Practical computer vision: Example techniques and challenges

S. Pankanti
L. Brown
J. Connell
A. Datta
Q. Fan
R. S. Feris
N. Haas
Y. Li
N. Ratha
H. Trinh

Humans, as well as many living organisms, are gifted with the power of "seeing" and "understanding" the environment around them using their eyes. The ease with which humans process and understand the visual world is very deceiving and often prompts us to underestimate the effort and methods needed to build practical, effective, and inexpensive computer vision systems. In essence, humans have a 500-million-year head start due to evolution; it is extremely difficult at this point to build a computer vision system that has the abilities of a three-year-old child. However, by confining ourselves to particular domains, we can often find shortcuts to solve particular problems. This paper illustrates a number of such solutions in various areas developed by our group at IBM. These include object finding for video surveillance, person identification via biometrics, inspection of manufactured items along railways, and scene understanding for driver assistance, as well as object recognition and motion interpretation for retail stores. We discuss the real-world constraints for each system and describe how we overcame the irksome variability inherent in each task. By further analyzing such successful systems and comparing them to each other, we can come to understand the common underlying problems and thus start to extend our initially limited areas of competence into a more general-purpose vision toolkit. This paper concludes with a set of challenging unresolved problems that if solved could spur great progress in practical computer vision.

Introduction

There has been spectacular growth in the computing industry and the digital camera industry over the last decade, which is starting to allow computer vision to become a part of our daily lives. With the advent of the high-definition television and the inexpensive complementary metal-oxide semiconductor (CMOS) sensors, even recent mobile phones have more than 5-megapixel acquisition capability. In terms of cost per bit of information, cameras far surpass nearly every other sensor; the trick is to make sense of these bits. This achievement has been aided by several other technologies, i.e., powerful computing, high-density storage, and high-bandwidth communications, all at affordable costs. In addition, machine learning has made inroads in many applications; therefore, having a large corpus of machine-readable data available on the web is a significant benefit.

Computer vision can be used for several different kinds of tasks. One long-standing application is in inspection and metrology. For instance, on an assembly line, vision can be used to ensure all bottles have their caps properly secured or to verify that there are no cracks in the glass. Similarly, quantitative visual measurements can check that steel beams are being produced with the correct thickness or can be used to develop 3-D terrain maps from stereo imagery (e.g., photogrammetry).

Another primary use of computer vision is for object recognition. This can be in the context of a very specific (often manufactured) object or the more difficult case of determining the generic class of some object (e.g., a flower). The problem is often further broken into the "what" and "where" questions. Vision systems try to either spot some

Digital Object Identifier: 10.1147/JRD.2011.2165676

0018-8646/11/$5.00 © 2011 IBM

object in a scene or identify an already delimited item. It is typically very difficult to do a complete parsing of the scene simultaneously into all objects and their identities.

Yet, another use of computer vision is estimating the pose (position and orientation) of an object and evaluating its spatial trajectory over time. This has applications in process control, e.g., plucking a part out of a bin or properly reorienting a unit for further assembly operations. Tracking is a crucial part of many unmanned aerial vehicle (drone aircraft) tasks but can also be used in commercial applications such as monitoring whether customers stop and look at the displays at the ends of grocery store aisles.

In addressing these applications, there are a few problems that arise repeatedly. First, before something can be characterized or measured, it must be separated from all the background clutter. Second, many visual properties, such as color and edges, are significantly affected by the ambient illumination conditions. Controlling or compensating for this is often necessary in order to obtain repeatable features. Finally, objects in a 2-D image can look significantly different depending on where the camera is. For instance, a direct overhead view of a person (e.g., a bald head and shoulders) can be quite confusing at first.

As will be seen in the sample projects that follow, these challenges can be handled in a number of ways. One is by devising special invariant descriptors. Another choice is to attempt to normalize the image to some canonical illuminant or viewpoint. Alternatively, the system can simply be engineered with careful placement of the camera and control of the background and by supplying its own light. In many industrial cases, this is easier, less expensive, and more reliable than trying to handle the problems computationally.

Much of our group's particular approach to computer vision is motivated by such commercial concerns. It is generally important that the deployed system be as inexpensive as possible, tending to rule out special-purpose sensors. It is also important that the system provide answers in real time, i.e., a constraint (in conjunction with cost) that favors simpler algorithms as opposed to highly iterative mathematical optimization approaches. Finally, customers want the system to work at least as well as a human and typically in all sorts of adverse imaging conditions. We do not have the luxury of tailoring the problem or throwing out some samples in order to remain in a comfortable part of the problem space.

The performance of computer vision systems is itself something that is problematic to measure. Computer vision is generally a form of pattern recognition and, as such, never presents "sure" answers. There is almost always a tradeoff between, e.g., false alarms and missed detections. Most vision systems have adjustable parameters that allow the user to choose an operating point within the specific distribution of inevitable errors. Yet, determining what constitutes acceptable overall performance is somewhat subjective. There are many possible metrics such as the equal error rate, the F measure, and the area under the receiver operating characteristic curve. However, in practice, we have found that there is a hard limit on the false-alarm rate for many applications. Thus, the most useful single number reflecting the deployed accuracy of the system is its corresponding detection rate at this upper limit.

In order to measure the performance of a vision system, it is necessary to collect quite a lot of data. Typically, this data must also be annotated (by hand) with the correct decision or features. Even more data is needed for a learning system to ensure separate training and test sets. Assembling a sufficient quantity of data is surprisingly time-consuming and expensive (not to mention tedious). One also has to be careful that the statistics of the database match or at least cover the actual statistics observed in the field. This means that, if it rains 20% of the time, then there should be a sufficient number of rainy images in the database. When debugging the system, it is also useful to have a "gap analysis" tool that will automatically link back to this database to show the developer under which conditions errors are occurring. We have found building such tools essential.

The following sections of this paper describe several projects in our group that illustrate different uses of computer vision and show how we have addressed the recurrent subproblems. In particular, we provide an overview of the advances that computer vision has made in video surveillance, biometrics, retail monitoring, rail safety, and driver assistance, as demonstrated by the customer solutions that our group has deployed over the past decade. The material in this paper has been described more completely in previous various conference and workshop papers; readers interested in the details should consult the relevant citations.

Video surveillance

The remarkable growth of sensor data acquisition has led to a situation where there is a shortage of personnel to monitor all of the data that is generated. Such a scenario is typical in a video surveillance situation, where a massive number of cameras are deployed to monitor large geographical areas such as cities. A typical municipal command and control center will have camera monitors covering an entire wall and a bevy of humans monitoring all the incoming video feeds for suspicious activities. Such human monitoring not only suffers from loss of attentiveness, since one cannot simultaneously focus on all the activities in all the cameras at once, but also from human fatigue and boredom while looking at these camera feeds for extended periods of time.

The IBM Smart Vision Suite (SVS) has been deployed in many cities around the world and has even helped the city of Chicago to solve a high-profile case [1]. SVS not only offers real-time alerting capabilities but also enables the user to search for events of interest after the fact. The number of pole-mounted street cameras in a large metro city is

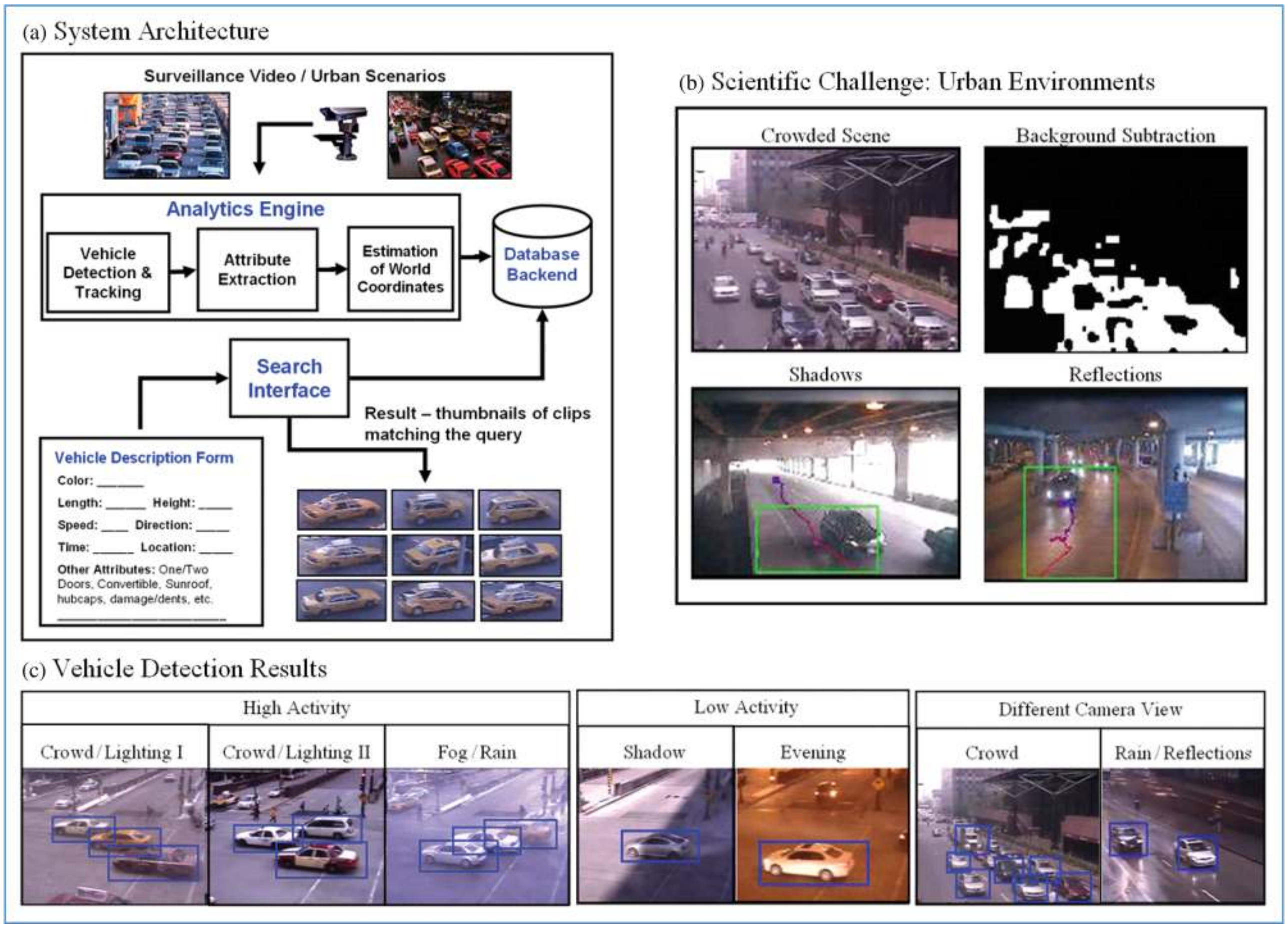
(a) System Architecture
Surveillance Video / Urban Scenarios
Analytics Engine
Vehicle Detection & Tracking
Attribute Extraction
Estimation of World Coordinates
Database Backend
Search Interface
Result – thumbnails of clips matching the query
Vehicle Description Form
Color: ______
Length: _____ Height: _____
Speed: ____ Direction: _____
Time: _____ Location: _____
Other Attributes: One/Two Doors, Convertible, Sunroof, hubcaps, damage/dents, etc.
(b) Scientific Challenge: Urban Environments
Crowded Scene
Background Subtraction
Shadows
Reflections
(c) Vehicle Detection Results
High Activity
Crowd/Lighting I
Crowd/Lighting II
Fog/Rain
Low Activity
Shadow
Evening
Different Camera View
Crowd
Rain/Reflections

Figure 1

(a) High-level system architecture for vehicle search based on fine-grained attributes. (b) Traditional methods based on background modeling fail to segment vehicles in common urban surveillance conditions. In crowded scenes, groups of vehicles are often clustered into the same blob obtained by background subtraction. Shadow effects and reflections also cause issues. Note that the vehicle bounding boxes are not precise, posing problems for subsequent modules such as vehicle color estimation. (c) Examples of vehicle detection results in challenging urban environments, involving crowded scenes, environmental factors, and different camera viewpoints using our novel motion-let approach (©2011, ACM and IEEE).

typically around 1,000–5,000. The number of events (vehicle traffic only) per day per camera on average is 50,000. Therefore, the number of events generated by a surveillance system is 50–250 million per day and 1.5–7.5 billion per month. The IBM SVS is one of the few surveillance systems that support indexing of such a large amount of data. It does this using a combination of web application server clustering and database partitioning.

Figure 1(a) shows the high-level architecture of our system. The analytics engine processes frames from the video cameras. Specifically, it detects and tracks people and vehicles, and then indexes and stores this data in a relational database. It also provides Structured Query Language-based event-retrieval mechanisms that let it respond to requests such as "show me all the people who entered this facility from time *X* to time *Y*" or "show me all the red cars that crossed this avenue last month." This second query illustrates an important and unique feature of IBM SVS: the ability to search for suspicious vehicles based on their fine-grained semantic attributes. Previous solutions have generally relied on license-plate recognition or vehicle classification, which may not be effective for low-resolution cameras or when the plate number is not known. Our current implementation [2] allows the user to search automatically for vehicles based on color, size, length, width, height, speed, direction of travel, date/time, and location, but many more attributes could be considered, including measurements from nonvisual sensors.

Most commercial surveillance systems rely on background modeling for detection of moving objects, particularly vehicles. However, they fail to handle crowded scenes well, since multiple objects close to one another are often merged into a single motion blob. Environmental factors such as shadow effects, rain, and snow also cause issues for object segmentation [see **Figure 1(b)**]. To support the vehicle-search feature, we instead have implemented a novel multiview detection system that relies on a set of "motion-let" classifiers. This consists of a bank of trained detectors using vehicle samples clustered in various parts of the motion configuration space [3]. We learn each detector by using massively parallel feature selection of local image patch descriptors. In addition, we can detect multiple types of vehicles, such as buses, trucks, sport utility vehicles, and compact cars, by training the motion-let detectors in a shape-free appearance space, where all training images are resized to the same aspect ratio. At test time, the system automatically adjusts the aspect ratio of the sliding window as appropriate for the various vehicle types.

Figure 1(c) shows examples of vehicle detection in challenging urban environments involving crowded scenes, environmental factors, and different camera viewpoints. Our detection system runs at 66 frames/s on a 2.4-GHz processor with 3 GB of random access memory. This is one example of robust real-world object recognition despite uncontrolled and often adverse imaging conditions. It can handle a broad range of camera view angles, ambient lighting, and weather conditions, as well as wide variations in vehicle structure. It essentially does segmentation (e.g., for later color and size determination) by direct recognition.

We have described our algorithms for attribute-based vehicle search in urban surveillance environments as part of the IBM Smart Surveillance System. Many other commercial systems for intelligent urban surveillance exist in the market. Examples include the systems of companies such as Siemens [4], ObjectVideo [5], and Honeywell [6], to mention just a few. Video surveillance has also been an active research topic in the academic community, e.g., at University of Central Florida [7], University of Southern California [8], and Ohio State University [9]. One of the key advantages of the IBM system over previous work is its ability to handle challenging urban environments such as crowded scenes and environmental factors. Direct model-based recognition of vehicles (and people), i.e., as opposed to tracking all moving objects, is one step in this direction.

Biometric recognition

Like surveillance, biometrics also deals with security. The task of an automated biometrics system is to recognize people based on their physiological or behavioral characteristics [10]. The system recognizes an enrolled subject at a later time either by identifying one person from among many (also known as 1 : N matching) or by verifying that a person's biometric characteristic matches with a claimed identity (also known as 1:1 matching). Biometrics are often preferred over token-based (e.g., card key) or knowledge-based (e.g., password) authentication methods because they cannot be lost or stolen and are tightly linked to a particular person (i.e., cannot be shared). Here, computer vision can contribute to enhancing both the security and the convenience of everyday activities such as airline check-in.

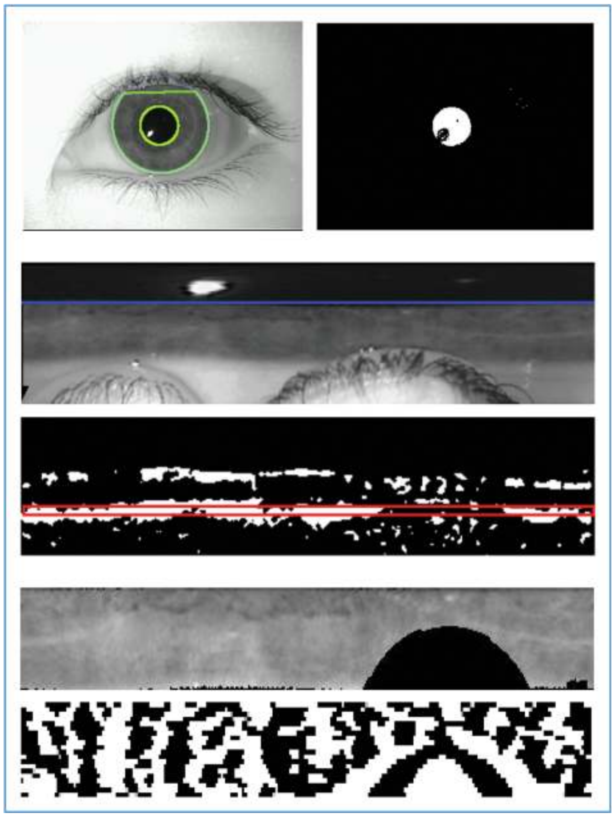

Figure 2

(top left) Iris recognition requires the system to analyze the colored part of the eye. It does this by (top right) first locating the pupil, (middle) unwrapping the circular surroundings, and then determining the outer boundary. (bottom) Actual binary iris code is an encoding of the texture in the extracted annular segment (©2008, IEEE).

A typical biometrics system consists of a front-end sensor, a feature extractor to compute salient attributes from the input signal, and a matcher for comparing two sets of such features [11]. Of particular interest are biometric systems based on the iris (colored part) of eyes since there is a surprisingly high information content in the visual texture of this region [12]. We have built a complete end-to-end system that extracts and compares such iris patterns [13], which illustrates many of the applications and challenges of computer vision.

As shown in **Figure 2**, we start by taking a close-up image of the eye, i.e., typically in the near infrared. The difficulty is then to find and characterize the iris itself (outlined in green). We start by localizing the pupil, which in such images, is usually the darkest region. The upper right in Figure 2 shows a simple intensity-thresholded version of the input image that gives an approximate segmentation of the pupil, albeit with some dropouts due to specular highlights. To better localize the pupil, we fill in small gaps and then fit an elliptical model to the binary blob.

After this, we apply a geometric transform to “unwrap” the rest of the image from the pupil. The top strip image shows this, where the blue line is the edge of the pupil ellipse. We now look for areas with high values of oriented contrast (next strip) to help delimit the outer boundary of the iris. Using this, we carve out the section between the blue and red lines to give a strip containing just the iris information. Note that this corresponds to the green annular region in the original image.

Finally, we convert the texture image to a binary vector (bottom) using the sign of a 1-D log Gabor convolution. The system uses this binary pattern to match against stored representations in a similar format. Note that, often, the complete iris is not visible in an image due to occlusion by the eyelids and lashes. This is indicated by the black “bite” taken out of the unwrapped iris image. To find such regions, we examine a thin region (red box) around the outer boundary in the edge image. If the boundary is not sharp due to occlusion, extra texture, specular reflections, or poor focus, we excise this region using a straight line in the original image (a parabola in the unwrapped version). We do not use the corresponding bits in the matching process.

To match two eyes, we measure the Hamming distance between their binary patterns [14]. Because the images of eyes are sometimes rotated, we compare these patterns with various lateral shifts to find the minimum value (i.e., the maximum number of bits in correspondence). We normalize this by the total number of valid bits in the iris codes to give an overall matching score. If the score is above some threshold, then we consider the two eyes to be from the same person. Of course, as **Figure 3** shows, by varying this threshold, we can alter the tradeoff between the genuine and false accept rates to any combination along the plotted red line. The data in this chart is from a small database of only 450 irises, yet the recognition performance is quite good in general.

The performance becomes even better (blue and green lines) if we exclude the iris images with the worst quality. For the chart in Figure 3, we simply measure quality as the percentage of possible iris bits that are valid (unoccluded). While others have explicitly tried to identify eyelids, eyelashes, highlights, and blurring (e.g., in [15, 16]), by construction, our valid region map incorporates all these effects implicitly. Unfortunately, while a quality threshold of 0.7 gives superb results, this excludes more than 60% of the people in the database, which is not viable in practice. Setting the quality to 0.5 instead still gives a noticeable improvement while only rejecting 10% of the enrollees. This illustrates a general point about vision systems: Good quality input is essential for high accuracy. Furthermore, while many systems can perform well with clean data, the true mark of a viable system is its ability to tolerate at least moderate noise.

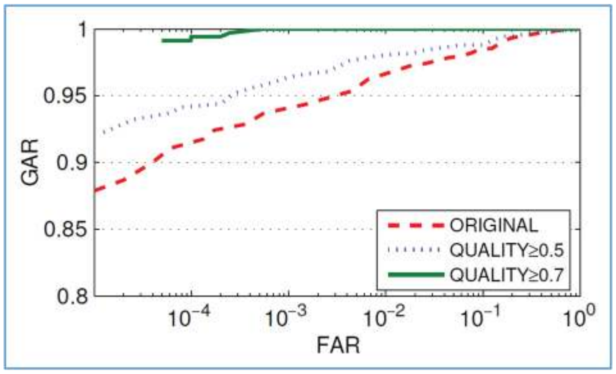

Figure 3

Recognition performance depends on image quality (©2008, IEEE).

We are currently looking to extend the iris system to mobile devices, such as phones, which are starting to have high-resolution cameras. The challenge here is to obtain good-quality images despite questionable hardware and optics and in uncontrolled field conditions. It is important that the eyes be in focus, with reasonable lighting (no sharp shadows) and have little occlusion by the eyelids (or image boundaries). Guiding the user to aim the camera properly and have the device evaluate candidate images automatically necessitates even more computer vision techniques despite the limited computing budget available.

Detection of noncompliance at retail stores

Beyond biometrics, computer vision can also be used for authenticating objects and even processes. For instance, retail stores suffer from a problem euphemistically called “shrink,” which is essentially deliberate theft by customers or cashiers. This costs retailers billions of dollars every year; therefore, catching even a fraction of it using computer vision is worthwhile. One source of shrink is “tag switching,” whereby one item is processed at the checkout stand but the barcode for a different (less expensive) item is scanned. Sometimes, customers print up their own barcodes and affix them to items, which is particularly effective at self-checkout lanes. Other times, a cashier will intentionally scan the barcode of a different item or even a completely separate barcode taped to their wrist. To reduce this sort of activity, we built a camera-based system that performs a check to ensure that the item scanned actually looks plausibly like what its barcode said it was.

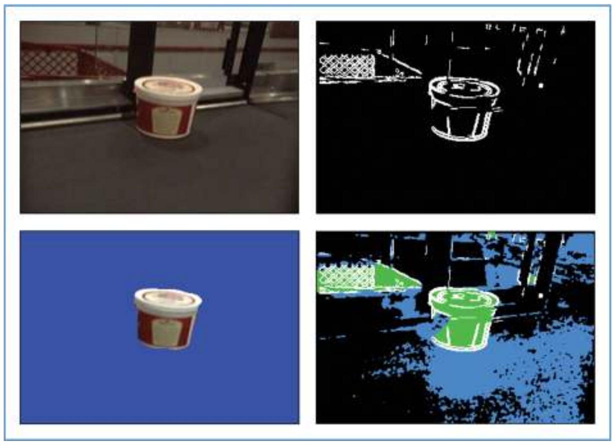

Figure 4

To match the appearance of an object against its barcode, the system must first separate the object from the background. It does this by (upper right) finding edges added to the scene and then (lower right) combines this with regions of color change (©2011, IEEE).

We installed the system on a large IBM System 170 self-checkout lane where it takes pictures of each item as the item progresses down the take-away belt, as shown in **Figure 4**. After locating and segmenting the object in the image, the system computes a number of color, shape, size, and texture features. The representation used is similar to that in our earlier Veggie Vision system [17] that identified loose fruits and vegetables at a grocery store. We then compare these visual features, along with the item's weight, to the item's expected properties based on the barcode that was read. If a significant discrepancy is noted, we raise an exception flag and call a human attendant to the station to investigate the matter.

In this way, we obtained more than twice the fraud detection rate of the standard height-and-weight system while having less than half the false alarm rate. Moreover, the system was able to recognize 6,000 different items while only requiring four training examples per class [18]. While there is a commercially deployed system called LaneHawk** [19] for recognizing items below the basket of a shopping cart, this system operates in identification, not verification mode, and with far fewer classes.

Another source of retail shrinkage is "sweethearting." This is a pervasive type of cashier fraud that is very difficult to catch. It refers to a form of collusion between the cashier and a customer whereby the cashier attempts to give her "sweetheart" free merchandise by deliberately failing to scan certain items. Note that real-time alerts are not necessarily required. Often, it is a history of questionable events that will lead to the firing of a cashier. To combat this, we built a computer vision system called CASE that uses an overhead camera to make sure that all items are actually scanned [20]. While surveillance cameras have long been present at checkouts, the amount of human labor necessary to monitor them has been excessive. An automated system is thus very desirable but must handle the challenges of changing viewpoints, occlusions, and cluttered backgrounds. In addition, retail is a notoriously low-margin business; therefore, it is critical that a fraud detection system be designed with careful control of the false-alarm rate (to reduce human labor) while still being scalable (to contain system costs).

In operation, a typical checkout process can be thought of as including three actions (primitives) in order: pick up, scan, and drop, as illustrated in **Figure 5(a)**. As a first step, we developed a method to recognize a set of such repeated, but non-overlapping, action sequences in a transaction. Our algorithm first detects primitives using robust hand movement analysis and then selectively combines these events into visual scans (see [21] for a survey of similar techniques) [see **Figure 5(b)**]. This is much easier than trying to find and track the objects themselves, which vary widely in appearance and are often occluded by the cashier's hand. A complete action sequence with an associated barcode entry confirms a valid checkout; otherwise, it is a potential candidate for a fraudulent incident or operational error.

Yet, instead of looking at each individual item scanned at the register (as the commercial StopLift** [22] system does), the current CASE implementation performs a holistic analysis of the entire transaction for optimal results. This is because there is a potential parallel overlap of the three actions during checkout for a fast cashier (a potential problem for competitors such as [23]). Yet, because checkout activity still exhibits strong temporal dependencies, we cast the action combination process as an optimization problem. Here, we use a specialized Viterbi algorithm to learn and infer the target events efficiently while simultaneously handling the event overlaps [see **Figure 5(c)**]. In this way, we exploit time-domain data as opposed to trying to squeeze more performance directly out of computer vision.

From a store point of view, reducing false alarms is a critical task, particularly since true fraudulent activity is rare. To mitigate the impact of potential false alarms, we further rank each incident according to its suspiciousness. By reviewing only a portion of the resulting ordered list, a human supervisor can still find a significant portion of the true cases quickly. To perform this automatic ranking, we adopt a two-class support vector machine (SVM) classifier. The features used in classification capture both local and contextual information of an alert, such as the time of occurrence, the duration, and the proximity of the alert. We then use the match score of the SVM as the rank for each incident.

CASE was tested in several stores of two large retailers over the course of several months. During this period, it has demonstrated its effectiveness by successfully detecting

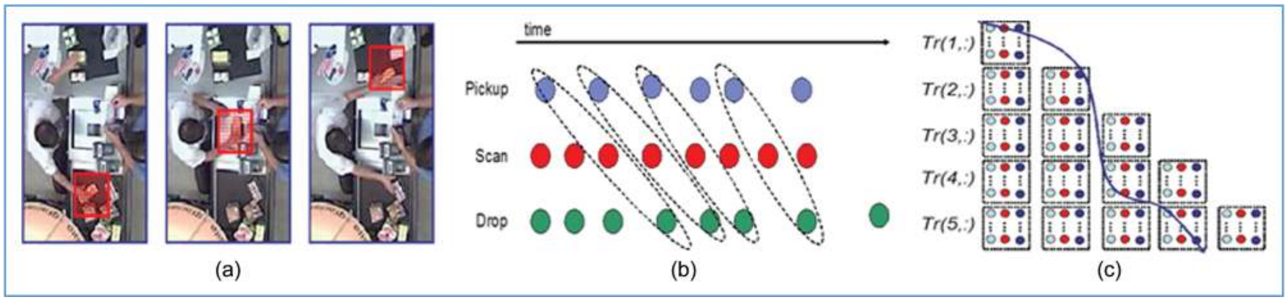

Figure 5

(a) A visual scan consists of three checkout primitives: pick up, scan, and drop (from top to bottom). (b) Given the primitives detected, we are interested in identifying a set of disjoint triplets (pick up, scan, and drop) that correspond to the true scans in a transaction. (c) We use a specialized Viterbi algorithm to find a maximum set of triplets on (blue line) an optimal constrained path (©2009, IEEE).

more than 100 items skipped by the cashiers, either intentionally or unintentionally. It has also identified numerous checkout-related efficiency issues and exceptions, providing additional value to retailers.

We have also conducted a quantitative evaluation of CASE using data recorded at a retail store. This data set includes a video from six lanes during one business day, accounting for a total of 32,969 registered items in more than 1,660 transactions. A manual evaluation found 13 items not registered in the data set. In general, CASE achieves a good detection rate of 62% with a low false-alarm rate of 2.5%. Here, we define the false-alarm rate as the total number of CASE alerts divided by the total number of scanned items. Similarly, the detection rate is the number of true fake scans detected by CASE divided by the total number of fake scans in the data set. After ranking and a cutoff are applied, the false-alarm rate is further reduced to less than 1% but at the expense of sacrificing some detections.

Rail safety

As with both urban and retail surveillance, one advantage of computer algorithms is that they are immune to fatigue and boredom. In some instances, they can also perform a job far more quickly than humans can. For instance, in a rail safety scenario, tracks need to be inspected regularly. Originally, such checks were performed manually, with rail inspectors walking the length of the track. This was extremely slow, as only a few miles per day could be covered. With the advent of computer vision, these inspections can now be performed using cameras and computers at a much higher rate, leading to increases in the safety of railroads (both for passengers and freight).

Track inspection covers a wide spectrum of tasks. Some, such as the measurement of the position, curvature, and alignment of the rails, have already been automated using special track geometry cars. Others, such as monitoring the spiking patterns and rail anchor positions, and the detection of raised, broken, or missing spikes, anchors, and joint bar bolts, are still done manually. Thus, the railroads have a great interest in using machine vision technology for more efficient, effective, and objective inspections. In particular, in conjunction with our customer, we have jointly identified the following key tasks: 1) monitoring of any noncompliant spiking and anchor patterns; 2) detection of spikes whose heads are raised above the tie plate by more than one inch or whose heads are broken off; 3) detection of displaced anchors that have moved more than half an inch away from the tie; and 4) detection of bolts, nuts, and washers missing from rail joint bars. Solving these problems would help keep track gauges from going out of tolerance and thus prevent buckled sections that can lead to the derailment of a train.

The challenges of our system are threefold: design and build a reliable imaging system that meets our stringent video capturing requirements, develop robust video analytics to accurately detect and recognize various objects of interest, and create an end-to-end system that reports exceptions at various levels in real time.

Applying machine vision technology to assist track condition monitoring is not a new research topic. In fact, some systems have been proposed, prototyped, and even deployed, for various specific tasks. Examples include the VisiRail Joint Bar Inspection System [24] developed by ENSCO, with high-resolution scan line cameras and laser sensors; the AURORA system [25] developed by Georgetown Rail for inspecting wood ties, rail seat abrasion, tie plates, anchors, and spikes; and the system developed by MERMEC Group for detecting track surface defects with high-speed line-scan cameras [26]. However, these systems focus on a different set of problems than that detailed here, or their performance has not been reported.

We thus designed and developed a completely new end-to-end system. **Figure 6** shows the imaging setup on the back of a high-rail truck. We use four diagonally aimed

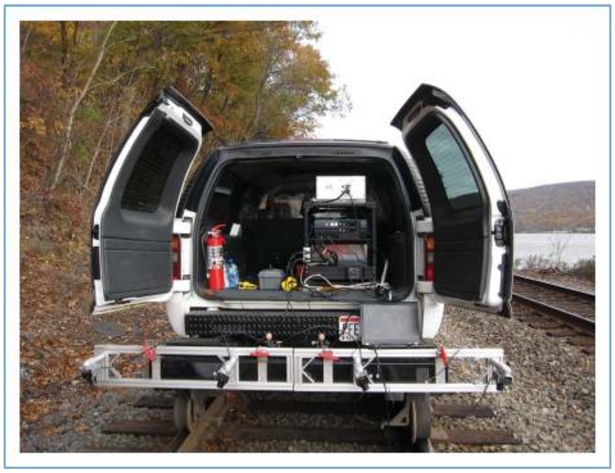

Figure 6

Imaging setup mounted on a high-rail vehicle (©2011, ACM).

cameras in total, producing lateral views of the gauge and field sides of both rails. When the truck travels on the rail, the four captured video streams are compressed and saved to a local disk. We either process these images onboard or save them for later analysis. Using a 12-core IBM System x3650 for real-time processing, we are able to run the inspection vehicle at 10–20 mph (a total of about 30 tie plates per second).

To identify rail components including tie plates, spikes, spike holes, joint bars, and joint bar bolts, we use various video analytics based on features such as color, texture, and edges [27]. We chose these relatively simple localization and segmentation methods because they were both fast and robust, given the typical composition of the images. We also use machine learning to validate further the presence or absence of spikes and anchors using SVM and AdaBoost learners, thus improving the overall error rates. **Figure 7** shows one detection example where the tie plate is indicated by a cyan rectangle and spikes are indicated by green rectangles. Both anchors and spike holes (potentially missing spikes) are bounded by red rectangles.

We evaluated the rail component detection on several test videos that cover a short track segment containing, in total, 797 tie plates, 2,287 spikes, 901 spike holes, and 1,483 anchors. For this data set, the tie plates have the highest detection rate (100%), whereas the spike holes have the lowest (94.2%). Overall, we achieved an average detection rate of 98.2% over all tie plates, spikes, spike holes, and anchors. However, there are inevitable false detections as well, occurring at a rate of about 1.6%.

Although these individual error rates sound good, in practice, they correspond to a combined raw error rate of 370 incidents per mile. Fortunately, we can eliminate most of these errors by taking advantage of camera synchrony. That is, if only one camera sees a tie plate, it is likely a mistake. If, instead, three cameras see a tie plate, then the fourth one should also. Using this heuristic, we reduce the error incidence to a more manageable nine per mile, all without changing the vision algorithms.

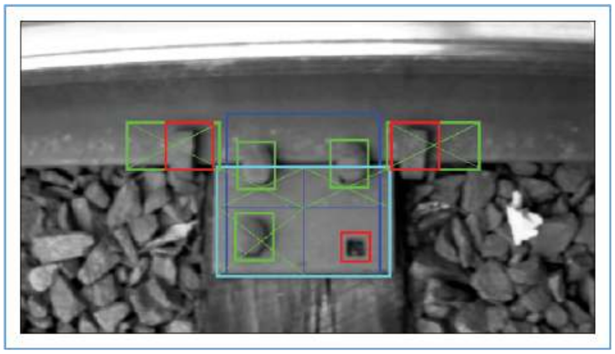

Figure 7

Detection example of various rail components (©2011, ACM).

Once we detect the tie plates, there are other interesting things we can do. Since there is a railroad tie roughly every 12–34 inches, this generates a great number of examples per mile. To exploit this, we also developed an automatic anomaly-detection scheme that lets us find tie plates with abnormal spiking patterns using a completely unsupervised approach. That is, the system learns what the typical local spiking pattern is (it changes depending on track class, curvature, etc.), then flags any plates that are significantly different. To do this, after the tie plate is detected, we identify four characteristic regions of interest (ROIs) based on edge information. These ROIs indicate the expected areas containing spikes and spike holes. We extract ten semantic features from each ROI for a total of 40 features and use these to represent the tie plate region (see faint blue boxes in Figure 7). Next, we measure this feature vector's similarity to a dynamically generated reference set of vectors. Finally, if a tie plate is significantly dissimilar to the majority of plates in the set, we declare it an anomaly and record its Global Positioning System location for later follow-up inspection or repair.

Driver assistance

Computer vision can be used to enhance safety not only for trains but also for passenger vehicles. Here, we describe a system that automatically controls the high beams of a car. Such a system has two benefits. First, it helps prevent inadvertent blinding of an oncoming vehicle if the driver has too slow a reaction speed. Second, it allows the driver to use his high beams more of the time, and hence better assess road conditions, because he no longer feels compelled to continue timidly using low beams.

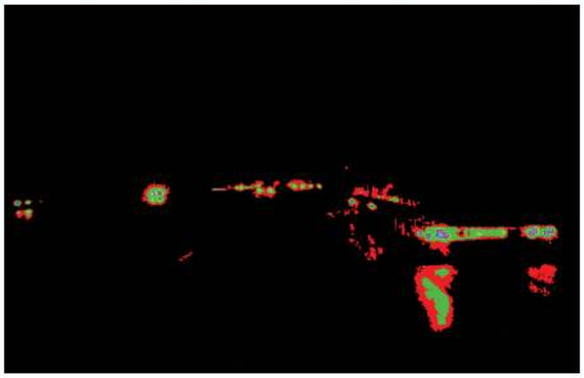

Figure 8

No single intensity threshold can cleanly find all headlights and tail lights, particularly when the windshield is wet (©2011, IEEE).

There are a number of commercial systems, such as Mobileye** [28], Conti [29], and Gentex** [30], already deployed in various brands of vehicles. However, details of the current versions of these algorithms are not publicly available nor are comparative performance results. The academic community has also studied this problem (e.g., [31, 32]), although generally, these systems have been only minimally tested with regard to detection distance and environmental conditions.

We designed our system to address a set of rigid customer-specified constraints. The system had to work in all weather conditions (despite a dirty windshield) and have at least a human level of performance (despite no knowledge of the driver's intentions), and the production cost had to be extremely low. Our industrial partner also wanted a learning system so that they could customize the device's behavior for different countries and driving styles by simply showing it examples. The cost constraint meant that thermal imaging and stereo systems were ruled out; we had to use a standard Video Graphics Array-resolution CMOS color camera. We also had much less compute power than a laptop, i.e., a digital signal processor running at several hundred megahertz; therefore, we had to do very lightweight processing in order to respond in real time. The system also had to function in rain, snow, and fog and not just in dry weather.

The key component in our system [33] is the spot finder. This essentially thresholds the incoming image based on brightness then runs a connected component algorithm to find possible light sources. However, instead of using a single threshold, we found it advantageous to have the system use eight different values, as shown in **Figure 8**. Starting with the highest (most restrictive) segmentation, we allow spots to grow either until they reach some minimum size or until they are merged by the next thresholding level. The eight levels are important since no single level will generate a clean segmentation for both distant headlights, distant tail lights, and nearby tail lights. This technique also helps immensely when there is a thin film of water on the windshield, which smears out all the light sources in the scene.

For speed, we leave the pixel domain as soon as possible and do further work only on the characteristics of the spots found. We use a rule-based system to examine standard blob parameters including area, squared perimeter versus area, bounding box center position, bounding box elongation, and bounding box fill ratio. We also look at the hue, saturation, and intensity for each spot as well as its surrounding "halo" region. The rules determine whether each candidate spot is a headlight, a tail light, or a streetlight, all of which are reasons to switch to low beams. We then apply temporal smoothing to handle spurious detections (and occasional dropouts) before generating the final switching signal.

Using a rule-based system, as opposed to a collection of statistical classifiers, makes the decisions of the system much easier for a human to understand and to adjust by hand if necessary. However, we had to invent a new learning technique, i.e., Structured Differential Learning (SDL) [34], to allow the several hundred thresholds and time constants to be automatically adjusted. SDL operates something like back-propagation over a set of fuzzy predicates and can percolate the credit (or blame) for a decision back to the single most likely parameter in the system. This also means the system can directly learn based on only the desired headlight output state; it just needs a collection of videos and the correct decision for each frame. An earlier approach [35] used an SVM to learn various classes of lights but required an onerously large number of hand-labeled examples for each type of object.

We trained the system on 16.2 hours of video (over 1.7 million frames) and then tested it in a car against several commercially available systems. We tallied both false negatives where the system failed to dim the headlights for some car or was too slow to go to low beams and false positives where the system dimmed the headlights for no apparent reason or was too slow to return to high beams. We normalize these mistakes relative to the correct number of switches that should have been made. The customer-specified vehicle detection ranges were very far: 400 m for tail lights and 1 km for headlights (they are only 1–2 pixels at this distance). The response times were a stringent 1 second to go to low beams and 2 seconds to return to high beams. Note also that we measured performance as event counts, whereas the system was tuned using frames. Nevertheless, as **Figure 9** shows that the final learning system was better than all the commercial systems on the desired metrics while remaining real time and cost appropriate.

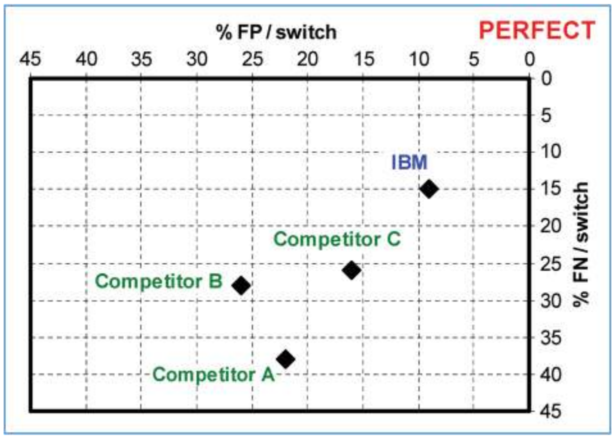

Figure 9

Delivered headlight controller had fewer false positives and false negatives than competing systems in real-world test drives (©2011, IEEE).

Future trends and challenges

While computer vision has made much progress over the last decade, there is still a long way to go before it rivals the performance of a human child. While it is unclear exactly what new techniques and representations might be needed, there are several "grand challenge" problems that could help us fill in the missing pieces.

One frontier for computer vision is object recognition "in the wild," i.e., recognizing objects in images that are not carefully composed and lit. While home photographs found on the web generally do not exhibit the care that goes into professional shots or network television footage, they often center the object of interest, show it largely unoccluded, and avoid sharp shadows on the subject. By contrast, images obtained from a mobile robot or a security camera are much less constrained, yet finding and recognizing objects in such scenarios has obvious economic value (such as biometrics on a mobile phone).

Another challenge is understanding actions in videos, particularly of humans. This is not simply a matter of detecting overt gestures such as waving or tracking body configurations when dancing the Macarena. The interesting case is when humans interact with objects or particular places in the world. For instance, it would be useful to detect when a customer reaches out and removes a can from a shelf in a retail store. Similarly, the sequence of a furtive glance of the head, a subtle extension of the hand, then the disappearance of an article of clothing from a rack can be an indication of potential shoplifting.

Then there is the observation that even if a human is viewing a static 2-D print, his brain seems to force a 3-D interpretation on the system. It is very difficult to defeat this and view the world as patches of color and dominant lines, i.e., the way some artists do. This suggests that there is typically enough information to infer a ground plane, find the bounding contours of objects, and determine both their relative distances and what occludes what. This sort of automatic rough parsing would be very useful for object segmentation and recognition and would also help delineate the environment in which an action was taking place.

We currently do not have the complete solution to any of these problems, but we have made inroads in special cases. The video surveillance work described finding people and cars in unconstrained images. The retail compliance work analyzed the motion of a cashier relative to the checkout task. Additionally, to some extent, the automotive headlight system did a coarse analysis of the overall environment to pick out relevant entities. In all of these cases, working with real-world images and having unflinching performance metrics helped push us into solving the difficult underlying problems. With the continuing drop in price for high-performance computing, there is hope that we can solve even more complicated problems and thus further extend these islands of competence for computer vision.

For more information on projects in IBM's Exploratory Computer Vision Group, please visit: http://www.research.ibm.com/ecvg.

Acknowledgments

The authors would like to thank Russell Bobbit, Arun Hampapur, Charles Otto, Chiao-fe Shu, and Jinyu Zuo, as well as the reviewers for their valuable comments.

**Trademark, service mark, or registered trademark of Evolution Robotics Retail, Inc., Stoplift, Inc., Mobileye Corporation, or Gentex Corporation in the United States, other countries, or both.

References

1. A. Hampapur, L. Brown, R. S. Feris, A. Senior, C. Shu, Y. Tian, Y. Zhai, and M. Lu, "Searching surveillance video," in *Proc. AVSS*, 2007, pp. 75–80.
2. R. Feris, B. Siddiquie, Y. Zhai, J. Petterson, L. Brown, and S. Pankanti, "Attribute-based vehicle search in crowded surveillance videos," in *Proc. 1st ACM ICMR*, 2011, p. 18. DOI: 10.1145/1991996.1992014.
3. R. Feris, J. Petterson, B. Siddiqie, L. Brown, and S. Pankanti, "Large-scale vehicle detection in challenging urban surveillance scenarios," in *Proc. WACV*, 2011, pp. 527–533.
4. Siemens, *Video Surveillance*. [Online]. Available: http://www.buildingtechnologies.siemens.com/bt/global/en/security/video-surveillance/Pages/video-surveillance.aspx
5. ObjectVideo, *Comprehensive Capabilities*. [Online]. Available: http://www.objectvideo.com/thesoftware/capabilities/
6. Honeywell, *Radar Video Surveillance*. [Online]. Available: http://www.security.honeywell.com/industrial/solutions/radar/index.html
7. S. Khan and M. Shah, "Tracking multiple occluding people by localizing on multiple scene planes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 3, pp. 505–519, Mar. 2009.
8. T. Zhao, R. Nevatia, and B. Wu, "Segmentation and tracking of multiple humans in crowded environments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 7, pp. 1198–1211, Jul. 2008.
9. K. Sankaranarayanan and J. Davis, "Attention-based target localization using multiple instance learning," in *Proc. Int. Symp. Vis. Comput.*, vol. 6453, *LNCS*, 2010, pp. 381–392.

10. A. Jain and S. Pankanti, “Beyond fingerprinting,” *Sci. Amer.*, vol. 299, no. 3, pp. 78–81, Sep. 2008.
11. R. Bolle, J. Connell, S. Pankanti, N. Ratha, and A. Senior, *Guide to Biometrics*. New York: Springer-Verlag, 2004.
12. J. Daugman, “High confidence visual recognition of persons by a test of statistical independence,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 11, pp. 1148–1161, Nov. 1993.
13. J. Zuo, N. Ratha, and J. Connell, “A new approach for iris segmentation,” in *Proc. CVPR Biometrics Workshop*, 2008, pp. 1–6.
14. J. Daugman, “New methods in iris recognition,” *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 5, pp. 1167–1175, Oct. 2007.
15. W. Kong and D. Zhang, “Detecting eyelash and reflection for accurate iris segmentation,” *Int. J. Pattern Recognit. Artif. Intell.*, vol. 17, no. 6, pp. 1025–1034, Sep. 2003.
16. N. Kalka, J. Zuo, N. Schmid, and B. Cukic, “Image quality assessment for iris biometric,” in *Proc. SPIE 6202D: Biometric Technol. Human Identification III*, 2006, pp. 1–11.
17. R. Bolle, J. Connell, N. Haas, R. Mohan, and G. Taubin, “VeggieVision: A produce recognition system,” in *Proc. WACV*, 1996, pp. 224–251.
18. R. Bobbit, J. Connell, N. Haas, C. Otto, S. Pankanti, and J. Payne, “Visual item verification for fraud prevention in retail self-checkout,” in *Proc. WACV*, 2011, pp. 585–590.
19. Evolution Robotics Retail, *LaneHawk BOB*. [Online]. Available: http://www.evoretail.com/products/
20. Q. Fan, R. Bobbitt, Y. Zhai, A. Yanagawa, S. Pankanti, and A. Hampapur, “Recognition of repetitive sequential human activity,” in *Proc. Comput. Vis. Pattern Recognit*, 2009, pp. 943–950.
21. P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea, “Machine recognition of human activities: A survey,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 11, pp. 1473–1488, Nov. 2008.
22. M. Kundu, V. Srinivasan, J. Migda, and X. Chen, “Method and apparatus for detecting suspicious activity using video analysis,” U.S. Patent 7 631 808, Dec. 15, 2009.
23. Agilence, *Non-Scan Analysis*. [Online]. Available: http://www.agilenceinc.com/advanced-video-analysis
24. A. Berry, B. Nejikovsky, X. Gilbert, and A. Tajaddini, “High speed video inspection of joint bars using advanced image collection and processing techniques,” in *Proc. World Congr. Railway Res.*, 2008, pp. 1–13.
25. Georgetown Rail Equipment Co., *AURORA: The Future is Finally Here*. [Online]. Available: http://www.georgetownrail.com/aurora.php
26. Mermec Group, *Track Surface Inspection System—TSIS*. [Online]. Available: http://www.mermecgroup.com/diagnostics/track-inspection/62/1/track-surface-inspection.php
27. Y. Li, C. Otto, N. Haas, Y. Fujiki, and S. Pankanti, “Component-based track inspection using machine-vision technology,” in *Proc. 1st ACM ICMR*, New York, 2011, Article 60, DOI: 10.1145/1991996.1992056.
28. G. Stein, O. Hadassi, N. Haim, and U. Wolfovitz, “Headlight, taillight, and streetlight detection,” U.S. Patent 7 566 851, Jul.28, 2009.
29. Continental, *Intelligent Headlamp Control*. [Online]. Available: http://www.conti-online.com/generator/www/de/en/continental/automotive/general/chassis/safety/hidden/lichtassistent_en.html
30. J. Stam, J. Bechtel, S. Reese, J. Roberts, W. Tonar, and B. Poe, “Vehicle lamp control,” U.S. Patent 6 947 577, Sep. 20, 2005.
31. A. Lopez, J. Hilgenstock, A. Busse, R. Baldrich, F. Lumbreras, and J. Serrat, “Nighttime vehicle detection for intelligent headlight control,” in *Proc. 10th Int. Conf. ACIVS*, J. Blanc-Talon, S. Bourennane, W. Philips, D. Popescu, and P. Scheunders, Eds., 2008, pp. 113–124, DOI: 10.1007/978-3-540-88458-3_11.
32. P. Alcantarilla, L. Bergasa, P. Jimenez, M. Sotelo, I. Parra, D. Fernandez, and S. Mayoral, “Night time vehicle detection for driving assistance lightbeam controller,” in *Proc. IEEE Intell. Veh. Symp.*, 2008, pp. 291–296.
33. J. Connell, B. Herta, S. Pankanti, H. Hess, and S. Pliefke, “A fast and robust intelligent headlight controller for vehicles,” in *Proc. IEEE Intell. Veh. Symp.*, 2011, pp. 703–708.
34. J. Connell and B. Herta, “Structured differential learning for automatic threshold setting,” IBM, Yorktown Heights, NY, Tech. Rep. RC-25144, Dec. 2010.
35. Y. Li and S. Pankanti, “A performance study of an intelligent headlight control system,” in *Proc. WACV*, 2011, pp. 440–447.

Received May 4, 2011; accepted for publication June 30, 2011

Sharath Pankanti *IBM Research Division, Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA (sharat@us.ibm.com).* Dr. Pankanti is a Research Staff Member in the Software Research Department at the Thomas J. Watson Research Center. He received a B.S. degree in electrical and electronics engineering from College of Engineering Pune in 1984, an M.Tech. degree in computer science from Hyderabad Central University in 1988, and a Ph.D. degree in computer science from the Michigan State University in 1995. He joined IBM in 1995 as a postdoctoral fellow and later in 1996 became Research Staff Member. He has been manager of Exploratory Computer Vision Group at the T. J. Watson Research Center since 2008. He has led a number of safety, productivity, and security focused projects involving biometrics, multi-sensor surveillance, rail-safety, driver assistance technologies that entail object/event modeling, detection and recognition from information provided by static and moving sensors/cameras. He is an author or coauthor of more than 20 patents and more than 70 technical papers. Dr. Pankanti is a Senior Member of the Institute of Electrical and Electronics Engineers (IEEE) and the Association of Computing Machinery (ACM).

Lisa Brown *IBM Research Division, Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA (lisabr@us.ibm.com).* Dr. Brown received her Ph.D. degree in computer science from Columbia University in 1995. She wrote her thesis on medical image registration while working in the Computer Assisted Surgery Group at the IBM T. J. Watson Research Center. For the past 15 years, she has been a Research Staff Member at IBM. She worked for three years in the Education Group, creating software that enables students to take measurements on images and videos. She is currently in the Exploratory Computer Vision Group. She is well known for her ACM survey paper in image registration, which was extensively cited and translated into several languages. She has published extensively, been an invited speaker and panelist to various workshops, and has filed numerous patents. Her primary interests are in head-tracking, head pose estimation, and more recently object and color classification and performance evaluation of tracking in surveillance.

Jonathan Connell *IBM Research Division, Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA (jconnell@us.ibm.com).* Dr. Connell has been a Research Scientist at IBM for the last 22 years. Prior to that, he received his S.B., S.M., and Ph.D. degrees, all from the Massachusetts Institute of Technology. At IBM he has worked on robot navigation, reinforcement learning, natural language processing, audio-visual speech recognition, video browsing, urban surveillance, retail recognition, automotive image analysis, and biometrics. Most recently, he has started a project on language-guided mobile robots and artificial intelligence. He has been an Adjunct Professor in the Cognitive Science program at Vassar College, has 44 issued U.S. and foreign patents, has published more than 70 technical papers, and is an author of three books. Dr. Connell is a member of AAAI, and a Senior Member of the IEEE.

Ankur Datta *IBM Research Division, Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA (ankurd@us.ibm.com).* Dr. Datta is a Research Staff Member at IBM Research. He received a Ph.D. degree in Robotics from Carnegie Mellon University on a NSF Graduate Fellowship in 2010 and a B.Sc. degree in computer science with University Honors and Honors in the major from University of

Central Florida in 2004. In 2010, he received the prestigious Far-Reaching Research (FRR) Grant from IBM Research to pursue research toward building the next-generation self-calibrating camera networks. He received the best paper award at the ICCV THEMIS workshop in 2009 for work on human motion estimation. He was also named a Computing Research Association (CRA) national finalist for his achievements in 2004. He has published at numerous venues including TPAMI, ICCV, CVPR, ICPR, ICME, among others. His current research interests are in machine learning, surveillance systems, human and object motion estimation, camera calibration, and structured learning.

Quanfu Fan *IBM Research Division, Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA (qfan@us.ibm.com).* Dr. Fan is a Research Staff Member at IBM T. J. Watson Research Center. He received his Ph.D. degre in computer science from the University of Arizona in 2008. He then joined the Exploratory Computer Vision Group at IBM, working on the Smart Surveillance System project. His research interests cover multiple aspects of computer vision, including image and video understanding, human activity recognition, and video analytics. Dr. Fan has published more than 20 peer-reviewed papers in journals and conferences and holds 16 U.S. pending patents.

Rogerio S. Feris *IBM Research Division, Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA (rsferis@us.ibm.com).* Dr. Feris is currently a research scientist at IBM T. J. Watson Research Center, New York, and an affiliate assistant professor at University of Washington. He received a Ph.D. degree in computer science from the University of California, Santa Barbara, an M.S. degree in computer science from University of Sao Paulo, Brazil, and a B.S. degree in computer engineering from the Federal University of Rio Grande, Brazil. He has published approximately 50 papers in peer-reviewed conferences and journals and has 16 patents granted or pending. For more information, see http://rogerioferis.com.

Norman Haas *IBM Research Division, Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA (nhaas@us.ibm.com).* Mr. Haas is a Senior Software Engineer in the Computer Sciences Department at the Thomas J. Watson Research Center. He received a B.S. degree in Physics from SUNY at Stony Brook in 1970, and an M.S. degree in computer science from Stanford University in 1978. He worked at SRI International from 1978 to 1981 and at Symantec from 1981 to 1984. He then joined IBM Research and worked on computational linguistics and cognitive modeling, prior to joining the Exploratory Computer Vision Group in 1992, where he has worked on vision and image processing applications for retailing, biometrics (fingerprints), broadcast television, automotive (in-car vision), and railroading (track inspection). He is a co-inventor of six patents and four technical papers.

Ying Li *IBM Research Division, Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA (yingli@us.ibm.com).* Dr. Li is a Research Staff Member in the Exploratory Computer Vision Group at the Thomas J. Watson Research Center. She received M.S. and Ph.D. degrees in electrical engineering from the University of Southern California in 2001 and 2003, respectively. She subsequently joined IBM at the T. J. Watson Research Center, where she has worked on various projects in the area of multimedia content analysis, e-learning, pattern recognition, and computer vision. Dr. Li has authored or coauthored 25 patents, around 50 peer-reviewed conference and journal papers including ACM MM, ICME, ICPR, CSVT, IEEE Multimedia, and PRL, and four books and book chapters on various multimedia and computer vision related topics. She is on the editorial board of *Journal of Visual Communication and Image Representation*, has chaired a few special sessions at ICME, and co-organized ICME 2009 and MMSP 2007. Dr. Li is a Senior Member of the Institute of Electrical and Electronics Engineers.

Nalini Ratha *IBM Research Division, Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA (ratha@us.ibm.com).* Dr. Ratha is a Research Staff Member in the Exploratory Computer Vision Group at the T. J. Watson Research Center. He received a B.Tech. degree in electrical engineering from Indian Institute of Technology, Kanpur, M.Tech. degree in computer science and engineering from Indian Institute of Technology, Kanpur, and Ph.D. degree in computer science from Michigan State University. He subsequently joined IBM T. J. Watson Research Center, where he has worked on biometrics. In 2006 and 2009, he received an IBM Outstanding Innovation Award for his work on biometrics. He is a coauthor of a popular book on biometrics and edited two other books in biometrics. Dr. Ratha is a Fellow of IEEE and IAPR. He is the President of IEEE Biometrics Council for 2011–2012.

Hoang Trinh *IBM Research Division, Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA (trinhh@us.ibm.com).* Dr. Trinh is a postdoctoral research fellow in the Exploratory Computer Vision Group at the T. J. Watson Research Center, with expertise in computer vision and machine learning. He received his M.S. and Ph.D. degrees in computer science from the Toyota Tech Institute, University of Chicago in 2006 and 2010, respectively. He subsequently joined IBM at the T. J. Watson Research Center, where he has worked on video analytics systems for checkout-related fraud and error detection at retail stores, and more recently, for railroad safety inspection. His work at IBM Research has resulted in three conference papers, seven pending U.S. patents, and a best poster award at the worldwide IBM's annual technical exchange conference, 2010.