Perceptual User Interface." *Intel Technical Journal*, 1998, pp1-15.

[6] Wachs, J.P.; Stern, H.; and Edan, Y. "Cluster Labeling and Parameter Estimation for Automated Set Up of a Hand Gesture Recognition System." *IEEE Trans. in Man, Systems and Cybernetics, Part A* (v35, n6, Nov. 2005), pp932-944.

[7] Foley, J.D.; van Dam, A.; Feiner, S.K.; and Hughes, J.F. "Computer Graphics: Principles and Practice." 2nd ed., Boston: Addison Wesley, 1987.

[8] Viola, P. and Jones, M. "Rapid Object Detection Using a Boosted Cascade of Simple Features." *Proc.* of IEEE Conf. on Computer Vision and Pattern Recognition, Kauai, Hawaii, Dec. 2001.

[9] Stern, H.; Wachs, J.P.; and Edan, Y. "Parameter Calibration for Reconfiguration of a Hand Gesture Tele-Robotic Control System." *Proc.* of JUSFA 2004 Japan – USA Symp. on Flexible Automation, Denver,

July 19-21, 2004.

[10] Lienhart, R. and Maydt, J. "An Extended Set of Haar-Like Features for Rapid Object Detection." IEEE ICIP 2002 (v1, Sept. 2002), pp900-903.

[11] **"InteractPlay Database**." Martigny, Switzerland: IDIAP Research Institute, 2004.

VISION

VISION highlights the latest developments in the machine vision industry, including applications, techniques and methods. Feature articles include updates on the latest manufacturing research, tutorials on a particular manufacturing technology and field reports on installed manufacturing technologies, all from a technical perspective. Each issue also includes additional brief articles, product announcements and event listings.

AUTHOR GUIDE

SUBSCRIBE

3rd Quarter 2006 Volume 22 No. 3

Content Online

THIRD QUARTER 2006 **www.sme.org/ama** VOL. 22, NO. 3

VISION

NEWSLINE

NIST Gears Up to Verify Short-Range 3-D Imaging

Three-dimensional imaging devices are becoming important measuring tools in manufacturing, construction and transportation. Numerous models, capable of digitally capturing the existing conditions of objects from as small as pipe fittings to as large as an entire bridge, are on the market. A lack of standard tests to verify manufacturers' performance specifications is inhibiting wider market acceptance of these devices. Researchers at the **National Institute of Standards and Technology** (Gaithersburg, Md.) recently established an indoor artifact-based facility to create new test protocols and performance measures to evaluate such 3-D imaging systems. Several prototype artifacts (spheres, a stairway and a slotted disc) are currently being tested for evaluating both instruments and software. The new facility is part of a larger

effort to provide standard test protocols and associated facilities for evaluating and calibrating these instruments. NIST also operates an indoor 60 m range calibration facility and is developing a separate 3-D facility so that manufacturers or research groups can send in instruments for spatial calibration. Finally, NIST will establish an outdoor ranging facility for evaluating the performance of 3-D imaging systems up to 150 to 200 m. A consensus-based standards development process began this summer. Protocol for evaluating the range performance of imaging devices and a draft list of commonly used terminology, developed during a series of workshops at NIST, will be submitted to ASTM International. These standards will provide objective, repeatable comparisons of different 3-D imaging devices, reduce confusion about terminology and increase user confidence in the systems. NIST Techbeat, April 2006.

Smart Sensor Brings Color to Vision Sensing

ZFV-C smart vision sensor offers sensing capabilities close to human vision, including the ability to distinguish shapes and color. Color sensing enables the sensor to see images invisible to monochromatic sensors. Integrated color filters enable sensing of only selected colors when necessary to get a better image. The sensor features a “teach-and-go” function using a color LCD screen and menu that reduce setup for inspections. The screen shows a live image for instant feedback during both setup and inspection operations. Eight sensor functions are available, depending on the controller user, including counting, pattern recognition, size verification, position, brightness, width, text verification and hue (comparing sensed color to a reference color). As many as five sensors can be connected to one controller bus through a snap-on system, allowing five different inspections in one pass. Contact **Omron Electronics** (Schaumburg, Ill.) to learn more.

Machine Vision Seals Beer-Barrel Integrity

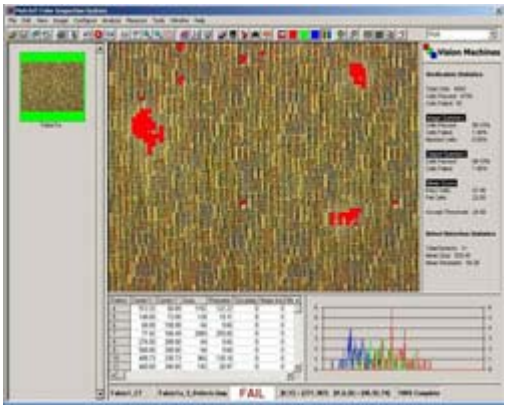
At the St. James Gate brewing plant of Guinness, the company’s draught beer is brewed and kegs are filled for domestic, European and international customers. To fill, check and ship these kegs at a rate of approximately 1,000 kegs/hr, Guinness has fully automated the plant. Recycled kegs are placed on an automated conveyor system, where they go through several operations before they can be reshipped to the company’s distributors. After the kegs are washed, rinsed and sterilized, they are filled with beer through a one-way valve at the top of the keg, known as a spear. To determine whether each keg is properly filled, the kegs are weighed and the seals checked for any leaks before they are finally packaged and shipped from the brewery. To automate the process of inspecting the kegs for leakage, Guinness called on Machine Vision Direction. After being filled and weighed, any improperly filled kegs are rejected from the conveyor, automatically inverted and emptied. While the beer is recycled, the kegs are returned to a station for refilling and the process repeated. Properly filled kegs travel along the conveyor and into an IP65-certified stainless steel enclosure that houses the machine vision system. As kegs move into the inspection station, their presence is detected by an EQ-500 adjustable-range reflective photoelectric sensor from PanaSonic Electric Works. To capture images of the top of the kegs, MVD uses a 0.3333 in. (0.85 cm) progressive-scan, 30 frames/sec. 1,024 x 76 XC-HR-70 monochrome camera from Sony. Fitted with a 25 mm C-mount lens from Pentax Imaging, the camera is mounted on a Robo Cylinder from Intelligent Actuator. After each keg is inspected, the pass/fail decision made by the Matrox 4SightII vision system is used to control the Panasonic PLC, which actuates a shunt mechanism to send failed kegs to a decanting station.

These kegs are emptied, the beer recycled and the keg sent for repair. To control the number of false rejects, the system's operator can program the level of acceptability of different keg types using the system's human-machine interface (HMI) that is displayed on the flat panel display. Written in Visual C++, the HMI runs on top of MIL to allow the user to set up each production run and monitor the hourly failure rate. Collected data include the date, time and quantity of kegs that have failed. **Vision Systems Design**, July 2006.

New, Lower-Cost Vision-Guided Motion Controller

Precise Automation's (Los Altos, Calif.) new Guidance 2400 series is the latest addition in a series of motion controllers based on a distributed network architecture that allows the controller to be located at the point of use. It is lower-cost, lower-voltage (24-90 VDC) version of Precise Automation's line of Guidance 3400 controllers. The Guidance 2400 contains four motor driver, eight encoder input channels, a 700 MIPS processor, digital and analog inputs and outputs, Ethernet and RS-232 interfaces. It can provide up to 2,000 W of motor power in an extremely small package—225 mm long by 138 mm wide by 61 mm high. It is a small enough to be placed inside machines, eliminating the need for a separate control cabinet. The user interface is based on a Web server that resides in the Guidance 2400, allowing the controller to be accessed from anywhere in the world for easy development, operation and maintenance. The software includes a complete set of motion commands, machine kinematics, a continuous path motion planner and trajectory generator, a powerful language with motion commands added to the Basic syntax, Active X and .Net links to Microsoft software and an optional machine vision package that can execute in a networked PC. The vision software contains measurement tools and a patented object locator that can locate multiple instances of parts in any orientation, even with cluttered backgrounds.

Software Facilitates Color and Texture-Based Inspection



The **Value Engineering Alliance's** (Cambridge, Mass.) HueView, a new innovative inspection technology specifically developed to deliver comprehensive color analysis of multicolored, patterned and textured items. Used to measure color (in RGD, HIS or LAB space) and texture, monitor product appearance, classify/grade goods, facilitate sorting operations, assess visual similarity or detect defects, by comparing images of items or scenes of interest to their respective stored reference images, HueView optimizes its flexibility, accuracy and repeatability by operating at high spatial resolutions and enabling inspections to be done based on color, texture or a user-selectable combination of the two. Allowing texture analysis to be factored into inspection as required ensures that items possessing sufficiently similar color content but featuring varying mixes of multiple colors or subtle defects in their patterns can still be identified as aberrant or of unsatisfactory appearance. HueView is currently available in the form of an easy-to-use WinTel-compatible software application that supports image acquisition using FireWire cameras, TWAIN and analog camera/frame grabber combinations, eliminates the need for programming, is quickly trained by simply being shown reference examples, incorporates powerful pattern recognition and learning capabilities and requires no follow-up data analysis. The level of interest expressed by software developers will determine whether the technology becomes available in a form suitable for implementation by

those ready, willing and able to create their own application programs.

Multi-Camera Vision System

The Keyence CV-3000 vision system is a new platform design that provides multiple color and monochrome camera connectivity using up to four cameras from eight different models. Triple processors provide processing times up to 10X faster than conventional systems, even when using two Mega-pixel color cameras. Flexible memory and new software tools handle applications that previously required application-specific vision systems. Built upon an advanced color engine, with 16.7 million “Auto-Tech” color filters, the CV-3000 provides precise color extraction, high contrast and easy-to-process images. Keyence has expanded on its proven multicamera connection technology to allow simultaneous use of up to four cameras on one controller. Any combination of the eight cameras available for the CV-3000 series can be used together. The connected cameras provide simultaneous imaging and processing. In addition, the system can flexibly cope with future additions and modifications to the inspection specifications. In addition to a lightning-fast RISC CPU chip, the CV-3000 series controller uses two DSP chips to achieve image processing speeds twice that of conventional models. The cameras have also been geared toward high-speed production. Contact **Keyence Corporation of America** (Woodcliff Lake, N.J.) to learn more.

VISION

VISION highlights the latest developments in the machine vision industry, including applications, techniques and methods. Feature articles include updates on the latest manufacturing research, tutorials on a particular manufacturing technology and field reports on installed manufacturing technologies, all from a technical perspective. Each issue also includes additional brief articles, product announcements and event listings.

AUTHOR GUIDE

SUBSCRIBE

3rd Quarter 2006 Volume 22 No. 3

Content Online

THIRD QUARTER 2006 **www.sme.org/ama** VOL. 22, NO. 3

VISION

CALENDAR OF EVENTS

British Machine Vision Conference 2006
September 4-7, 2006 (Edinburgh, Scotland)
British Machine Vision Association

International Conference on Image Analysis and Recognition
September 18-20, 2006 (Póvoa de Varzim, Portugal)
Universidade do Porto, Portugal/University of Waterloo, Canada

Advanced Concepts for Intelligent Vision Systems
September 18-21, 2006 (Antwerp, Belgium)
University of Antwerp

International Conference on Image Processing
October 8-11, 2006 (Atlanta)
IEEE Signal Processing Society

Workshop on Distributed Smart Cameras (DSC 2006)
October 31, 2006 (Boulder, Colo.)
Institute for Technical Informatics

Twenty-First International Conference Image and Vision Computing New Zealand
November 27-29, 2006 (Great Barrier Island, New Zealand)
University of Auckland

9th International Conference on Control, Automation, Robotics and Vision
December 5-8, 2006 (Singapore)
Nanyang Technological University

Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP) 2006
December 13-16, 2006 (Madurai, India)
Indian Institute of Technology, Delhi