Smart Automotive Airbags: Occupant Classification and Tracking

Michael E. Farmer, *Senior Member, IEEE*, and Anil K. Jain, *Fellow, IEEE*

***Abstract*—The introduction of airbags into automobiles has significantly improved the safety of the occupants. Unfortunately, airbags can also cause fatal injuries if the occupant is a child smaller (in weight) than a typical six-year-old. Between 1986 and 2001, 19 infants and 85 children were killed by airbags during relatively minor vehicle collisions. In addition to these infant and child deaths, there have also been seven adults killed by airbags due to their proximity to the airbag during deployment. In response to these deaths, the National Highway Transportation and Safety Administration has mandated that, starting in the 2006 model year, all automobiles be equipped with automatic airbag suppression. The suppression of the airbag based on the type of occupant can be framed as a two-class classification problem, while the suppression of the airbag based on the location of the occupant relative to the airbag can be framed as an occupant-tracking problem. This paper describes an integrated real-time vision-based occupant classification and tracking system using a single grayscale camera with commercially available processing hardware. The classification system has achieved a classification accuracy of approximately 98%. Likewise, the tracking system has demonstrated the ability to detect a dangerous proximity of the occupant relative to the airbag within only 7 ms.**

***Index Terms*—Airbag suppression, occupant classification, occupant tracking.**

I. Introduction

THE INTEGRATION of airbags into passenger vehicles during the 1980s and 1990s has been particularly effective in reducing the number of highway fatalities in the United States. This is primarily due to the fact that automobile passengers in the United States often do not wear their seat belts while driving. The airbag is deployed when the vehicle experiences a crash greater than 14 mi/h, and the airbags were designed to protect a 95th percentile adult male during a 30-mi/h crash [11]. Clearly, the force required to protect such a large occupant is much greater than is required for children, particularly during low-speed crashes. Between 1986 and 2001, 19 infants and 85 children were killed, and in most of these cases, the deaths occurred during low-speed crashes, where it would have been safer if the airbag had been disabled [11]. Fig. 1 shows the effects of an airbag on a rear facing infant seat (RFIS) during deployment [9].

Fig. 1. Effect of airbag deployment on RFIS. (a) RFIS prior to deployment. (b) RFIS during deployment. (c) RFIS after deployment [9].

To address these child safety issues, the U.S. National Highway Transportation and Safety Administration (NHTSA), in May 2001, defined the Federal Motor Vehicle Safety Standard (FMVSS) 208 that mandates automatic airbag suppression when an occupant smaller than or equal (in weight) to a six-year-old child is in the passenger seat, while enabling the airbag when the occupant is a fifth percentile (by weight) adult female or larger [12]. The disabling of the airbag based on the recognition of the occupant type (i.e., infant, child, adult, empty seat) is referred in the NHTSA specification as "static suppression." The NHTSA specification is written around the use of a seat weight sensor that measures the weight of the occupant and infers whether the occupant is an infant, child, or adult. A wide variety of systems have been proposed for solving this problem [1]–[8].

Additionally, during the same timeframe (1986–2001), seven adult passengers were killed by airbags because they were too close to the airbag at the time of deployment [11]. In response to these adult deaths, the NHTSA specification also recommends disabling the airbag if an adult occupant is too close to the airbag [12]. The area within this unsafe proximity is referred to as the automatic suppression zone (ASZ) [12]. This proximity-based suppression is referred to as "dynamic suppression" since the occupant may enter the ASZ due to dynamic events such as normal human motion or severe vehicle braking. Note that, due to the speed with which the airbags deploy, this sensor does not need to monitor the location of the occupant during the crash but rather at the moments prior to the actual crash event.

Dynamic suppression is impossible to achieve with weight-based sensors since a weight sensor cannot infer the motion of the occupant simply by monitoring the static weight load on the seat over time. Other sensors, such as ultrasound and radar, can provide an excellent dynamic suppression mechanism but are poorly suited for static suppression [1]–[6]. Machine vision is unique in its ability to satisfy both static and dynamic requirements with a single sensor. By integrating classification

Manuscript received May 25, 2004; revised November 15, 2004, May 18, 2005, and October 21, 2005. The review of this paper was coordinated by Prof. D. Lovell.

M. E. Farmer is with the Department of Computer Science, Engineering Science, and Physics, University of Michigan-Flint, Flint, MI 48502 USA (e-mail: farmerme@umflint.edu).

A. K. Jain is with the Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824 USA (e-mail: jain@cse.msu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2006.883768

0018-9545/$25.00 © 2007 IEEE

TABLE I
NHTSA OCCUPANT SIZE CATEGORIES IN FMVSS 208 [10]

Occupant Type	Occupant Measurements			
	Height (inches)		Weight (lbs)	
	Min	Max	Min	Max
3 year old	35	39	29.5	39.5
6 year old	45	49	46.5	56.5
5th % female	55	59	103	113
50th % male	69 in. nominal		171 lbs. nominal	

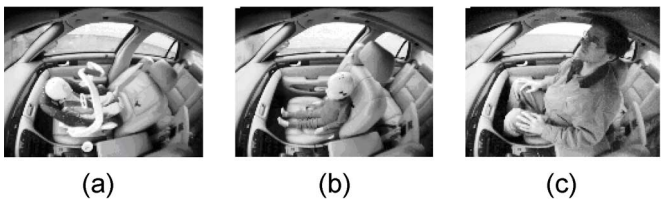

Fig. 2. Examples of the three types of occupants in a vehicle. (a) Infant. (b) Child. (c) Adult.

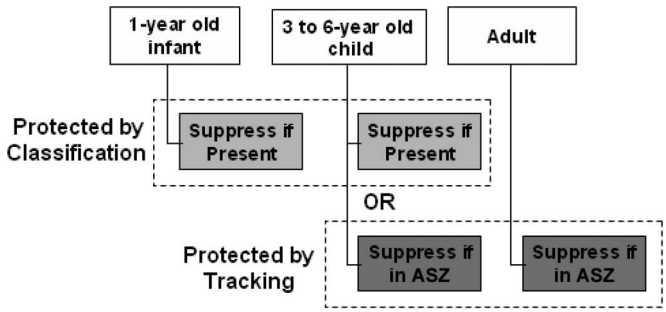

Fig. 3. Overview of the possible mechanisms for suppressing the airbag, depending on the occupant type.

and occupant tracking into a single system, we are able to approach the airbag suppression problem in a novel way compared to other existing systems.

The aim of this paper is to show the applicability of machine vision to the total airbag suppression problem, which encompasses both occupant classification and tracking. The format of the remainder of this paper is as follows: We demonstrate some of the difficulties of the airbag suppression problem throughout the remainder of this section. We provide an overview of our image-processing framework in Section II and then provide details of the algorithms in Section III. We then assess the processing requirements and system hardware required to implement our solution in Section IV. In Section V, we provide the resulting performance for both static and dynamic suppression subsystems. Section VI addresses areas of ongoing and potential future research work.

A. Details of Airbag Suppression

The airbag suppression decision problem addresses the protection of infants, children, and adults. The specific details regarding the sizes of each of the occupants are given in Table I, and example images of each of these occupant types are shown in Fig. 2. NHTSA has defined a set of possible combinations of static and dynamic suppression methods that can be used to protect occupants, as shown in Fig. 3.

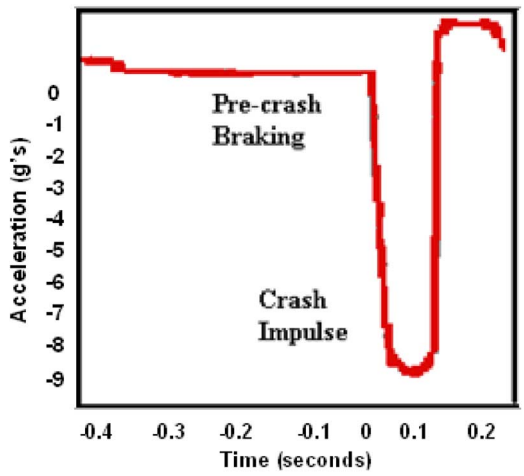

Fig. 4. Definition of dynamics typical for a precrash braking event followed by an actual crash event [14].

The existing seat-based sensors adopt the suppression approach highlighted as protected by classification, where the decision boundary is between a six-year-old child and an adult [3]–[6]. Notice, however, that NHTSA allows for children to be protected "either" using static suppression or dynamic suppression. The infant class, however, must be protected only using static suppression. We will show that by developing a system that is capable of both static and dynamic suppression, we will be able to maximize the safety of all of the occupant types.

The NHTSA specification requires that the classification system must determine the class of the occupant within 10 s of a change of occupant in the vehicle (i.e., from empty seat to adult) [12]. In addition to this time constraint, the NHTSA specification also requires 100% correct classification for a subset of possible seating positions for each of the occupant types listed above [12].

The NHTSA specification for the dynamic suppression system requires detecting the intrusion of an occupant into the ASZ within 20 ms (roughly half the time required for an airbag to deploy) [12]. In addition to this requirement of intrusion time, there is also a requirement regarding the actual accelerations that the occupant is expected to experience during a precrash event. This value is usually specified by the particular automotive manufacturer and is therefore proprietary; however, the typical range of specified accelerations have varied between 0.85 and 1.2 g. These values can be confirmed by looking at the flat portion of the curve in Fig. 4, where an actual high-speed sled test fixture was used to simulate precrash braking deceleration followed by an actual crash event (the sharp dip in the curve). Note that the precrash braking deceleration converges to slightly less than 1.0 g, which is expected for passenger cars with traditional tires (i.e., noncompetitive racing tires). Note in this figure that the time $t = 0$ corresponds to the initiation of an actual crash event (this is where the simulated precrash braking deceleration ends).

There has been considerable interest in using these occupant sensing systems to determine if the passenger seat is empty, in addition to classifying the occupant as a child or adult. If the passenger seat is empty, then it is beneficial to disable the

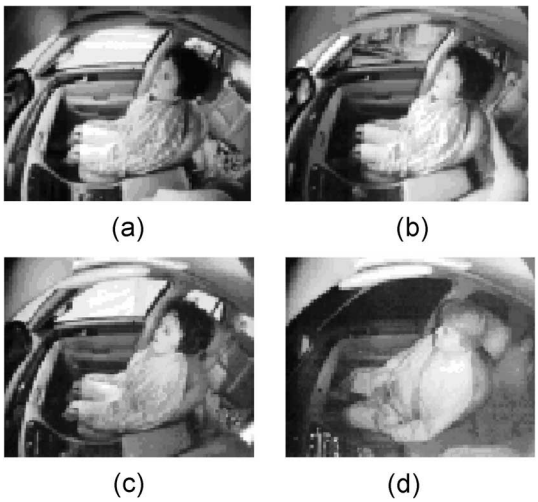

Fig. 5. Effects of external illumination on occupant imagery. (a) Bright sunlight with banding on legs. (b) Bright sunlight across chest. (c) Indoor lighting. (d) Night with supplemental illumination.

airbag to prevent an unneeded deployment. Every year, there is a considerable expense in replacing passenger side airbags after crashes when there was no passenger in the seat, which represents a considerable waste in insurance costs. Consequently, the ability to detect the empty seat and disable the airbag may also be considered a requirement for this system.

B. Complexities of Applying Machine Vision to Airbag Suppression

The use of computer vision in the automobile environment is challenging due to the extreme variations in lighting from bright daylight to dark night, as seen in Fig. 5. In addition to this broad range of illumination levels, the image may also have considerable internal dynamic range due to the simultaneous existence of shadows near the occupant's legs and light bands due to direct sunlight on the head and torso. Since the vehicle is moving, these shadow and light bands are both moving and stationary, which further complicates the problem.

Another complication includes the large intraclass variability for all of the occupants. For the child and infant classes, there are a number of seat types, as well as seating positions, that must be recognized, and the similarity between them is often not very high, resulting in large intraclass variability. For example, there are over 20 infant or child seats currently listed by NHTSA, a few of which are shown in Fig. 6 [10].

The adult class also has a large amount of intraclass variability, as shown in Fig. 7, due to the following four factors: 1) variable in size from the fifth percentile female to the 95th percentile male, which encompasses a 10-in difference in height and 75-lb difference in weight, 2) variability in appearance of adults due to hair and clothing variations, including seasonal variability in clothing (e.g., from only shirts to down parkas and hats), 3) variability in posture of the adult occupant from fully reclined to sitting fully upright, and 4) hand and arm motions (e.g., using a cell phone, drinking a beverage, stretching, etc.).

The problem is further complicated by the small interclass separability between the classes. For example, there is very

Fig. 6. Collage of infant seats showing the large intraclass variability for the infant class.

Fig. 7. Collage of adult images showing the large intraclass variability for the adult class.

little variability between a small fifth percentile female adult and a six-year-old child sitting on a booster seat, as can be seen in Fig. 8. Likewise, constructing the decision boundary between the infant and the child class can also be difficult since the infants and the three-year-olds can use the same child seats. The added requirement of empty seat detection is also very difficult since small children sitting on the seats are completely engulfed within the seat, as shown in Fig. 9.

There are further constraints imposed on the problem due to the unique characteristics of the automotive marketplace. For example, the system must be extremely low cost while also providing service life on the order of 15 years. The goal of this work effort is to investigate the potential performance of a design with the lowest possible cost by using a monocular vision system with no supporting sensors, such as either a second camera for stereo processing, a separate ranging sensor, or supplemental seat-based sensors as were used in [1], [8], or [2], respectively. Additionally, the automotive marketplace is extremely style conscious, which means the system must be

Fig. 8. Small interclass separation between the child class and the adult class. (a) Six-year-old child crash dummy on booster seat. (b) Fifth percentile adult female.

Fig. 9. Example image showing that the occupant can be completely engulfed within the boundary of the vehicle seat.

able to be easily integrated into the vehicle with minimal impact on the appearance of the interior of the vehicle to the passenger.

To summarize, a vision-based system for airbag suppression must be robust enough to handle the following conditions and requirements: 1) large intraclass variability of the various occupant types under consideration, 2) camouflaged classes (e.g., blanketed infants), 3) large variation in light levels (day to night), 4) large lighting variations within an image (shadows to bright direct sunlight), 5) severe automotive environmental conditions, 6) low system cost, and 7) extremely high reliability and performance.

C. Related Work

As mentioned earlier, the bulk of the research into sensor systems for airbag suppression have involved technologies other than computer vision. There are two classes of related work in using computer vision for airbag suppression: one system uses an active time-of-flight illumination source with a high-speed camera [1], and the other approaches rely on stereovision systems [7], [8]. Previous work involving vision for airbag suppression only considered the classification capability of the sensor and had not developed a framework for simultaneously tracking the occupant [7], [8].

The earliest vision work for airbag suppression was performed by Kirk and Krumm [7]. In their paper, they tested both monocular and stereo systems for occupant classification. The system was tested on a small sample of infant seats with good results. Their approach used a set of eigen-images of the occupant types and then mapped the incoming image into the eigen-image space for comparison. They found that using monocular images and performing the eigen-image analysis on the grayscale images were too sensitive to illumination effects. They proposed an alternative approach where they performed the eigen-image analysis on a disparity map generated from stereo images. This system showed better immunity to the variable illumination found in the automotive environment. This system focused on occupant classification and did not address the dynamic suppression capability.

The work by Owechko *et al.* is more recent and also more complete in that it provides a solution for both static and dynamic suppression using occupant classification and occupant position calculation [8]. As with the system defined in [7], this system is a stereo camera vision system. The system uses a number of features of the images for occupant classification including edge density and edge boundary analysis. The edge boundary information is classified using a Hausdorff distance metric where a set of templates are modified using an affine transform and the best overall match corresponds to the occupant type. As in our system, this system recognizes that edge information is less sensitive to illumination effects than the grayscale information used in [7]. This system also integrates motion information into the classification decision by looking for motion above the head rest. If there is motion in that area, then the occupant is most likely an adult. However, recall from Fig. 8 that small stature adults are indistinguishable from children on booster seats based on their height in the seat.

This system also uses motion to perform head location estimation for performing dynamic suppression. The system operates at a 40-Hz image rate and determines from the stereo information the probable location of the head over time. There is no explicit use of the dynamics information of the occupant in the form of a tracking function in the system, which implies that by the time the airbag is disabled, the occupant may already be too close. Our system, however, supports anticipation of proximity to the airbag so that the airbag can be disabled just prior to the occupant becoming too close. Additionally, our system supports the tracking of children as well as adults since we are not limited to initiating tracking based on motion above the head rest.

Lastly, our system has inherent cost benefits over either of these systems since our system is based on monocular vision, thereby requiring only one camera module and the corresponding reduction in signal processing computational power and memory associated with processing only one image.

II. System Overview

The most important design decision in developing an airbag suppression system is the approach for protecting the occupants. There are a number of approaches a vision-based airbag suppression system can take to protect the occupants in the vehicle. One approach is to simply use the vision system in a manner analogous to a weight sensor and classify the occupants based on the six-year-old and adult boundary. As demonstrated in Fig. 8, there is virtually no distinguishable difference in an image between a six-year-old on a booster seat and a small adult. There is also considerable interest in using the airbag suppression system to detect empty seats. However, as Fig. 9 shows, a small child can easily be completely engulfed within the passenger seat. Thus, following the conventional approach adopted by competing technologies such as weight sensors, namely using only classification decisions, leads to two difficult decision boundaries. We propose an alternate approach

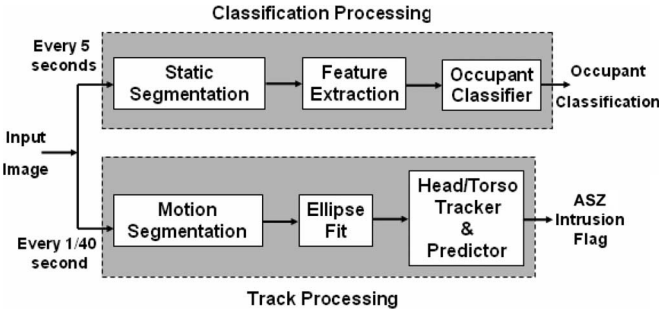

Fig. 10. Software architecture for the computer vision-based smart airbag suppression system.

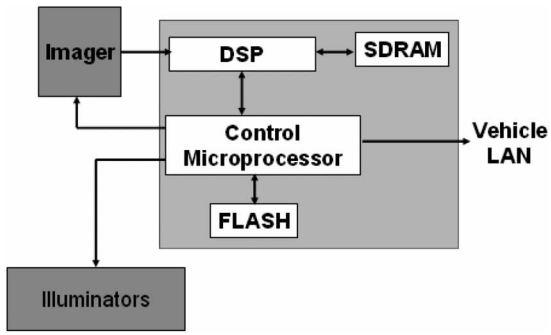

Fig. 11. System hardware architecture.

to disabling the airbag that is more naturally suited to the strengths of the vision system.

A. Operational Concept

We will statically disable the airbag through image classification by defining a two-class problem, where one class is an infant and the other is adult (where adult includes adults as well as any children too large to be considered an infant), as was shown to be allowed by NHTSA in Fig. 3.

We will track any occupant that is labeled as an adult and disable the airbag based on the occupant's proximity to the ASZ. Note that this means the child class is protected either through static or dynamic suppression based on their size. If the child is small, she will be classified as an infant, and the airbag will be disabled. Likewise, if the child is large, she will be classified as an adult and protected by the dynamic suppression. The child can safely be protected via the dynamic suppression in one of two ways: 1) by having the original equipment manufacturers (OEMs) define an ASZ that is based on these smaller occupants, or 2) using the size of the occupant from the tracking information to dynamically define a safe ASZ boundary. We believe this is a more natural means for protecting children smaller than the six-year-old/adult boundary used by static-only suppression systems.

Last, we propose detecting the empty seat using a set of image-processing algorithms that are distinct from those used for either occupant classification or tracking. This capability will be considered as future functionality and, since it is not safety critical, will be addressed at another time.

B. Software Architecture

The design of a pattern recognition system requires three basic components, namely 1) data acquisition and preprocessing, 2) data representation, and 3) decision making [19], [24]. Data acquisition for the smart airbag application consists of the imager module in the hardware subsystem. Data representation and decision making are implemented in the system software. To support both static and dynamic suppression, there are two distinct processing paths. Static suppression is implemented as a classification task, and dynamic suppression is implemented as a tracking task, as shown in Fig. 10. Operationally, the system performs classification of the incoming images. If an image is of the adult class, then the dynamic suppression is enabled.

C. Hardware Architecture

The hardware system is physically composed of a single monochrome digital CMOS camera with a wide field-of-view (FOV) lens, a bank of LED illuminators, a digital signal processor (DSP), and a control microprocessor, as shown in the system hardware diagram in Fig. 11. The hardware consists of four major subsystems, namely 1) the image capture subsystem, 2) the processing subsystem, 3) the memory subsystem, and 4) the vehicle interface subsystem.

The image capture subsystem consists of three elements, namely 1) a lens, 2) an imager chip, and 3) supplemental illuminators for night operation. The lens design is outside the scope of this paper and will not be addressed. The imager chip is a conventional CMOS commercial off-the-shelf device with at least 400×320 pixel resolution. To facilitate the night operation, we do not utilize standard IR filters on the imager chip to benefit from the supplemental IR illumination. The illuminators consist of a bank of infrared LEDs that are geometrically configured to provide roughly uniform illumination over the entire passenger area of the vehicle. The illuminators also contain a diffuser to ensure eye safety and uniform light levels over the FOV.

The processing subsystem consists of a DSP and a microcontroller. This subsystem performs the image-processing and system diagnostics. The memory subsystem provides the real-time memory required for processing the images as well as the long-term nonvolatile memory for storing the program and other critical data. The vehicle interface subsystem connects the airbag suppression system with the other vehicle subsystems via the vehicle bus. It is responsible for the receipt of data from other vehicle subsystems and the transmission of the suppression signal to the vehicle airbag control module.

Section IV-A provides a summary of the processing requirements, and Section IV-B provides details on some potential implementations of the three subsystems that comprise this architecture.

The complete hardware system is located in the roof liner of the vehicle, along the vehicle center line, and near the edge of the windshield, as shown in Fig. 12. This location provides a near profile view of the occupant in the passenger seat, which aids both the classification and the tracking of the occupant. This location also reduces the likelihood of the occupant blocking the sensor and makes styling the sensor into the vehicle easier. The typical FOV required for most passenger

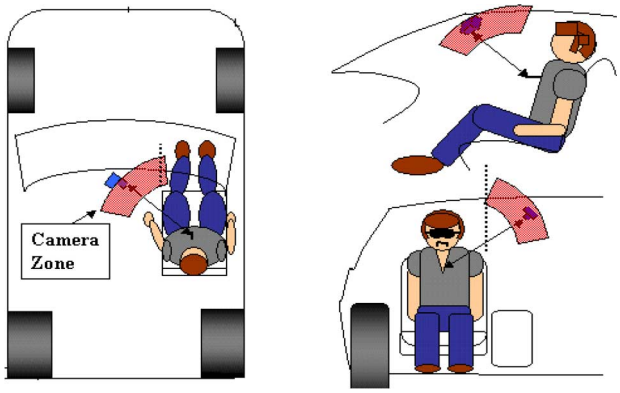

Fig. 12. Installation of the camera system within the vehicle showing plan, profile, and forward viewing angles.

vehicles is roughly 100° vertical FOV and 120°–130° horizontal FOV. This FOV ensures coverage of the occupant from the instrument panel to the rearmost seating position when the seat is fully reclined.

III. Details of System Design

The system is logically subdivided into classification processing and track processing, as shown in Fig. 10. Section III-A will discuss classification processing, and Section III-B will discuss track processing.

A. Occupant Classifier

The occupant classifier determines the class of the occupant and then disables the airbag if the occupant is an infant or if the passenger seat is empty. The first stage in the occupant classifier task is the static image segmentation, as shown in Fig. 10.

1) Static Image Segmentation: The challenge in static image segmentation is to find some common attributes for our occupant classes that we can use for segmenting the occupants from the background. The large intraclass variability of our occupant classes makes defining a set of common characteristics for segmentation extremely difficult. We are, however, in a fixed and structured environment, namely inside a passenger car. In light of this, we will utilize all the available information regarding the interior of the vehicle in the segmentation process. Therefore, rather than trying to segment the occupant from the background, we will try to subtract all of the known background information from the input image. The remaining pixels will then most likely belong to the occupant. This is the well-known approach called background removal.

There are a host of methods available for background removal, including background decorrelation, eigen-background subtraction, and background statistical modeling [41], [59], [60]. The eigen-image subtraction removes the background through global processing (computation of the eigen-background representation). One of the difficulties of applying vision techniques to the airbag suppression application is the existence of moving shadow and light bands across the image. These are not global effects and therefore would be difficult to compensate for by using the eigen-image method. The method of background statistical modeling generates statistics

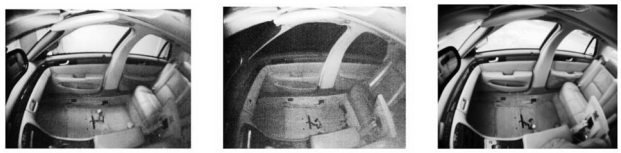

Fig. 13. Empty reference images for each of the three lighting conditions considered. (a) Indoor. (b) Night. (c) Outdoor.

for individual pixels or neighborhoods of pixels throughout the image [59]. Unfortunately, for our application, due to the extreme range of illumination variability, coupled with the fact that the occupant can be wearing any type of clothing (including clothing that matches the background), it is not possible to develop a statistical model that would reliably separate the occupant from the background based on grayscale value.

In background decorrelation, we compare the correlation between the incoming image and a reference image. We develop a robust model of the background by collecting empty vehicle reference images, as shown in Fig. 13. Note that these images do not even contain the vehicle seat. Since the occupant seat can be in a variety of locations, therefore, it cannot easily be removed using background removal. Thus, we will treat the seat as part of the occupant rather than part of the background and classify the occupant and the seat as an entity.

The correlation is computed on gradient images to improve its robustness against the effects of variable illumination. As was also demonstrated by Kirk and Krumm, the image intensity values are too variable due to the illumination effects in the vehicle, and performing correlations on the edge rather than the amplitude images made the system considerably more robust [7].

Since background decorrelation processing uses the gradient image, we have the possibility to use either edge magnitudes or edge directions at every pixel location. Since the structure (and texture) differences in the input and reference images are key attributes for differentiating the occupant from the vehicle interior, we will use the edge directional components comprising the gradient vector $\mathbf{g}(i,j)$

$$\mathbf{g}(i,j) = [g_x(i,j), g_y(i,j)]^T \tag{1}$$

where $g_x(i,j)$ is the gradient in the x-dimension (column wise), $g_y(i,j)$ is the gradient in the y-dimension (row wise), and the superscript T denotes the transpose.

The correlation of these vector fields will be computed over subregions of the two images since we are interested in local correlations to determine the existence of an occupant in a region rather than an overall image correlation. The vector field correlation is written as

$$C = \frac{\sum_A \sum_B \mathbf{g}_I(x,y) \cdot \mathbf{g}_R(x,y)}{\sqrt{\sum_A |\mathbf{g}_I(x,y)|^2 \sum_B |\mathbf{g}_R(x,y)|^2}} \tag{2}$$

where A and B are $N \times N$ regions in the incoming image $\mathbf{g}_I(x,y)$ and the reference image $\mathbf{g}_R(x,y)$, respectively. We have found that $N = 10$ to be effective. Regions of high

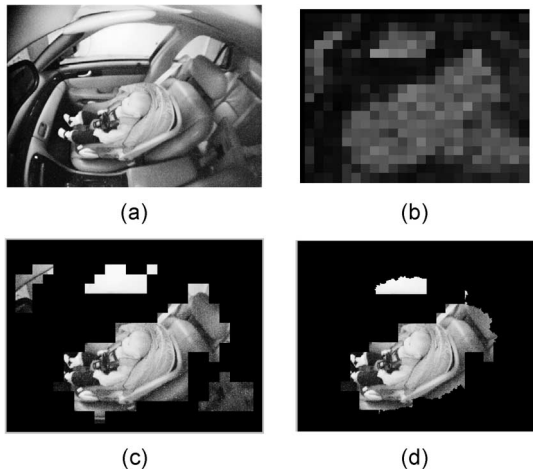

Fig. 14. Example of image segmentation. (a) Infant image. (b) Edge correlation image. (c) Thresholded correlation image. (d) Final segmentation.

correlation vary little from the reference image and therefore can be ignored. Regions of low correlation are kept for further processing since they did not match the known vehicle interior. Once the correlation value for each $N \times N$ region is determined, an adaptive threshold is applied, and any region that falls below that threshold is passed onto the postprocessing module. Fig. 14 (a) shows the incoming image, (b) the decorrelation image, and (c) the resultant image after adaptive thresholding. There are two problems that arise in the output of the background removal, namely 1) holes in the occupant regions and 2) extraneous background regions. Examples of these effects can be seen in Fig. 14(c), where holes in the occupant and extraneous regions in the area of the window and the rear seat are visible.

The two methods utilized for postprocessing are binary morphological operators for filling the image holes and grayscale morphological operators via the watershed algorithm for removing extraneous background pixels. For hole filling, we use a 5×5 closing operation, which fills in small holes and gaps within the segmented object and fills dents in the contours of the object. There are two classes of watershed algorithms, namely 1) blind and 2) with markers. In blind watershed, the image is divided into catchment basins with no other information. In the marker approach, catchment basins in the area of a marker are combined together.

In the airbag suppression application, we have used watershed by markers since we want to use as much contextual information as possible for the segmentation [43]. The markers allow us to define the likely regions for the occupant and the vehicle background. We define a foreground marker and a background marker and then begin the flow from these regions while also merging catchment basins that fall within these marker regions. For the airbag suppression application, we define the markers based on modeling the occupant by its bounding ellipse, as shown in Fig. 15. As we will show in Section III-B, the concept of the bounding ellipse of the occupant is used throughout the system. For occupant tracking we have found the bounding ellipse of the entire occupant to be more reliable than simply the monitoring the head regions, as was used in [8].

Fig. 15. Examples of marker placement. Red ellipse is the measured ellipse in the image, the inner ellipse is the occupant marker, and the outer ellipse is the background marker.

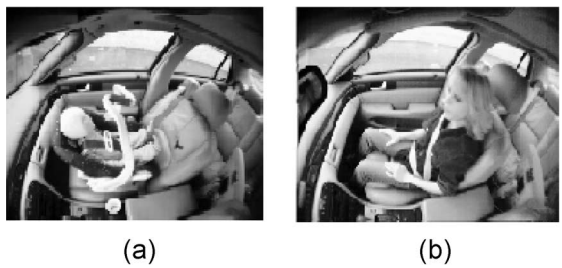

Fig. 16. Demonstration of commonality of color and texture in infant and adult images. (a) Infant image. (b) Adult image.

2) Feature Extraction: Feature extraction involves defining the proper feature space that will provide the best discrimination between the pattern classes under consideration [20]. There are four basic feature spaces that can be utilized in image classification, namely 1) color or grayscale, 2) texture, 3) shape, and 4) spatial locations [20], [23]. In our application, there is no clear separation between the occupant classes based on grayscale and texture. This is because infant, child, or adult occupants can all be wearing garments that have similar color and texture, as can be seen in Fig. 16. The infant, however, is clearly of a different shape than the adult. Therefore, we will use "shape" as our classification feature.

In order to compute features to represent the shape of an object, there are three general design issues that must be addressed, namely 1) boundary pixels versus interior pixels, 2) numeric versus nonnumeric, and 3) information preserving versus noninformation preserving (whether representation supports image reconstruction) [21].

To address the first design issue, we should recall that in our background subtraction approach to segmentation, the seat and occupant are segmented together. The segmented boundary alone is not a reliable discriminator since too much information is lost, and a small adult with their legs up can have a potentially similar boundary to an infant seat on the passenger seat. Likewise, only using internal edge features can be confusing since they may be more dependent on the type of clothing the occupant is wearing than the boundary of the occupant. Therefore, feature extraction schemes that simultaneously exploit both the internal edge and boundary edge features are required. Fig. 17 demonstrates the edge representation for an adult image and an infant image. Note the difference in the structure of the edges in each of the two images.

To compute the edges, we used the x- and y-direction gradients at each pixel. The overall edge amplitude at each

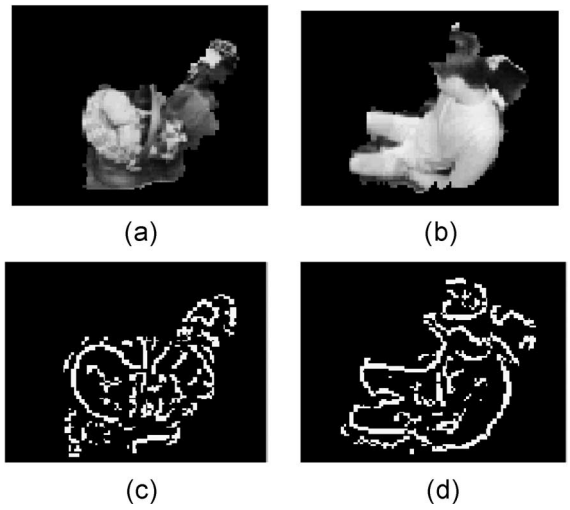

Fig. 17. Edge maps of infant and adult segmented images. (a) Segmented RFIS. (b) Segmented adult. (c) Edge image of RFIS. (d) Edge image of adult.

pixel is computed using the root sum of squares of the directional gradients. We then perform a dynamic thresholding of these edge amplitudes based on preserving 65% of the edge values in the image. We found that this threshold value preserves a reasonable number of lower-level edge amplitude features.

To address the second design issue, we must consider the type of processing architecture that will be used. Since we are using low-cost DSPs for processing, and since the majority of the other algorithms (e.g., the tracker) are largely floating-point numeric computations, we will use the numeric feature representation. For the third design issue, the airbag suppression application has no clear requirement for information preservation in terms of the ability to reconstruct the image from the extracted features. However, we prefer the reconstruction capability during the engineering phase of our development effort in order to better understand the performance of the system when misclassifications do occur.

There are numerous candidate shape features that can be used [20]–[22]. Based on the above three design issues for feature representations, we chose the moments representation of the edge image. Edge-based representation provides knowledge of the internal and boundary structure of the image to address the first design issue. Likewise, the use of moments provides us with numeric data that support reconstruction, thereby addressing the second and third design issues, respectively. We tested the use of geometric, Legendre, and Zernike moments.

The geometric moments M_{mn} of order $(m+n)$ for an $\mathrm{Nrow} \times \mathrm{Ncol}$ image are defined as [25]–[28]

$$M_{mn} = \sum_{i=1}^{\mathrm{Nrows}} \sum_{j=1}^{\mathrm{Ncols}} I(i,j) x(i)^m y(j)^n \tag{3}$$

where it is convenient to scale the row and column dimensions to $x(i) \in [-1,1]$ and $y(j) \in [-1,1]$ to make the moments invariant to the size of the image (but not invariant to the size of the object within the image) being processed. Likewise, the Legendre moments L_{mn} of order $(m+n)$ are defined by [25]

$$L_{mn} = \frac{(2m+1)(2n+1)}{4} \sum_{j=0}^{m} \sum_{k=0}^{n} C_{mj} C_{nk} M_{jk} \tag{4}$$

where C_{mj} and C_{nk} are the Legendre polynomial coefficients, and M_{jk} are the traditional geometric moments of order $(j+k)$.

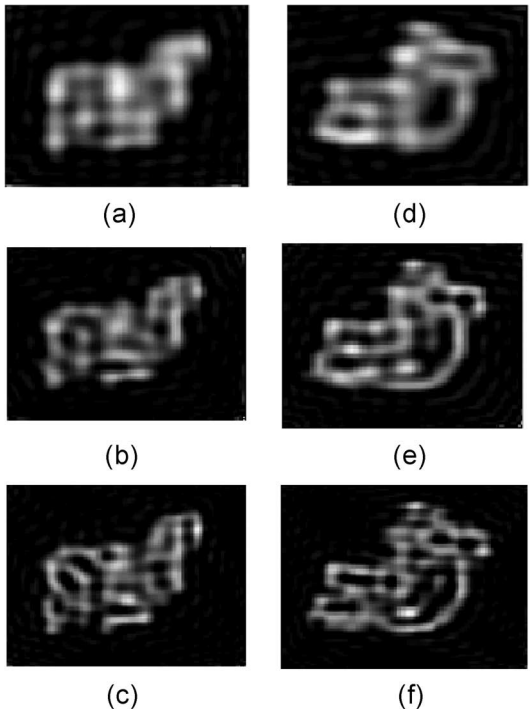

Fig. 18. Reconstructions of infant and adult images. (a) Infant at 25th order. (b) Infant at 35th order. (c) Infant at 45th order. (d) Adult at 25th order. (e) Adult at 35th order. (f) Adult at 45th order for Fig. 17.

We computed moments up to the 45th order since beyond that we exceed the limits of the IEEE double precision. Fortunately, the image reconstruction provided by the 45th-order moment is quite representative of the overall edge structure of the image, as can be seen in Fig. 18 for moments of the 25th, 35th, and 45th order.

Among geometric, Legendre, and Zernike moments, the geometric moments provided greater separability; however, a subset of Legendre moments with the highest Mann–Whitney Z-statistic was slightly better than the corresponding geometric moments, as shown in Fig. 19. Therefore, our features are based on the 45th-order Legendre moments. Zernike moments were provided for comparison since they are very popular for other moments-based recognition tasks such as character recognition [27].

3) Feature Selection and Classification: The complete set of moments, up to the 45th order, generates a total of 1081 features. Feature selection will allow us to reduce this set of features to the most discriminating subset. Feature selection is a critical component of many pattern recognition applications. Feature selection is very simply defined as "given a set of d features, select a subset of size m features that leads to the

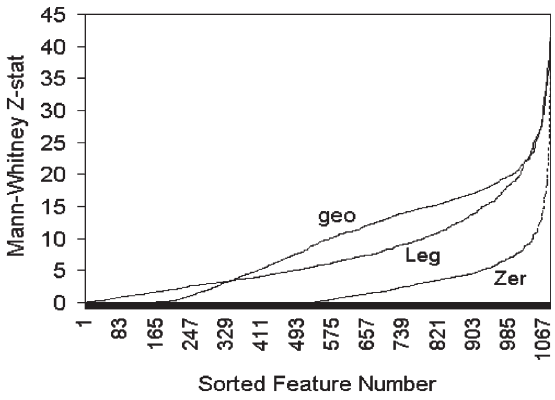

Fig. 19. Sorted Mann–Whitney Z-statistic for 45th-order moments.

smallest classification error" [24]. There are three primary goals in feature selection, namely 1) reduce the processing time to extract features, 2) improve the classification accuracy, and 3) improve the reliability of the system performance estimates (i.e., system generalizability) [34].

Additionally, by removing unnecessary features, the system becomes more robust since these extraneous nondiscriminating features are sources of additional noise in the system. There is a wealth of literature on methods for feature selection. The most straightforward means of feature selection would be to try all possible combinations of subsets of size m from the initial set of d features, but this leads to a combinatorial explosion since the number of subsets increases exponentially. For the airbag suppression application, we use moments up to the 45th order, producing 1081 features, which would result in 2.6×10^{325} possible combinations if subsets of all possible sizes were considered.

Two common and distinct mechanisms for feature selection have been devised to make the problem computationally manageable, namely 1) the wrapper method and 2) the filter method [34], [36], [37]. Wrapper methods use a specific classifier and then use the resultant probability of error from the classifier to select the feature subsets. The feature selection algorithm is wrapped inside the classifier. Of course, the use of a different classifier in the wrapper can lead to the selection of different features [24]. Filter methods analyze features independent of the classifier and use a "goodness" metric to decide which features should be kept. Since we do not know in advance what classifier will provide the best performance, we will choose to adopt a filter-based approach to feature selection.

Since we are interested in finding the subset of features that maximizes the discrimination between classes, it is possible to frame the problem in a purely statistical manner. If each labeled feature value is considered to be from a distinct distribution, then a good feature selector would be the one that chooses features based on maximizing the distances between these distributions. The Mann–Whitney statistic is a nonparametric statistic that determines whether multiple data sets are from the same or different underlying distributions [36]. We apply a best-first search algorithm using the Mann–Whitney statistic to select the features with the greatest discrimination ability [36]. This method processes each feature individually, but it is often advantageous to analyze the interdependencies of features (e.g., correlation among features) and remove features that are highly correlated. To test for these interdependencies, we analyze all of the features for potential correlations, and only the features that are uncorrelated (at some predefined confidence level) are retained. Since we do not know the form of the underlying distributions for the data, we adopt a nonparametric approach to correlation testing. We initially rank all of the features based on their Mann–Whitney values. We then compute the Spearman-R correlation value for all of the features relative to the other features. Then, starting with the most discriminating feature, we remove any feature with lower Mann–Whitney values whose Spearman-R value exceeds a threshold.

The occupant classifier takes as input the feature vector generated by the feature extraction and selection processing. Its output is an estimate of the pattern class that most closely resembles the input-imaged occupant. There are clearly a large number of candidate classifiers that can be used in supervised learning; however, there is no evidence that there is one method that is superior to any other method for every application domain [29], [30]. Consequently, we show results for Bayesian, k-nearest neighbor, and support vector machine (SVM) classifiers.

We use the following assumptions for each of these classifiers. For the Bayesian classifier, we use a quadratic decision boundary, which assumes a different covariance matrix for each of the pattern classes. For the k-nearest neighbor algorithm, we will use $k = 9$. We tested a number of k-values from 3 to 15 and found that after $k = 9$, there was no additional improvement in performance of the classifier. For the SVM algorithm, we use a radial basis function (RBF) kernel with $\sigma^2 = 10.0$. The actual software we used is SVM-Lite [33].

B. Occupant Tracker

The occupant tracker determines when an occupant may cross the ASZ boundary of the airbag. The location of the ASZ is unique for each vehicle. The system must not only determine the actual location of the occupant at any given time but also predict future positions of the occupant. This ability to predict ahead in time will allow us to meet the stringent 20-ms intrusion detection time with a commercial camera, which is only capable of 25-ms frame updates (a 40-Hz camera).

The underlying assumptions for the tracker are that the occupant can be modeled using a bounding ellipse, and that the motion of the ellipse can be reliably tracked to predict when the occupant's head or torso enters the ASZ region. Fig. 20 shows the bounding ellipse assumption, which provides considerable reduction in the complexity of the motion tracking problem by allowing us to only track the centroid, major and minor axes, and in-plane rotation θ of the ellipse. These ellipse parameters serve as the input measurement vector for the subsequent Kalman filter-based tracking. Additionally, this simplifying assumption provides increased robustness against extraneous motion, such as that found outside the passenger window, or due to the occupant's movement of her hands and arms.

1) Motion Segmentation: Motion segmentation extracts the occupant from the interior of the vehicle using the motion of

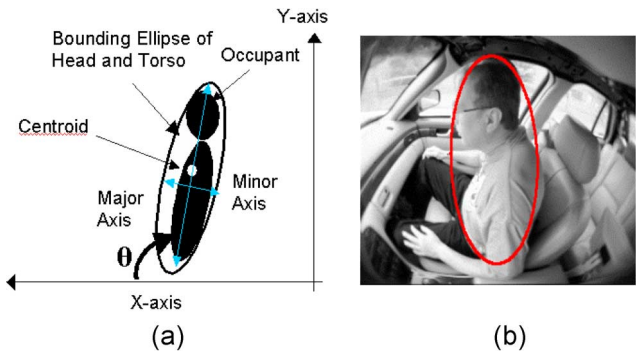

Fig. 20. Human geometry representation for tracking. (a) Definition of ellipse parameters. (b) Ellipse fit to the human occupant.

Fig. 21. Difficulty in using optical flow for occupant segmentation. (a) Person seated normally. (b) Same person putting on a sweater, which generates optical flow but no occupant motion.

the occupant as the cue for segmentation. The segmentation of image sequences must be accomplished at a considerably faster rate than for single-frame static image segmentation defined in Section III-A1. Two common paradigms for motion-based segmentation are 1) template matching and 2) optical flow [45], [46]. Template-based segmentation does not rely on the actual motion in the image but rather attempts to find a known template of the object being tracked in successive image frames. Optical flow methods specifically rely on the motion of the object relative to the background in order to segment the object. Consequently, optical flow techniques are able to estimate the motion of regions of the image quite accurately but do not necessarily provide an accurate boundary of the object [45]. Using optical flow could be particularly problematic for events such as that shown in Fig. 21, where the occupant is not moving but considerable optical flow is generated in the image by the occupant's behavior. Since we require a robust segmentation of the occupant at all illumination conditions, as well as all motion conditions, we will adopt the method of template matching for motion segmentation.

There are three approaches to template matching, namely 1) two-dimensional (2-D) templates, 2) line-based templates, and 3) point-based templates. Point-based templates require the least amount of processing and therefore will be used here to meet the stringent real-time requirements of our application. Within point-based templates, there are two common mechanisms for matching, namely 1) point-by-point matching and 2) point-set level matching. The limitation of point-by-point matching is that if there are gaps in the image contour, then the algorithm will fail to find proper point matches and may not converge properly. Fig. 22 shows a typical sequence (roughly

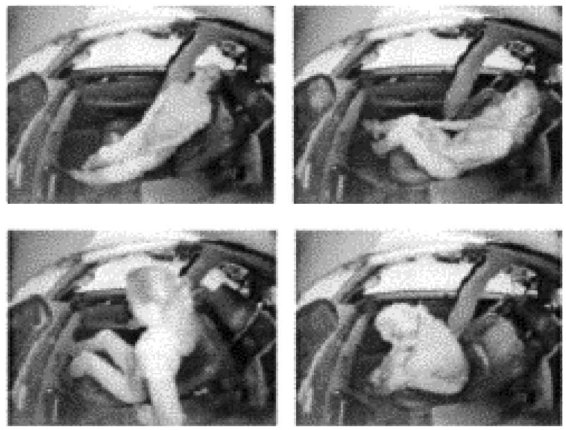

Fig. 22. Difficulty in finding point-to-point matching in occupant template.

every 60 frames) of an adult occupant moving within the vehicle. Note the obvious difficulty in finding point-by-point associations throughout this image sequence. Consequently, we will use point-set matching, in particular the Hausdorff distance, for our application [46].

Template matching across the entire image can be computationally prohibitive and can also cause many false matches in images with complex backgrounds. We take advantage of the relatively high video frame rate and place a bound on the possible distance the occupant can move between video frames to constrain the range of template displacements that must be tested. We use feedback from the track processing to predict the occupant position at the next image frame time. We then form a region-of-interest about this future position based on the occupant velocity from the tracker to account for the possibility that a crash event was initiated since the last frame and the occupant is accelerating forward toward the airbag. This region is typically only one-third of the total extent of the image FOV. Once this region has been extracted, the next step in template matching is to generate two binary images, namely 1) an edge image of the current image frame and 2) the difference image between the current image and the previous image.

The silhouette is computed by multiplying the difference image by the gradient image. This multiplication removes image differences due to simple illumination changes while preserving the difference features near regions containing gradients, such as along the edges of the occupant's head and the window background. The subsequent generation of the template from the occupant silhouette is shown in Fig. 23(a). We then compute the major and minor axes of this contour and compute the upper focal point. From this focal point we then extend radials every N degrees and record the point where these radials hit the extracted boundary as shown in Fig. 23(b). When a gap in the boundary is encountered, no value is entered for that angle. This set of points defines the boundary of the ellipse at the current time. We then perform Hausdorff matching to determine which subset of these points best matches the contour from the last time frame. For Hausdorff matching, we attempt to match these points with the previous sets of points that are translated through a number of rotations and translations based on the occupant velocities ($\dot{x}$ and $\dot{\theta}$) from the tracker.

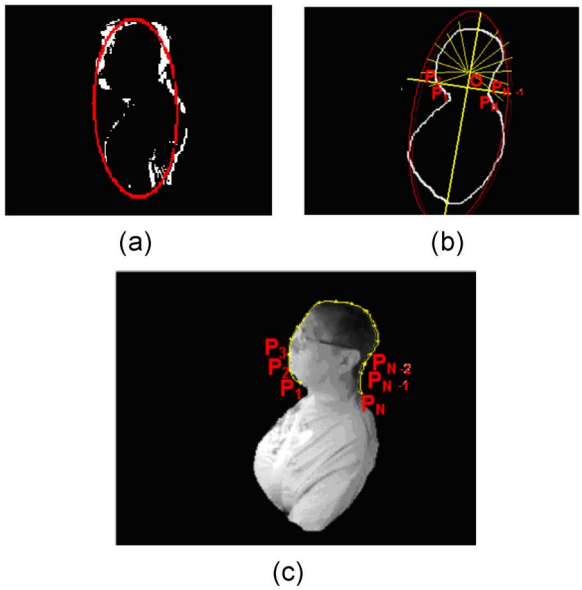

Fig. 23. Method for defining points to be tracked for template-based segmentation. (a) Template matching-based segmentation as input. (b) Spoke arrangement on silhouette. (c) Resultant points overlaid on segmented occupant.

2) Ellipse Fitting: The ellipse is computed from the sets of points that define the occupant's boundary, as computed above. The ellipse-fitting approach is based on least-squares fitting an ellipse to this set of boundary points, as shown in Fig. 23(c) [47]. Each template point is treated as an (x, y) sample value that is used for the least-squares fit. In addition to the template points, we also artificially generate a number of points at the base of the previous ellipse after it was rotated and translated to the current position if there was no discernable motion in this region. The lower portion of the occupant is usually not moving and also does not provide a clearly defined boundary and, consequently, is very noisy. By using the lower portion of the last ellipse, we ensure that the ellipse remains correctly oriented with respect to the lowermost portion of the ellipse on the seat. This is a simplifying assumption that we can use since we have a constrained problem where the occupants are nominally in a seated position while we image them.

The general least-squares fitting of conics does not necessarily generate ellipses particularly if the data are slightly skewed from a true ellipse [48]. Fig. 23(a) shows that the actual contour data extracted from an image to represent an adult occupant is far from a perfect ellipse. Therefore, our application will be very sensitive to the robustness of the ellipse-fitting algorithm.

Rather than performing the least-squares fit to a simple conic, we will need to explicitly assume an elliptical form to ensure robustness [48]. Recall that a general conic is of the form [47]

$$f(\mathbf{a}, \mathbf{x}) = ax^2 + bxy + cy^2 + dx + ey + \mathrm{f} \tag{5}$$

where we can write $\mathbf{a} = [a\, b\, c\, d\, e\, f]$ and $\mathbf{x} = [x^2\ xy\ y^2\ x\ y\ 1]$. We define $f(\mathbf{a}, \mathbf{x}_i) = D$ as the distance from a particular point $\mathbf{x}_i$ to the conic defined by $f(\mathbf{a}, \mathbf{x}_i) = 0$. The problem of fitting an input set of data points to the curve in (5) can then be formulated as minimizing the total distance to all the points in the least-squared error sense [47], i.e.,

$$\hat{\mathbf{a}} = \underset{\mathbf{a}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} f(\mathbf{a}, \mathbf{x}_i)^2 \right\}. \tag{6}$$

Specific constraints must be imposed on $\mathbf{a}$ to ensure a unique and elliptical solution. The constraint to ensure that the output shape is an ellipse is of the form $b^2 - 4ac < 0$ [47]. This quadratic constraint can be enforced on (6) by minimizing the cost function [47] as

$$\frac{1}{b^2 - 4ac} \cdot \sum_{i=1}^{N} f(\mathbf{a}, \mathbf{x}_i)^2. \tag{7}$$

This can be formulated as an eigenvalue problem by defining the constraint equation in matrix form as [47]

$$b^2 - 4ac = \mathbf{a}^T \begin{bmatrix} 0 & 0 & -2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ -2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{a} = \mathbf{a}^T \mathbf{C} \mathbf{a} < 0. \tag{8}$$

For the constraint defined in (8), there will be only one positive eigenvalue, and this eigenvalue defines the solution to the ellipse-specific least-squares fit problem. The corresponding eigenvector corresponds to the vector of the ellipse fit parameters $\mathbf{a} = [a\, b\, c\, d\, e\, f]$ [48].

3) Track Processing: The occupant tracker is responsible for estimating and predicting the location of the occupant relative to the ASZ. The tracker is implemented using a Kalman filter, where inputs are generated from the earlier ellipse fit processing, which determines the $\boldsymbol{x}$ and $\boldsymbol{y}$ coordinates of the centroid, the major and minor axes, and the in-plane rotation angle θ. The basic model equations for the linear Kalman filter are [49]–[51]

$$\mathbf{x}(k) = \mathbf{\Phi}(k-1) \cdot \mathbf{x}(k-1) + \mathbf{v}(k-1) \tag{9}$$

$$\mathbf{z}(k) = \mathbf{M}(k) \cdot \mathbf{x}(k) + \mathbf{w}(k) \tag{10}$$

where $\mathbf{\Phi}(k-1)$ is the state transition matrix, $\mathbf{x}(k-1)$ is the state vector, $\mathbf{v}(k-1)$ is the process noise, $z(k)$ is the measurement value, $\mathbf{M}(k)$ is the measurement matrix, and $\mathbf{w}(k)$ is the measurement noise, all at the current or last times k or $k-1$, respectively [49]–[51].

The model defined by (9) is then used to compute the prediction $\mathbf{x}(k|k-1)$ of the track state vector at time k given that its estimate through time $k-1$ is [49]–[51]

$$\mathbf{x}(k|k-1) = \mathbf{\Phi}(k-1) \cdot \mathbf{x}(k-1|k-1) \tag{11}$$

where $\mathbf{x}(k-1|k-1)$ is the estimate of the state vector at time $k-1$ given the measurement through time $k-1$. The estimate at time k given the measurement through time k is [49]–[51]

$$\mathbf{x}(k|k) = \mathbf{x}(k|k-1) + \mathbf{G}(k) \cdot \mathbf{residue}(k) \tag{12}$$

where $\mathbf{G}(k)$ is the gain (defined below), and the residue, which is the difference between the predicted measurement and the actual measurement, is defined by [49]–[51]

$$\textbf{residue}(k) = \mathbf{z}(k) - \mathbf{M}(k) \cdot \mathbf{x}(k|k-1). \tag{13}$$

The equations for the gain $\boldsymbol{G}$ and covariance matrix $\boldsymbol{P}$ of the predictions are then [49]–[51]

$$\mathbf{G}(k) = \mathbf{P}(k|k-1)\mathbf{M}(k)^T \cdot \left[\mathbf{M}(k) \cdot \mathbf{P}(k|k-1) \cdot \mathbf{M}(k)^T + \mathbf{R}(k)\right]^{-1} \tag{14}$$

and

$$\mathbf{P}(k|k-1) = \mathbf{\Phi}(k-1) \cdot \mathbf{P}(k-1|k-1) \cdot \mathbf{\Phi}(k-1) + \mathbf{Q}(k-1) \tag{15}$$

where $\mathbf{Q}(k)$ is the process noise, $\mathbf{R}(k)$ is the measurement noise, $\mathbf{M}(k)$ is the measurement matrix from (10), and $\mathbf{\Phi}(k)$ is the state transition matrix from (9) [45], [50], [51]. Equations (11)–(15) represent the complete implementation of the Kalman filter for tracking some state vector $\boldsymbol{x}$. Note that the basic Kalman filter implementation implies that the dynamics of the object being tracked can be modeled by a single process model. This implies a single set of dynamic parameters to model the allowable accelerations of the object.

The motion of a human occupant in the vehicle, however, cannot be modeled as a single type of motion but rather as a sequence of distinct motion types. For example, the occupant can be sitting still and then move forward to open the glove box, or the occupant can be moving normally and then be suddenly propelled forward due to precrash braking. The occupant's motion within the vehicle, and particularly his motion toward the airbag, is the critical behavior that must be modeled in order for the motion to be accurately tracked. We propose that the motion of the occupant can be modeled by three possible dynamics states:

1) stationary (no motion);
2) human motion (roughly 0.1-g acceleration);
3) precrash braking motion (roughly 1.0-g acceleration).

The transitions between these various models have been referred to as kinematic discontinuities that must be accounted for to minimize prediction errors [57]. It is well known that complex dynamics can be modeled by sequences of atomic, possibly linear, dynamic states, where the transition between each set of states represents a kinematic discontinuity. Deutscher *et al.* have shown that even the extended Kalman filter alone is unable to account for transitions across these discontinuities [57]. This is expected since the extended Kalman filter can linearize nonlinear dynamic models but still relies on a fixed underlying process model that cannot support these discontinuities [45], [50]. We proposed using interacting multiple model (IMM) filtering for tracking the occupant through these various model transitions [52], [53].

The IMM tracker simultaneously tracks all the possible motion types and then generates a final state by mixing these atomic tracks together. The concept behind IMM tracking is that the dynamics of a tracked object can be modeled as the weighed combination of a set of tracks, corresponding to multiple simultaneous models of the observed system. Each of these trackers is characterized by a different assumption of motion, i.e., a different dynamics or process model [15], [52], [53]. The true motion of the system is then estimated by combining these atomic tracks. The key to using the IMM framework is to define motion models for each possible set of dynamics the system may experience. For our system, the IMM tracker models the stationary, human motion, and precrash braking motions and tracks them individually. Finally, it combines this set of tracks to derive the best single track of the motion of the occupant.

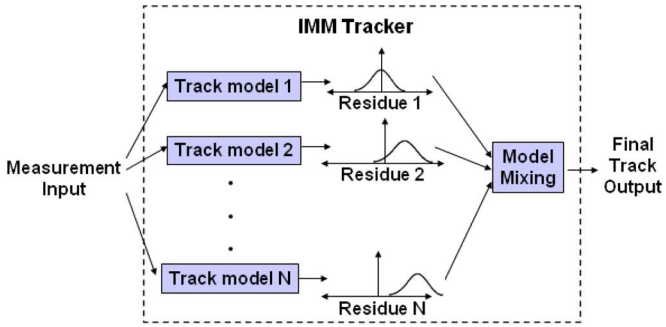

Fig. 24. Architecture of the IMM tracker.

The architecture of the IMM Kalman filter is shown in Fig. 24, where we see that the IMM is basically a set of N Kalman filters executed in parallel. These N filters are then combined based on the residues (or innovations) for each filter, which represent the difference between the expected track position and the actual measured position. Any deviation from Gaussianity in these residues represents a deviation of the dynamics of the system from the specified process or dynamics model.

By combining (11) and (12), the residues of the track for each model m, for $m = 1, \ldots, N$, where N is the number of models, are defined to be [50], [53]

$$\mathbf{Z}_m(k) = \mathbf{z}(k) - \mathbf{M}(k)\mathbf{\Phi}_m(k-1)\mathbf{x}_m(k-1|k-1) \tag{16}$$

where $\mathbf{M}(k)$ is the measurement matrix, $\mathbf{\Phi}_m(k-1)$ is the state transition matrix for model m, and $\mathbf{x}_m(k-1|k-1)$ is the estimate of the state vector for model m through time $k-1$. The innovations for each model are then used to derive the likelihood of the model $L_m(k)$ based on the assumption that $\mathbf{Z}_m(k)$ is zero-mean Gaussian according to [50], [53]

$$L_m(k) = N\left[\mathbf{Z}_m(k); 0, \mathbf{Q}_m(k)\right] \tag{17}$$

where $N[x]$ is a normally distributed variable being evaluated at $\mathbf{Z}_m(k)$, with mean 0, and covariance $\mathbf{Q}_m(k)$, where $\mathbf{Q}_m(k)$ models the process noise for model m at time k.

We must also compute two model probabilities, namely 1) $\mu_{s|m}(k-1)$, which is the probability that model s is correct at time k, given that model m was correct at time $k-1$ and

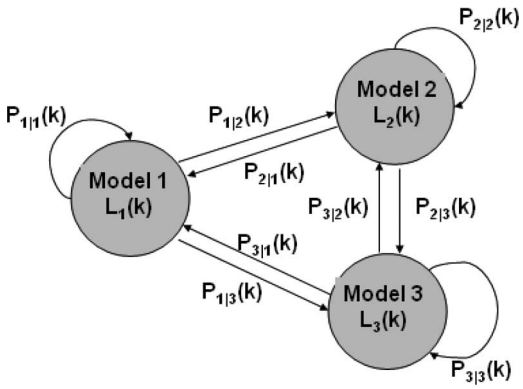

Fig. 25. State transition diagram for a three-model IMM system.

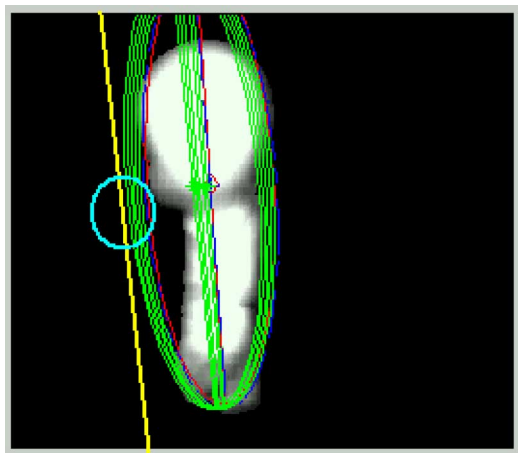

Fig. 26. Geometry of occupant ellipse and ASZ boundary intersection with intersection point highlighted by a circle.

2) the overall probability for each model m, $\mu_m(k-1)$. These are computed by [50], [53]

$$\mu_{s|m}(k-1) = \frac{1}{\sum_{s=1}^{N} p(s|m)\mu_m(k-1)} p(s|m)\mu_m(k-1) \tag{18}$$

and

$$\mu_m(k) = \frac{1}{\sum_{s=1}^{N} L_s(k) \cdot \sum_{t=1}^{N} p(s|t)\mu_t(k-1)} \cdot L_m(k) \cdot \sum_{s=1}^{N} p(s|t)\mu_s(k-1) \tag{19}$$

where $L_m(k)$ is defined in (17), $p(s|m)$ is the state transition probability, which is the probability of the motion of the occupant transitioning from state m into state s at any given time, and $\mu_m(k-1), \mu_m(k)$ are the model probabilities for times $k-1$ and k, respectively, and finally, the summation is over the N possible motion models.

For example, the probability that the occupant will transition from a stationary (seated still state) to a precrash braking state (the driver slams on the brakes) is defined by $p(s = \text{precrash} \mid m = \text{stationary})$. These probabilities $p(s|\,m)$ correspond to the transitions shown in Fig. 25 and are precomputed constants derived empirically from videos of occupant behavior in a vehicle.

Once the probabilities of each model are computed, it is possible to compute the estimate of the state vector for each model. The estimate of the state vector for each model is computed by a weighed combination of the estimates of the state vectors for all of the models, weighted by the probability of the current model being valid, given that each of the other models was valid at the last time instance, according to [50], [53]

$$\mathbf{x}_m(k-1|k-1) = \sum_{s=1}^{N} \mathbf{x}_s(k-1|k-1)\mu_{m|s}(k-1). \tag{20}$$

We then compute the prediction of the state vector to time k, $\mathbf{x}_m(k|k-1)$, using (11) and then compute the estimate to the current time k for each model m using (13) [50], [53] as

$$\mathbf{x}_m(k|k) = \mathbf{x}_m(k|k-1) + \mathbf{G}_m(k)\mathbf{Z}_m(k). \tag{21}$$

The final combined output state vector of the system is a weighed sum over all of the models of the state vectors, weighed by their current model probabilities, according to [50], [53]

$$\mathbf{x}_{\text{combined}}(k|k) = \sum_{s=1}^{N} \mu_s(k)\mathbf{x}_s(k|k). \tag{22}$$

4) ASZ Intrusion Detection: The final task of the tracking subsystem is to predict potential intrusions into the ASZ by the occupant's head or torso. The motion tracker provides the predictions of the occupant's center of mass through the state variable x_{centroid} and his angular tilt forward toward the airbag through the state variable θ. In order to predict the time of the intrusion into the ASZ, two tasks must be performed, namely 1) provide predictions of the occupant's position and orientation of his bounding ellipse at an output rate of 5 ms and 2) detect if the forwardmost edge of his bounding ellipse has entered the ASZ.

The prediction of the position and orientation of the bounding ellipse at high-output data rate is accomplished through predicting the combined state estimate to these future 5 ms intervals through a modification to (11) as

$$\mathbf{x}_{\text{combined}}(k|k-1) = \mathbf{\Phi}^{\text{ASZ}}(k-1) \cdot \mathbf{x}_{\text{combined}}(k-1|k-1) \tag{23}$$

where the time now between $k-1$ and k is 5 ms rather than 25 ms for the camera updates. This new update interval is factored into the intrusion detection state transition matrix $\mathbf{\Phi}^{\text{ASZ}}(k-1)$. We produce eight of these updates, which provides a maximum future prediction of 40 ms, which is to the midpoint of the next image update interval. The sequence of predicted ellipses is shown in Fig. 26. For each of these

occupant ellipse updates, we must compute if the forwardmost point of the ellipse has crossed the ASZ boundary.

A vertical plane in front of the airbag defines the ASZ boundary, but a vertical line in the vehicle is warped into a sloped line (to first order approximation) in the image due to the perspective effects of the camera. Therefore, we must compute the intersection of an ellipse at some orientation angle θ with the ASZ boundary line of predefined slope. An ellipse at some orientation θ is defined by

$$\frac{(x\cos(\theta)+y\sin(\theta))^2}{a^2}+\frac{(y\cos(\theta)-x\sin(\theta))^2}{b^2}-1=0 \tag{24}$$

where a and b are the major and minor axes, and θ is the orientation angle of the ellipse.

Now, we must solve this equation for y, then compute the slope dy/dx, and, last, determine the point along the ellipse that has the same slope as the ASZ boundary line. We want to find the point of equal slopes because this would be the very first point on the ellipse to cross the ASZ boundary line, as highlighted by the circle in Fig. 26. First, define the ellipse as a function $y(x)$ as

$$\begin{aligned} y(x) = {} & \frac{1}{2(a^2\cdot\sin(\theta)^2-b^2\cdot\sin(\theta)^2-a^2)} \\ & \cdot\Big\{2b^2\cdot x\cdot\cos(\theta)\sin(\theta)-2a^2\cdot x\cdot\cos(\theta)\sin(\theta) \\ & +2\cdot\Big[b^4\cdot x^2\cdot\cos(\theta)^2\sin(\theta)^2-2\cdot b^2\cdot a^2\cdot x^2 \\ & \cdot\cos(\theta)^2\sin(\theta)^2+a^4\cdot x^2\cdot\cos(\theta)^2\sin(\theta)^2 \\ & -4\cdot a^4\cdot b^2\cdot\sin(\theta)^2+a^2\cdot b^2\cdot x^2\cdot\sin(\theta)^2 \\ & -2\cdot a^2\cdot b^2\cdot x^2\cdot\sin(\theta)^4+a^4\cdot x^2\cdot\sin(\theta)^4 \\ & +a^2\cdot b^4\cdot\sin(\theta)^2-b^4\cdot x^2\cdot\sin(\theta)^2 \\ & +b^4\cdot x^2\cdot\sin(\theta)^4+a^4\cdot b^2-a^2\cdot b^2\cdot x^2 \\ & -a^4\cdot x^2\cdot\sin(\theta)^4\Big]^{1/2}\Big\}. \end{aligned} \tag{25}$$

Next, the value of x on the ellipse, where the slope of the ellipse at x is equal to the slope of the ASZ boundary line e, is computed. This point where the slopes are equal is defined by

$$\begin{aligned} & x_{\text{equal slopes}} \\ & = -\Big[a^2\cdot\sin(\theta)^2-\text{slope}^2\cdot a^2\cdot\sin(\theta)^2 \\ & \quad -2\cdot a^2\cdot\text{slope}\cdot\cos(\theta)\sin(\theta) \\ & \quad +\text{slope}^2\cdot a^2+\text{slope}^2\cdot b^2\cdot\sin(\theta)^2 \\ & \quad +2\cdot b^2\cdot\text{slope}\cos(\theta)\sin(\theta)-b^2\cdot\sin(\theta)^2\Big]^{1/2} \\ & \quad\cdot\Big[b^2\cdot\cos(\theta)\sin(\theta)+a^2\cdot\cos(\theta)\sin(\theta) \\ & \quad +\text{slope}\cdot a^2\cdot\sin(\theta)^2 \\ & \quad -\text{slope}\cdot b^2\cdot\sin(\theta)^2-\text{slope}\cdot a^2\Big] \\ & \Big/\Big\{-a^2\cdot\sin(\theta)^2+\text{slope}^2\cdot a^2\cdot\sin(\theta)^2 \\ & \quad +2\cdot a^2\cdot\text{slope}\cdot\cos(\theta)\sin(\theta) \\ & \quad -\text{slope}^2\cdot a^2+b^2\cdot\sin(\theta)^2 \\ & \quad -\text{slope}^2\cdot b^2\cdot\sin(\theta)^2 \\ & \quad -2\cdot b^2\cdot\text{slope}\cdot\cos(\theta)\sin(\theta)-b^2\Big\} \end{aligned} \tag{26}$$

where "slope" is the slope of the ASZ boundary line in the image, and θ is the orientation of the ellipse. We now define $x_{\text{intrusion}}$ as the position $x_{\text{equal slopes}}$ from (22) plus an offset corresponding to the centroid of the ellipse

$$X_{\text{intrusion}}=X_{\text{centroid}}-X_{\text{equal slopes}}. \tag{27}$$

Finally, we must now compute the y-value of this intrusion on the ellipse using (25) above, i.e., $y_{\text{intrusion}}=y(x_{\text{intrusion}})$. Then, we need to offset this y-value by its centroid

$$y_{\text{intrusion}}=y_{\text{intrusion}}+y_{\text{centroid}}. \tag{28}$$

Once we have computed this location $(x_{\text{intrusion}},\ y_{\text{intrusion}})$ on the bounding ellipse of the occupant, we can determine if this point is to the right or to the left of the line defining the ASZ by

$$x_{\text{ASZ}}=C_{\text{ASZ}}(1)*y_{\text{intrusion}}+C_{\text{ASZ}}(2) \tag{29}$$

where $C_{\text{ASZ}}(1)$ and $C_{\text{ASZ}}(2)$ are the two coefficients to define the ASZ boundary line. If $x_{\text{intrusion}}\le x_{\text{ASZ}}$, then we declare an ASZ intrusion. This set of calculations is performed for the ten 5 ms time intervals for which we predicted the ellipse parameters. In case of intrusion detection, both the fact of an intrusion and the specific time of the intrusion prediction are forwarded to the airbag controller to disable the airbag.

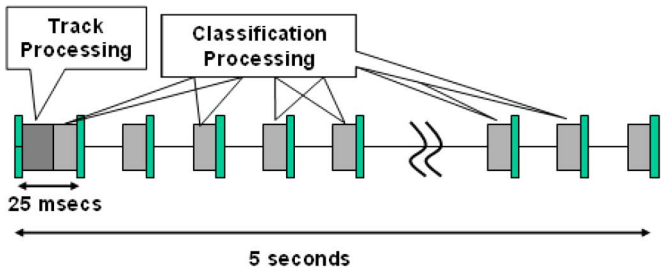

Fig. 27. Processing timeline for airbag suppression.

IV. Processing Requirements and Hardware Implementation

Since this system is addressing a real-world application, it must meet the economic requirements of the automotive marketplace as well. In Section IV-A, we derive processing throughput estimates for each stage of the processing that has been identified for the classification and track processing. In Section IV-B, we will discuss two of the commercially available low-cost DSP chips that meet the processing load defined in Section IV-B.

TABLE II
ORDER OF OPERATIONS FOR THE TRACK PROCESSING FOR AIRBAG SUPPRESSION

Processing Stage	Order of Operations	Comments
I. Segmentation		
1.Extract region of interest	N_{rows} x N_{cols}	
2.Compute edge image	2 x (2 x N_{rows} x N_{cols}) + 4 x N_{rows} x N_{cols}	- gradient calculation in each direction - approx of sqrt(sum of squares of two gradients)
3. Adaptive threshold	2 x N_{rows} x N_{cols} + 2 x N_{rows} x N_{cols}	- compute histogram - compare each pixel and set value
4. Compute abs-diff image	2 x N_{rows} x N_{cols}	
5. Multiply edge and difference images	N_{rows} X N_{cols}	
6. Compute boundary point set	H x 8 x max (N_{rows} s, N_{cols} s)	- H=number of boundary points retained. Compute (X,Y) pairs along H radials.
7. Hausdorf distance for K locations	T x R x H *10 + T x R x (2H x log(2H))	- Sizing for computing rotations and translations +Hausdorff distance matching from [46].
II. Ellipse Fit and Kalman filtering		
1. Ellipse fit & Kalman filter	200 K	
Total Operations:	2.28 Million operations per image frame	

TABLE III
ORDER OF OPERATIONS FOR THE CLASSIFICATION PROCESSING FOR AIRBAG SUPPRESSION

Processing Stage	Order of Operations	Comments
I. Segmentation		
1. Edge Processing	2 x (2 x N_{rows} x N_{cols}) + 4 x N_{rows} x N_{cols}	- gradient calculation in each direction - approx of sqrt(sum of squares of two gradients)
2. Decorrelation Processing	2 x 400 x N_{rows} x N_{cols}	- using a 20x20 size convolutional kernel. Factor of 2 for multiplies and add operations.
3. Adaptive Thresholding	2 x N_{rows} x N_{cols} + 2 x N_{rows} x N_{cols}	- compute histogram - compare each pixel and set value
4. Image Closing	100 x N_{rows} x N_{cols} + 100 x N_{rows} x N_{cols}	- using a 10x10 size convolutional kernel for the erosion and the dilation
5. Watershed Processing	2 x ($N_{rows\,s}$ x N_{cols} + 2 x N_{rows} x N_{cols} x log(2 x N_{rows} x N_{cols}))	- Watershed distance sizing from [44]
6. Region Labeling	100 x N_{rows} x N_{cols}	- estimate from algorithm provided in [42]
II. Feature Extraction		
1. Moments Calculation	4 *(200 x N_{rows} x N_{cols} x 3 +3* $N_{moments}$ x N_{order} x N_{order})	-geometric moments calculation + conversion to Legendre
2. Classification	2* $N_{moments}$ * $N_{training}$ + $N_{training}$ *log2($N_{training}$)	-distance calculation -sorting for k-NN
Total Operations:	300 Million operations per image	

A. Processing Requirements

The most critical real-time requirement of the system is track processing, which must run at the 40-Hz rate of the incoming video stream and cannot drop any of the video frames. The classification processing must execute within 5 s, and consequently, its processing will be time sliced to execute during the idle time between the track processing intervals, as shown in Fig. 27. The estimates for the order of operations for the processing stages and the underlying component algorithms for the track processing are provided in Table II. Likewise, the estimates for the order of operations for the processing stages and the underlying component algorithms for the classification processing are provided in Table III.

Track processing is dominated by the motion segmentation stage of the processing. The critical parameters for the low-level image-processing functions that comprise the motion segmentation stage are the image size (in rows and columns) and the frame rate. Recall from Section I that the incoming image is 400×320 pixels. The final parameter to define is frame rate, and recall for track processing that the image frame rate is 40 Hz. The remaining processing, for track processing, namely ellipse fitting and IMM Kalman filtering, is significantly less than that for segmentation processing.

The Hausdorff distance matching and the ellipse fitting vary by the number of boundary points, which was $H = 32$. The other key factor for Hausdorff processing is the number of translated and rotated positions (T and R, respectively) we will transform the current template to match with the incoming point set. We have found that ten translations and ten rotations are adequate. Ellipse fitting requires the solution of an eigenvalue problem, but it is only calculated once every image frame. Additionally, the form of the eigenvalue problem is such that the matrix is tri-diagonal, which reduces the processing

to on the order of $8/3n^3$, where n is the dimension of the array [66].

The IMM Kalman filter processing varies according to the number of measurements N_{measures} and the number of tracked states N_{states}, which are $N_{\text{measures}} = 1$ for the motion tracker (required for both x_{centroid} and slope trackers), $N_{\text{measures}} = 3$ for the shape tracker (major axis, minor axis, and y_{centroid}), and $N_{\text{states}} = 3$ (position, velocity, and acceleration) for the two motion trackers and $N_{\text{states}} = 9$ (position, velocity, and acceleration for the three state variables) for the shape tracker. Additionally, for the motion tracker, there are three models being tracked for both x_{centroid} and slope trackers (stationary, human, and pre-crash). The most time consuming operation in the tracker is the matrix inversion computed in (14); however, in the worst case, this requires a 3×3 matrix inversion for the shape tracker (since it has $N_{\text{measures}} = 3$). Other operations such as the likelihood calculations in (19) and the ASZ intrusion detection in (25) and (26) are minimized through lookup tables containing the transcendental function values.

The total operation for track processing is on the order of 2.28 MOPs/image frame, which does not include memory accesses (including stores and loads) and loop overheads, which conservatively we will cover with a 8×1 multiplier, resulting in only 18.24 MOPS/image frame, resulting in an aggregate throughput of roughly 730 MOPS.

For classification processing, image segmentation is again a significant driver in the total throughput; however, the moments calculations and the classification must also be considered. For moments and classifier sizing, the other critical parameter is N_{features}, which is the number of features (moments used). Note that feature selection reduces the total number of moments from 1081 to only $N_{\text{features}} = 50$ features based the methods defined in Section III. Since the actual features selected vary slightly between training sets, we will assume an average loading of all the 50 features of the processing based on the mid-point of the 1081 features, which would be a typical feature of the moment of order $M_{\text{order}} = (22, 22)$. For moments calculations, we need to compute only 1/5 of the 1081 exponents to generate the required lower-order moments for combination into Legendre moments in (4). Additionally, to minimize the processing burden, we precompute the factors of the moments calculations related to the exponentials of the position values in (3) since they are constants for every image. This is another effective memory-processing tradeoff. The additional factor of 4 in the moments calculations is for the overhead associated with processing these in double precision in a single-precision architecture. To additionally reduce computation for classification processing, we reduce the resolution of the images $2:1$ in each direction for the moments calculations (to 200×160).

As shown in Table III, the total operations for classification processing are on the order of 300 MOPs, which does not include memory accesses and loop overheads. For classification processing, we will consider a much greater overhead due to the fact that the classification images must be swapped between external and internal memory multiple times to interleave it between the track processing, so we will utilize a 16×1 multiplier (accounts for additional clock cycles required for all external memory accesses), resulting in 4.80 GOPS/image (rather than per second).

Recall that our desired processing time for classification processing was 5 s, then the aggregate processing required to support the classification processing is 960 MOPS. Thus, the combined processing for track and classification processing is 1.70 GOPS, which as we will see in the next section is within the abilities of a new class of DSPs specifically designed for video processing.

B. Hardware Implementation

The most critical aspect of hardware design is the signal processing throughput requirements of the application coupled with the low-cost requirement of the automotive industry. The DSP industry has made dramatic advances in performance over the past five years. This has been driven by two significant market niches, namely 1) telecommunications and 2) desktop video. The telecommunications industry initially drove the development of very high performance DSPs such as the Texas Instruments C62X and C67X processors for cell phone base station processing. While these processors are suitable for multichannel audio processing, the demands of video processing are significantly different due to the sheer volume of pixels and the continuous nature with which the video signal is driven into the processor.

To support the processing of real-time streaming video, the industry has developed alternative architectures that are suited to the unique characteristics of video, namely the fact the pixels can be very efficiently processed in groups using a single-instruction multiple-data architecture that allows multiple pixels to be processed in a single clock cycle. Two processor families specifically designed for video applications are 1) the Texas Instruments C64X and 2) the Analog Devices Blackfin [67], [68].

There are four key issues when selecting the processor, namely 1) instruction set architecture, 2) throughput, 3) memory architecture, and 4) I/O support [including direct memory access (DMA) control functionality]. Both of the processor families support 8-bit, 16-bit, and 32-bit fixed and floating-point processing modes that make them ideal for our application, which has a combination of 8-bit image processing and 32-bit processing for the classifier feature extraction and for the tracker Kalman filter processing [67], [68]. C64X supports 64-bit ALU outputs and has internal 64-bit data buses to support our moments calculations, and Blackfin also has support for double-precision additions and multiplications with only a modest degradation in performance. Additionally, both devices have numerous single clock cycle instructions specifically designed for image processing. For example, C64X has a sum-of-absolute difference operation that performs this function on four pixels in one clock cycle, while Blackfin has extended the SAD function by adding a final accumulator operation while still operating within a single clock cycle [67], [68]. In general, the devices are capable of four simultaneous 8-bit operations per instruction clock cycle, providing roughly 0.5–5.0 billion operations per second (depending on the specific device and its external clock speed), which make them

ideal for this application [67], [68]. The memory architectures of both devices are also specifically designed for video processing with significant amounts of on-chip RAM for zero-wait state memory accesses. Finally, both devices have significant DMA functionality to support real-time input of video streams into the internal memory without interfering with the internal arithmetic units. Both of these families of processors have versions that have been targeted for low-cost embedded applications such as automotive, where the unit prices are well below $10U.S.

V. Experimental Results

The occupant classification and tracking system has been integrated into a full-sized passenger car for both training data collection and performance tests. We will describe the data collection methodologies used for both the classification and the tracking. We will then provide quantitative results for feature selection, occupant classification, and occupant tracking.

A. Data Collection

An extensive data collection effort was undertaken to ensure that an adequate number of images was available for training and testing both the classifier and the tracker. These images are collected in the target vehicles using the camera placement expected in the final production system. A distinct set of images was collected for the classification training and testing and the tracker testing. There was no specific training phase required for the tracker, but rather, the characteristic data were derived from viewing video sequences involving normal occupant motion as well as video sequences involving precrash braking dynamics.

1) Occupant Classifier Data Collection: The classification training data set contains both infant and adult occupants and captures adequate occupant variability to provide generalizable results. The infant images were collected with many of the NHTSA-approved infant seats used in both forward- and rear-facing positions. Due to the difficulty in working with actual infants, we used realistic looking dolls in a variety of clothing. The adult subjects were a combination of crash test dummies and actual human subjects. The details of the training and testing data sets are provided in Tables IV and V, respectively. The independent test set did not necessarily contain the same infant seats and was guaranteed to contain none of the same adults since it was collected at a significantly different point in time with a different set of data collection technicians.

We used three methods for testing the algorithms in the system, namely 1) cross validation, 2) independent validation test data set, and 3) live testing in vehicles. The results below will be for the cross validation and independent validation methods. Note that the distribution of infant and adults is biased toward infants in Table V, but the NHTSA test scenarios are roughly in equivalent proportions due to the large number of possible infant seats that must be tested [10].

2) Occupant Tracker Data Collection: There are two objectives for the data collection for occupant tracker analysis, namely 1) verify tracking of human occupants during normal behavior under various driving conditions and 2) verify tracking of occupants during precrash braking events and verify performance of the ASZ intrusion detection. The first objective is easily performed using real human subjects; however, the second objective cannot safely be accomplished using human subjects. Therefore, a robotic crash test dummy was used for these tests. The images for both of these test methods are collected at the full 40-Hz video rate of the camera.

TABLE IV
Number of Images per Occupant Class in the Training Data Set

Occupant Type	Pattern Class	Number of Images
Rear Facing Infant Seat (RFIS)	Infant	1485
Forward Facing Infant Seat (FFIS)	Infant	1112
Car-bed	Infant	60
5th Percentile adult	Adult	378
50-95th percentile adults	Adult	605
Total images:		**3640**

TABLE V
Summary of the Number of Images per Occupant Class for the Validation Testing Data Set

Occupant Type	Classification	Number of Images
Infant (RFIS+FFIS)	Infant	1283
Adult	Adult	131
Total number of images:		**1414**

Vehicle drive testing collects images of occupants performing typical motions under a variety of driving and illumination conditions, including 1) night, 2) diffuse sunlight, 3) direct sunlight, 4) direct sunlight with moving shadow bands (e.g., sunlight passing through overhead trees), 5) global lighting variation over time (e.g., through freeway tunnels and underpasses), 6) with structured background clutter (e.g., driving past parked vehicles and the buildings), and 7) with cars moving outside the passenger window at different speeds (e.g., on multilane roads). Aside from these environmental and situational conditions, images are collected for multiple occupants in a variety of clothing.

The robotic test fixture supports continuous testing of the occupant tracker during precrash braking events. The known intrusion time is determined from a LED light beam, oriented along the ASZ boundary, and the relative time difference between the actual intrusion event and when the tracker predicts an intrusion is recorded. The experiment can be run for hundreds of iterations. The robotic test fixture is shown in Fig. 28. The system moves the dummy using a constant acceleration motion profile, which matches the dynamics profile of actual vehicle braking. The fixture can simulate any of the following motion profiles: 1) pure translational motion (simulates an

Fig. 28. Robotic test fixture for testing ASZ intrusion time. (a) View through driver's side rear door. (b) View through sunroof.

TABLE VI
SUMMARY OF CLASSIFICATION ACCURACIES FOR THE THREE CLASSIFICATION METHODS

Classifier Method	50/50 Cross-validation	Independent Test set
Bayesian	99.7 %	98.2 %
K-Nearest Neighbor (k=9)	99.5 %	96.7 %
Support Vector Machine (RBF, $\sigma^2 = 10.0$)	98.9 %	96.8 %

TABLE VII
CONFUSION MATRIX FOR k-NEAREST NEIGHBOR WITH $k = 9$ ON THE INDEPENDENT TEST DATABASE

	Classified as Infant	*Classified as Adult*
True Infant	*1270*	*13*
True Adult	*34*	*97*

unbelted occupant), 2) pure rotational motion (simulates an occupant with only the waist belt used), and 3) translational motion followed by rotational motion (simulates an occupant with only a loosely adjusted waist belt).

B. Classifier Results

We use two methods for verifying the performance of the classifiers, namely 1) 50/50 cross-validation testing using the training data set in Table IV and 2) testing on the independent test image database in Table V. The classification results are provided in Table VI. We tested the system using three different classifiers, namely 1) a quadratic Bayesian (Gaussian class-conditional densities with unequal covariance matrices), 2) a k-nearest neighbor (with $k = 9$), and 3) a support vector machine (with a radial basis function kernel). These results are from using the 50 best features from the 1081 possible moment values selected by the Mann–Whitney feature selector. Note that the Bayes classifier with a quadratic decision boundary performed the best in terms of both the 50/50 cross validation and the independent test data set. The detailed confusion matrices for each of the three classifiers are provided in Tables VII–IX.

Note that the classifiers tend to perform better on the infant class than on the adult class. This is explainable since, while there are a number of infant seats that must be classified, they all consist of a rigid shape, which is repeatable over time. The adult class, on the other hand, has a nonrigid shape that is deformable

TABLE VIII
CONFUSION MATRIX FOR THE BAYES CLASSIFIER ON THE INDEPENDENT TEST DATABASE

	Classified as Infant	*Classified as Adult*
True Infant	1282	1
True Adult	25	106

TABLE IX
CONFUSION MATRIX FOR THE SVM CLASSIFIER (USING THE RBF KERNEL WITH $\sigma^2 = 10.0$) ON THE INDEPENDENT TEST DATABASE

	Classified as Infant	*Classified as Adult*
True Infant	*1254*	*29*
True Adult	*16*	*115*

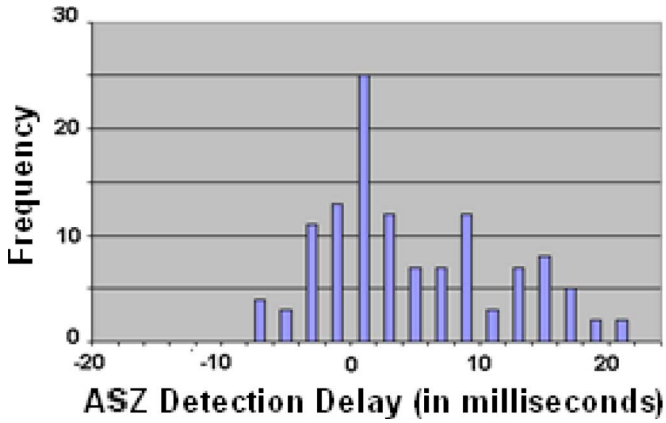

Fig. 29. Histogram of the relative differences between the actual ASZ intrusion and the ASZ intrusion calculated by the tracker.

within some bounds, thereby making it more difficult for the shape-based classifier to capture the complete description of that class. Recall that the NHTSA requirement was for 100% correct classification, which we are close to performing for the infant class, but the adult class needs improvement.

While our system performance is below the required NHTSA accuracy, it is important to note that this performance is for a single (static) image, without any integration of the classification results from multiple image frames (i.e., temporal information). The integration of a sequence of classification results over time should improve our overall classification accuracy. In addition, there is considerable information available from the real-time occupant motions that are computed by the tracker but not currently integrated with the classifier. Section VI discusses ongoing research to address improving the performance of the system for adult occupants.

C. Tracker Results

The tracker results are presented in terms of the difference between the actual and estimated intrusion times into the ASZ from simulated precrash braking maneuvers conducted on the robotic test fixture. The results for the intrusion prediction time errors collected for over 100 iterations of the robotic test fixture are provided in Fig. 29. The distribution of intrusion detections

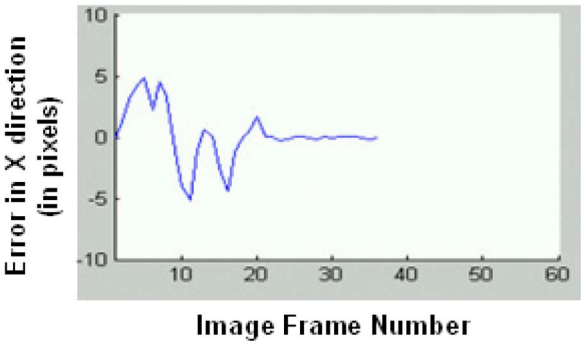

Fig. 30. Track position error during a precrash braking sequence in the direction toward the airbag.

appears to be Gaussian with both negative and positive intrusion time errors due to the fact that the tracker system is actually predicting ahead in time, when the intrusion is to occur. When the tracker predicts that the intrusion will be sooner than it actually occurs, a negative intrusion time error is recorded.

Fig. 30 shows the resultant track accuracies during the entire crash event for one example run of the ASZ tester. The worst-case errors were less than ± 5 pixels during a sequence that includes a transition from fully stationary state to the high-speed precrash braking state, where the dummy reaches a velocity of over 7 mi/h at the time it breaches the ASZ. Note that for the image sizes used in this example, each pixel in the image near the ASZ is less than 0.5 cm. Therefore, the worst-case position predictions were on the order of ± 5 pixels, which correspond to ± 2.5-cm position error.

VI. Summary and Future Work

We have described the design of a vision system for classifying and tracking occupants in a passenger vehicle for the airbag suppression problem. We have provided the details of this unique and important application and defined some of the challenges associated with integrating a real-time vision system into the automobile environment. We have defined a novel approach to the problem that protects the occupant using both classification and tracking of the occupant, thereby improving the safety of the occupant.

For occupant classification, we first introduced a background decorrelation method of background subtraction. We have used a shape-based representation of the occupant based on Legendre moments. We provided the classification results using both a 50/50 cross validation and an independent test data set showing the system achieved an overall accuracy of 98% on single static images; however, the system performs better on infants than adults.

For occupant tracking, we have shown that the IMM Kalman filter provides a robust framework for tracking human motion through the series of discontinuities naturally present in the airbag suppression application. The mode switching capability allows the system to minimize tracking error by quickly adjusting the gain of the filter to reduce tracker latency. The tracking system is capable of minimal error even through events such as high-speed precrash braking maneuvers. The system had an average intrusion delay of 7 ms, which is well within the NHTSA specification of 20 ms. Also, the worst-case error was less than ± 2.5 cm during the highest speed motion of the occupant. By focusing on the motion of the human head and torso and using the simplifying bounding ellipse representation of the occupant, we demonstrated the ability to successfully track the occupant in the presence of motion discontinuities. The simplified bounding ellipse head–torso model allows us to ignore extraneous motions of the subject's limbs and other background motions and effectively track the human subject.

There are a number of areas we wish to continue to explore during our ongoing research. The most critical area of continued research is to improve the classification performance of the adult occupants. There are two avenues we will follow to address this. First, we will explore a better avenue for classifying the adult occupants. The classification methods we used all expect the class to be contained within a cluster of some shape. Recent methods in manifold-based classification schemes may be more applicable for describing the subtle variations in the shape of the adult class [69]. The second area of research to improve the adult performance is to fuse the classification information with another data source. It is clear from the adult performance that additional information is required. Some researchers have proposed stereo algorithms in order to improve the segmentation of the adult from the seat, while others have proposed integrating the vision sensor with seat-based sensors. Rather than incur the additional cost of another camera or additional sensors (and the added processing burden), we will investigate using the occupant motion information from the tracking subsystem and fusing this information with the classification results.

Recall that one additional area of interest from the OEMs was the ability of the system to detect empty seats and disable the airbag. Since we showed the difficulty of using our shape-based classifier to differentiate it from a small child on the seat, we plan on investigating various texture features to develop an empty seat detection algorithm that will execute in cooperation with the existing occupant classification algorithm.

Finally, we plan to extend our current tracker to use real-time changes in the shape information of the bounding ellipse of the adult occupant to infer a three-dimensional (3-D) pose of the occupant from the deformations in the shape of the ellipse in the 2-D image sequences. We believe that these deformations can be exploited to provide this 3-D occupant pose information without the need for complex point tracking and correspondence algorithms.

Acknowledgment

The authors would like to thank Eaton Corporation for their support throughout this project, which included not only financial support for the researchers but also the staffing and human resources required to implement such a complex system. Additionally, the project would not have been possible if not for the incredible individual efforts of the various Eaton Corporation engineers and technicians throughout the project lifecycle. The authors would also like to thank the editors and reviewers for their numerous helpful comments, without which, this document would not have been possible.

REFERENCES

[1] P. Mengel, G. Doemens, and L. Listl, "Fast range imaging by CMOS sensor array through multiple double short time integration (MDSI)," in *Proc. IEEE Int. Conf. Image Process.*, 2001, pp. 169–172.

[2] A. P. Corrado, S. Decker, and P. Benbow, "Automotive occupant sensor system and method of operation by sensor fusion," U.S. Patent 5 482 314, Jan. 9, 1996.

[3] J. H. Semchena, E. Faigle, R. Thompson, J. Mazur, and C. Steffens, Jr., "Apparatus and method for controlling an occupant restraint system," U.S. Patent 5531472, Jul. 2, 1996.

[4] L. Eisenmann, Y. Lu, S. Sauer, and C. Marschner, "Process for the capacitive object detection in the case of vehicles," U.S. Patent 6 442 464, Aug. 27, 2002.

[5] V. C. Patel, T. Thuen, J. K. Hanninen, and H. T. Kuisma, "Force sensor assembly," U.S. Patent 6 089 106, Jul. 18, 2000.

[6] P. B. Blakesley, "Vehicle seat weight sensor," U.S. Patent 6 407 347, Jun. 18, 2002.

[7] J. Krumm and G. Kirk, "Video occupant detection for airbag deployment," in *Proc. IEEE Workshop Appl. Comput. Vis.*, 1998, pp. 30–35.

[8] Y. Owechko, N. Srinivasa, S. Medasani, and R. Boscolo, "Vision-based fusion system for smart airbag applications," in *Proc. IEEE Intell. Vehicle Symp.*, 2002, pp. 245–250.

[9] [Online]. Available: http://www.highwaysafety.org/safety_facts/qanda/airbags.htm

[10] National Highway Transportation and Safety Administration, Dec. 2001. Federal Motor Vehicle Safety Standard # 208.

[11] General Accounting Office, *Vehicle Safety—Technologies, Challenges, and Research and Development Expenditures for Advanced Air Bags*, Jun. 2001.

[12] National Highway Transportation and Safety Administration, *1998 Motor Vehicle Occupant Safety Survey, Volume 3, Child Safety Seat Report*, Jul. 2000.

[13] [Online]. Available: http://www.nhtsa.dot.gov

[14] [Online]. Available: http://www.exponent.com/practices/vehicles/TEC

[15] M. Farmer, R. L. Hsu, and A. Jain, "Interacting multiple models (IMM) Kalman filter for robust high speed human motion tracking," in *Proc. IEEE Int. Conf. Pattern Recog.*, 2002, vol. 2, pp. 20–23.

[16] M. Farmer and A. Jain, "Occupant classification system for automotive airbag suppression," in *Proc. IEEE Conf. Comput. Vis. and Pattern Recog.*, 2003, pp. 756–761.

[17] ——, "Integrated segmentation and classification for automotive airbag suppression," in *Proc. IEEE Int. Conf. Image Process.*, 2003, vol. 3, pp. 1053–1056.

[18] M. Farmer, "Segmentation, classification, and tracking of humans for smart airbag applications," Ph.D. dissertation, Dept. Comput. Sci. Eng., Michigan State Univ., East Lansing, MI, 2004.

[19] A. K. Jain and C. Dorai, "Practicing vision: Integration, evaluation and applications," *Pattern Recognit.*, vol. 30, no. 2, pp. 183–196, 1997.

[20] O. D. Trier, A. K. Jain, and T. Taxt, "Feature extraction methods for character recognition—A survey," *Pattern Recognit.*, vol. 29, no. 4, pp. 641–662, 1996.

[21] R. C. Veltkamp and M. Hagedorn, "State-of-the-art in shape matching," in *Principles of Visual Information Retrieval*. New York: Springer-Verlag, 2001, pp. 87–119.

[22] S. Loncaric, "A survey of shape analysis techniques," *Pattern Recognit.*, vol. 31, no. 8, pp. 983–1001, 1998.

[23] M. Safar, C. Shababi, and X. Sun, "Image retrieval by shape: A comparative study," in *Proc. IEEE Int. Conf. Multimedia and Expo.*, 2000, pp. 141–144.

[24] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 1, pp. 4–37, Jan. 2000.

[25] M. R. Teague, "Image analysis via the general theory of moments," *J. Opt. Soc. Amer.*, vol. 70, no. 8, pp. 920–930, 1980.

[26] R. Mukundan, S. H. Ong, and P. A. Lee, "Image analysis by Tchebichef moments," *IEEE Trans. Image Process.*, vol. 10, no. 9, pp. 1357–1364, Sep. 2001.

[27] R. Mukundan and K. R. Ramakrishnan, *Moment Functions in Image Analysis*. Singapore: World Scientific, 1998.

[28] Q. Lu and K. H. Lee, "Recognition of Chinese characters by moment feature extraction," in *Proc. IEEE Int. Conf. Comput. Process. Orient. Lang.*, 1997, pp. 566–571.

[29] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York: Wiley, 2000.

[30] R. P. W. Duin, "A note on comparing classifiers," *Pattern Recognit. Lett.*, vol. 17, no. 5, pp. 529–536, 1996.

[31] J. Shawe-Taylor and N. Cristianini, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge, U.K.: Cambridge Univ. Press, 2000.

[32] D. Cook, D. Caragea, and V. Honavar, *Visualization for Classification Problems, With Examples Using Support Vector Machines*. [Online]. Available: http://www.public.iastate.edu/~dicook/Limn/svm/paper.pdf

[33] [Online]. Available: Http://svmlight.joachims.org/

[34] M. Dash and H. Liu, "Feature selection for classification," *Intell. Data Anal.*, vol. 1, no. 3, pp. 131–156, May 1997.

[35] M. Kudo and J. Sklansky, "Comparison of algorithms that select features for pattern classifiers," *Pattern Recognit.*, vol. 33, no. 1, pp. 25–41, 2000.

[36] D. Koller and M. Sahami, "Toward optimal feature selection," in *Proc. 13th Int. Conf. Mach. Learn.*, 1996, pp. 197–243.

[37] D. W. Aha and R. L. Bankert, "A comparative evaluation of sequential feature selection algorithms," in *Learning From Data: AI and Statistics*. New York: Springer-Verlag, 1996.

[38] R. Lowry, *VassarStats: Web Site for Statistical Computation*. [Online]. Available: http://faculty.vassar.edu/lowry/VassarStats.html

[39] R. J. Larsen and M. L. Marx, *An Introduction to Mathematical Statistics and Its Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1986.

[40] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: Principles and practice of background maintenance," in *Proc. IEEE Int. Conf. Comput. Vis.*, 1999, pp. 255–261.

[41] N. R. Pal and S. K. Pal, "A review of image segmentation techniques," *Pattern Recognit.*, vol. 26, no. 9, pp. 1277–1294, 1993.

[42] L. Shapiro and G. Stockman, *Computer Vision*. Englewood Cliffs, NJ: Prentice-Hall, 2001.

[43] J. B. T. M. Roerdink and A. Meijster, "A watershed transform: Definitions, algorithms, and parallelization strategies," *Fundamenta Informaticae*, vol. 41, no. 1/2, pp. 187–228, 2000.

[44] P. Felkel, M. Bruckschwaiger, and R. Wegenkittl, "Implementation and complexity of the watershed-from-markers algorithm computed as a minimal cost forest," *Comput. Graph. Forum*, vol. 20, no. 3, pp. 26–35, Sep. 2001.

[45] S. Beauchemin and J. Barron, "The computation of optical flow," *ACM Comput. Surv.*, vol. 27, no. 3, pp. 433–467, 1996.

[46] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, "Comparing images using the Hausdorff distance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 9, pp. 850–863, Sep. 1993.

[47] M. Pilu, A. W. Fitzgibbon, and R. B. Fisher, "Ellipse-specific direct least-square fitting," in *Proc. IEEE Int. Conf. Image Process.*, 1996, pp. 599–602.

[48] M. Pilu, A. W. Fitzgibbon, and R. B. Fisher, "Ellipse-specific direct least-square fitting," in *Proc. IEEE Int. Conf. Pattern Recog.*, 1996, pp. 253–257.

[49] A. Gelb, *Applied Optimal Estimation*. Cambridge, MA: MIT Press, 1974.

[50] M. Pekkarinen, "Multiple model approaches to multisensor tracking," M.S. thesis, Tampere Univ. Technol., Tampere, Finland, 1999.

[51] G. Welch and G. Bishop, "An introduction to the Kalman filter," Dept. Comput. Sci., Univ. North Carolina, Chapel Hill, NC, TR-95-041, 2002.

[52] H. A. P. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with Markovian switching coefficients," *IEEE Trans. Autom. Control*, vol. 33, no. 8, pp. 780–783, Aug. 1988.

[53] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan, "Interacting multiple model methods in target tracking: A survey," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 34, no. 1, pp. 103–123, Jan. 1998.

[54] R. A. Singer, "Estimating optimal tracking filter performance for manned maneuvering targets," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-6, no. 4, pp. 473–483, Jul. 1970.

[55] M. Moore and J. Wang, "An extended dynamic model for kinematic positioning," *J. Navigation*, vol. 56, no. 1, pp. 79–88, 2003.

[56] M. Isard and A. Blake, "CONDENSATION—Conditional density propagation for visual tracking," *Int. J. Comput. Vis.*, vol. 29, no. 1, pp. 5–28, 1998.

[57] J. B. N. Deutscher, B. Bascle, and A. Blake, "Tracking through singularities and discontinuities by random sampling," in *Proc. IEEE Int. Conf. Comput. Vis.*, 1999, vol. 2, pp. 1144–1149.

[58] J. K. Aggarwal and Q. Cai, "Human motion analysis: A review," *Comput. Vis. Image Underst.*, vol. 73, no. 3, pp. 428–440, Mar. 1999.

[59] I. Haritaoglu, D. Harwood, and L. S. Davis, "W4: Real-time surveillance of people and their activities," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 809–830, Aug. 2000.

[60] N. Oliver, B. Rosario, and A. Pentland, "A Bayesian computer vision system for modeling human interactions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 831–843, Aug. 2000.

[61] J. Rittscher and A. Blake, "Classification of human body motion," in *Proc. IEEE Int. Conf. Comput. Vis.*, 1999, pp. 634–639.
[62] C. Bregler, "Learning and recognizing human dynamics in video sequences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 1997, pp. 568–574.
[63] C. Wren, A. Azarbayejani, T. Darrel, and A. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 780–785, Jul. 1997.
[64] J. Yamato, J. Ohya, and K. Ishii, "Recognizing human actions in time-sequential images using hidden Markov models," in *Proc. IEEE Conf. Comput. Vis. and Pattern Recog.*, 1992, pp. 379–385.
[65] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," in *Readings in Speech Recognition*, A. Waibel and K.-F. Lee, Eds. San Francisco, CA: Morgan-Kaufman, 1990.
[66] [Online]. Available: http://www.cs.utk.edu/~dongarra/etemplates/node93.html
[67] Texas Instruments, *TMS320C64x Technical Overview*, 2001. No. SPRU395B.
[68] Analog Devices, *Blackfin Embedded Processor ADSP-BF536/ADSP-BF537 Preliminary Technical Datasheet*, 2005.
[69] B. Raytchev, I. Yoda, and K. Sakaue, "Head pose estimation by nonlinear manifold learning," in *Proc. IEEE Int. Conf. Pattern Recog.*, 2004, pp. 462–466.

Michael E. Farmer (M'89–SM'02) received the B.S. degree in applied physics from Columbia University, New York, NY, the M.S. degree in physics from the University of Michigan-Flint, the M.S. degree in management of technology from the University of Minnesota, Minneapolis, and the Ph.D. degree in computer science from Michigan State University, East Lansing.

He held various positions in industry, including the Lead Software Technologist with Eaton Corporation and the Director of Advanced Programs with Unisys. He is currently an Assistant Professor with the Departments of Computer Science, Engineering Science, and Physics, University of Michigan-Flint. He holds roughly 20 patents in radar and vision systems design for a variety of applications including military surveillance, precision agriculture, and automotive safety. His research interests include context-based image segmentation and object motion and shape tracking and their application to advanced automotive safety.

Dr. Farmer is a member of the Association for Computing Machinery (ACM) and the Society of Automotive Engineers (SAE). He is also a member of the National Who's Who for executives and professionals.

Anil K. Jain (S'70–M'72–SM'86–F'91) received the B.Tech. degree from the Indian Institute of Technology (IIT), Kharagpur, India, and the M.S. and Ph.D. degrees from Ohio State University, Columbus, all in electrical engineering.

He is currently a University Distinguished Professor with the Departments of Computer Science and Engineering and Electrical and Computer Engineering, Michigan State University, East Lansing. He is also a member of the study team on Whither Biometrics being conducted by the National Academy of Sciences (CSTB). His research interests include statistical pattern recognition, data clustering, texture analysis, document image understanding, and biometric authentication. He holds six patents in the area of fingerprint matching. He is the author of a number of books, including *Biometric Systems, Technology, Design and Performance Evaluation* (Springer, 2004), *Handbook of Face Recognition* (Springer, 2004), *Handbook of Fingerprint Recognition* (Springer, 2003), which received the 2003 PSP award from the Association of American Publishers, *BIOMETRICS: Personal Identification in Networked Society* (Kluwer, 1999), *3-D Object Recognition Systems* (Elsevier, 1993), *Markov Random Fields: Theory and Applications* (Academic, 1993), *Neural Networks and Statistical Pattern Recognition* (North-Holland, 1991), *Analysis and Interpretation of Range Images* (Springer-Verlag, 1990), *Algorithms For Clustering Data* (Prentice-Hall, 1988), and *Real-Time Object Measurement and Classification* (Springer-Verlag, 1988).

Dr. Jain is a Fellow of the Association for Computing Machinery (ACM) and the International Association of Pattern Recognition (IAPR). He was the Editor-in-Chief of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE between 1991 and 1994. He received awards for best papers in 1987 and 1991 and for outstanding contributions in 1976, 1979, 1992, 1997, and 1998 from the Pattern Recognition Society. He also received the 1996 IEEE TRANSACTIONS ON NEURAL NETWORKS Outstanding Paper Award. He has also received a Fulbright Research Award, a Guggenheim Fellowship, and the Alexander von Humboldt Research Award. He delivered the 2002 Pierre Devijver lecture sponsored by the IAPR.