CHAPTER

In-vehicle vision systems 23

This chapter considers the value of in-vehicle vision as part of the means for providing driver assistance systems. To achieve this, many objects have to be identified, including not only the roadway itself but also the lane and other markings on it, road signs, other vehicles, and pedestrians. The latter are particularly important as their actions are relatively unpredictable, and people who wander into the roadway are liable to cause accidents—unless the driver assistance system can help to avoid them.

Designing in-vehicle vision systems is anything but trivial, as they necessarily deploy moving cameras, which means that all objects in a scene are moving; hence, it becomes quite difficult to eliminate the background from consideration. For these reasons, it becomes necessary to rely more on recognition of individual objects than on motion-based segmentation.

Look out for:

- how the roadway, road signs, and road markings may be located
- the availability of several distinct methods for locating vehicles
- what information can be obtained by viewing license plates and wheels
- how pedestrians may be located
- how vanishing points (VPs) can be used to provide a basic understanding of the scene
- how the ground plane may be identified
- how a plan view of the ground plane can be obtained and used to help with navigation
- how vehicles can be guided using vision to compensate for roll, pitch and yaw.

While it is easy to set out strategies for building in-vehicle vision systems that will work well in normal conditions on the roadway, it is far from simple to design them to operate on the less structured environments of farms or fields. Indeed, much additional reliance on GPS (global positioning system) and other methodologies will often be needed for the purpose.

23.1 INTRODUCTION

This chapter provides an introduction to in-vehicle vision systems. The topic clearly overlaps with many of the ideas of the previous chapter, particularly regarding traffic surveillance, as here we are regarding the flow from inside a vehicle rather than from a stationary camera mounted (typically) on an overhead gantry. However, although the environment may be similar, the situation is

Computer Vision. DOI: http://dx.doi.org/10.1016/B978-0-12-809284-2.00023-X

essentially different, because the camera platform is in motion and almost nothing that is viewed appears stationary (Table 23.1). This means that it is extremely difficult to use methods such as background subtraction. Note that while it is theoretically possible to find a general perspective transformation that makes a sequence of frames exactly coincide so that background subtraction can be achieved, to do this would be to replace a technique that is intended to be a simple way of cueing into images into one that is, highly complex, and the process of finding a sufficiently exact perspective transformation would itself require considerable computation, so this is unlikely to provide a useful strategy for analyzing image sequences.

Given the more difficult problem of analyzing scenes containing moving objects from a moving platform, we have to find ways of tackling the task equitably. Fortunately, with vehicles on a road, the range of types of scene is highly restricted. In particular, the roadway is always present in the image foreground and thus is easily identifiable. Likewise, it normally has a characteristic dark intensity, and thus its recognition right into the distance need not be too problematic: the fact that it is moving relative to the camera is relatively immaterial. In fact, it may even be quite difficult to detect motion by looking downward toward the road surface. Next, there are a whole host of standard types of object that are likely to be visible from within a vehicle—buildings, other vehicles, pedestrians, road markings, road signs, telegraph poles, lamp standards, bollards, and so on. The high frequency with which each of these can appear indicates that it will be necessary to have the capability of recognizing each of them independently, at any range and at any speed. This means that it is better, *as a first stage in the analysis*, to revert to ignoring speed of motion and to concentrate on pattern recognition. In fact, recognition can be helped by considering the range, which is readily deduced, approximately at first, from the lowest location on the object, which is where it meets the road (it is here assumed that the road has already been segmented from the remainder of the scene as an important preliminary stage of the analysis). Note that depending on the aim of the analysis (a point to which we shall return below), it is likely to be more important to identify objects that lie within the road region, so segmentation of the latter is all the more important as a first stage. Then, these objects—now restricted mainly to the subset, other vehicles, pedestrians, road markings, road signs, traffic lights—each needs to be identified in its own right. Later, the exact motion of the moving platform, and subsequently its location relative to all the other objects, will need to be ascertained.

Table 23.1 Levels of Difficulty When Motions Can Occur

1. Locating stationary objects from a stationary platform
2. Locating moving objects from a stationary platform
3. Locating stationary objects from a moving platform
4. Locating moving objects from a moving platform

We next consider the aims of implementing in-vehicle vision systems. Broadly, there are two: (1) navigation along the road, including staying in lane and finding out from road signs and traffic signals where to go, when to stop and other such information (here, to simplify matters, we ignore use of GPS and other types of help, and how information from the various sources can be fused together reliably); (2) driver assistance, which can include a variety of matters, particularly informing the driver of all aspects included in (1), and alerting him or her to important factors, such as vehicles that are braking, or pedestrians who are moving onto the roadway. In fact, much of the information that is acquired by the vision system will need to be conveyed to the driver in one way or another. However, of particular interest is the fact that drivers will sometimes not be able to act rapidly enough to avoid pedestrians, vehicles that brake unpredictably, overtaking vehicles that suddenly cut in, and so on. There is also the problem that drivers may be drowsy or may for various reasons—e.g., because of distractions from other occupants or those caused by the simultaneous need to navigate—react too slowly, so that an accident could become imminent. In such cases, driver assistance that could automatically initiate breaking or swerving might be crucial. We can also envisage various situations where the vision system would be part of a fully automatic driving system: here, there is bound to be a problem of legality, and who or what would be to blame for an accident (viz. the driver, car manufacturer, vision system designer, or whoever). We shall not delve into such problems here, but just consider the vision system as an enabling technology. However, once vision and driver assistance systems become sufficiently powerful, they will doubtlessly become part of other schemes such as those for driving in tight convoys—deemed by many to be the best way of achieving rapid safe transit along our motorways. In addition, there are other ways in which driver assistance can be valuable: these range from cruise control to automatic parking.

In this chapter, we focus generally on providing a vision system that can perceive all that might be needed for vehicle guidance and driver assistance, with emphasis on locating the roadway and road lanes, identifying other vehicles and locating pedestrians close to or on the roadway. As indicated above, the whole process starts by locating the roadway, as discussed in the following section.

23.2 LOCATING THE ROADWAY

Chapter 4, The Role of Thresholding described a technique that was capable of locating the roadway using a multilevel thresholding approach (see Fig. 4.9B). In fact, the roadway was identified by the third and fourth thresholds as that section of the image with gray levels in the approximate range 100−140. Similar results are obtained in other cases, e.g., Fig. 23.1B, where the two threshold values demarcate an even greater gray-scale range, c. 60−160. While these can be construed as being reasonably ideal cases, thresholding is such a basic technique that it should be

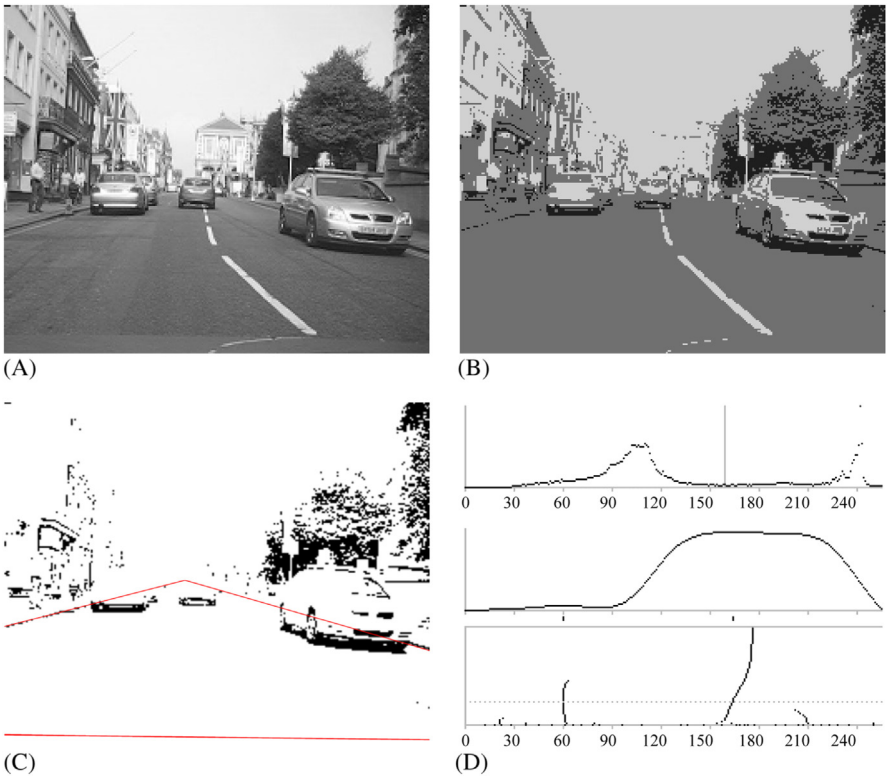

FIGURE 23.1

Frame of video taken from a moving vehicle. (A) Original image. (B) Doubly thresholded image. (C) Result of only applying the lower threshold. (D) Top: intensity histogram of the original scene. Middle: result of applying the global valley transformation and smoothing. Bottom: the dotted line shows the two thresholds used in (B) being located automatically. For further details, see text in Chapter 4, The Role of Thresholding. The red lines in (C) demonstrate that, within the road area, the lower threshold predominantly identifies under-vehicle shadows.

possible to extend it to cover less ideal situations. For example, if shadows appear on the roadway, the latter would in many cases appear as two contiguous sets of regions with two prominent intensity levels and could indeed be identified by the same method. Note also that varying illumination levels would be likely to make one intensity elide smoothly into another, and if a suitable range of intensities between thresholds (as in Figs. 4.9 and 23.1) were taken into account, the segmentation problem might still be solved in exactly the same way. However, ultimately the problem is one of pattern recognition and can be solved by (1) eliminating other

objects, such as road lane markings, (2) identifying the limits to the roadway, and (3) taking other features such as color or texture into account. Note that as the color of the roadway is often a bland gray, it may only be made to stand out by noting the colors of the other surroundings, such as grass, trees, or brickwork on buildings. Clearly, this would make the whole system more complex, but in a well-known and well-worn way—pattern recognition by now being a reasonably mature subject. To some extent, the situation may be helped by bringing the motion of the vehicle into the picture (we have so far resisted this, to bring the discussion to the simplest possible base level). In that case, without calculating the exact motion of the vehicle, we can take account of the fact that the roadway stretches for a long distance ahead, so any part of it that is established to be roadway will remain so until the vehicle passes over it. Furthermore, on the road ahead, any vehicle that is located evidently runs on the roadway, so parts of it are continuously being identified. Thus, the camera vehicle merely needs to keep a record of all candidate regions that have been positively identified, so that any ambiguities from identification via intensities can be eliminated. Finally, this time taking motion parameters into account, keeping a tally on the road boundaries with the aid of Kalman filters will solve many of the remaining issues.

23.3 LOCATION OF ROAD MARKINGS

It will have been noticed from Figs. 4.9 and 23.1 that the multilevel thresholding technique used to locate the gray surface of the road simultaneously segments white road markings. However, white road markings are seldom pure white and may be worn or even partly duplicated by older markings. In any case, segmenting them by thresholding is not the same as absolute identification. One way around this dilemma is that of fitting the road markings to suitable models. Often straight lines are adequate, though sometimes parabolas have been used for the purpose. Fig. 23.2 shows a case where continuous and broken road markings have been identified using the RANSAC (random sample consensus) technique, which helps to locate the VP on the horizon to a reasonable approximation. The widths of the road lane markings can also be measured in this way. Fig. 23.3 takes this even further. In this case, a greater degree of reliability and accuracy is obtained by locally bisecting each lane marking horizontally before feeding the data to RANSAC. In this way, extraneous signals can be eliminated—if necessary by filtering the horizontal widths. Note how RANSAC is able to find the best fit straight line section even when the road lane markings are curved. Likewise, it is able to eliminate lane markings that have been distorted by the presence of older lane markings (Fig. 23.3A). As described in Chapter 10, Line, Circle, and Ellipse Detection, the version of RANSAC used for the tests successively eliminates the data points used to fit line segments, and the width delete threshold d_d is made larger than the fit threshold d_f so that no data points are retained that could

FIGURE 23.2

Application of RANSAC for locating road lane markings. (A) Original image of road scene with lane markings identified by RANSAC. (B) The edge point local maxima used by RANSAC for locating the road lane markings. While the lane markings converge to approximately the right point on the horizon line, the parallel sides of the individual lane markings do not converge quite so accurately, indicating the limits achievable with so few edge points. This is more a failure of the edge detector than of RANSAC itself.

mislead the algorithm while searching for subsequent line segments (see the algorithm flowchart in Fig. 23.4).

23.4 LOCATION OF ROAD SIGNS

We now continue with the process of analyzing the vehicle's environment and consider the most relevant remaining stationary parts that lie on or adjacent to the roadway. These include the traffic signs. It will not be possible to examine more than one or two cases, but amongst these are various relevant warnings, including those for road bumps and "GIVE WAY": note that many others appear in the same style—with the message in black on a white background and enclosed in a red triangle. To locate these signs, some tests were made without using the color aspect as this might represent too easy an approach (note also that in the wrong lighting conditions, color can be misleading): instead, an idealized small binary template of size 22×19 pixels was employed. While apparently crude, this small template had the advantage of requiring very little computation to locate the relevant objects. In fact, the chamfer-matching technique (Section 22.7) was used for detecting the traffic signs shown in Fig. 23.5. While the template was primarily designed to detect the road bump sign, it also gave a sizeable signal for the GIVE WAY sign. Indeed, the two signals found using the template were both well above the signal-to-noise ratio elsewhere in the image, the closest possible false alarms being high up in the trees, which contain a plethora of random shapes. Note that the picture was taken under highly nonideal conditions on a wet day when there were a number of reflective areas on the road. Overall, the chamfer-

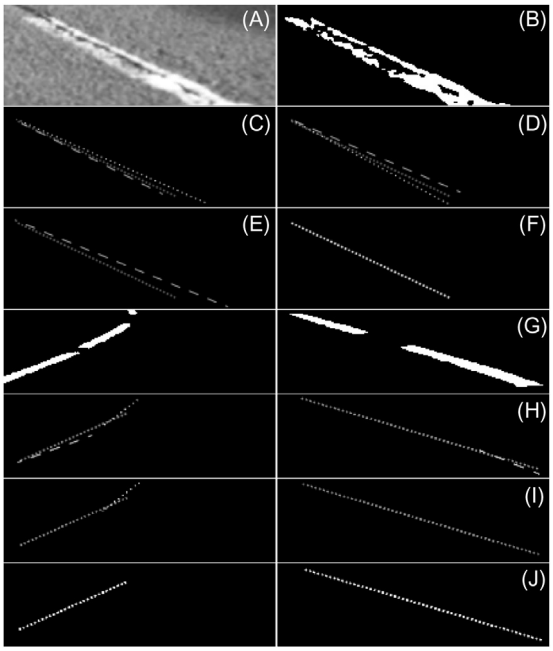

FIGURE 23.3

Further tests of RANSAC for locating road lane markings. (A) Original Image 1: a distorted set of double line road markings. (B) Thresholded version of (A). (C) 3–3. (D) 3–6. (E) 3–10. (F) 3–11. (The notation "d_f–d_d" means that d_f is the "fit distance," and d_d is the "delete distance:" see text.) (G) Original Image 2, already thresholded: the central section of the road containing no markings has been eliminated to save space. (H) 3–3, (I) 3–6, (J) 3–11. Parts (F) and (J) show the final results as dense dotted lines: in other cases, dots and dashes are used to distinguish the different lines. Note that immediately after thresholding, the horizontal bisector algorithm finds the midpoints of white regions along horizontal lines and feeds them to RANSAC for fitting.

matching technique seems well suited to rapidly locating fixed road signs of various sorts.

There is some possibility of designing a single idealized template for locating all triangular signs. Note first that a blank white interior would be more suitable than the road bump structure in Fig. 23.5E: this corresponds to disregarding the center of the template, taking it as being composed of "don't care" locations. In fact, the template should really be designed by a suitable training approach such as the one outlined by Davies (1992d). In this method, a matched filter approach is used in designing templates, with local variability of training samples (represented by standard deviation $\sigma(\mathbf{x})$) being taken to correspond to noise, thereby necessitating reduced local weighting: the local matched filter

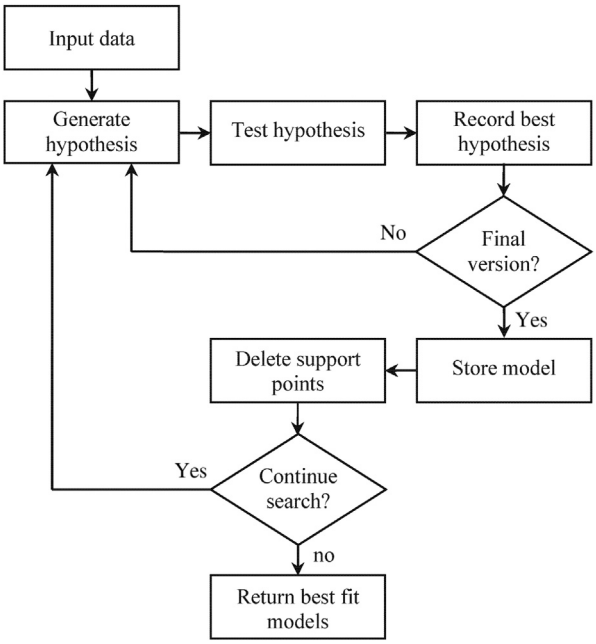

FIGURE 23.4

Flowchart of the lane detector algorithm used for the tests in Fig. 23.3.

weighting is thus (Davies, 1992d) taken as $\overline{S}(\mathbf{x})/\sigma(\mathbf{x})^2$ rather than $\overline{S}(\mathbf{x})$, where $\overline{S}(\mathbf{x})$ is the mean local signal at $\mathbf{x}$ during training. For the types of road sign considered above, variable distributions of black within the central white area would be treated optimally by this method.

23.5 LOCATION OF VEHICLES

In recent years, a number of algorithms have been designed for locating vehicles on the road, whether in surveillance applications or by in-vehicle vision systems. One notable means for achieving this has been by looking for the shadows induced by vehicles (Tzomakas and von Seelen, 1998; Lee and Park, 2006). Importantly, the strongest shadows are those appearing beneath the vehicle, not least because these are present even when the sky is overcast and no other shadows are visible. Such shadows are again identified by the multilevel thresholding approach of Chapter 4, The Role of Thresholding. Fig. 23.1 shows a particular instance of this, where almost the only dark pixels appearing within the roadway region are the under-vehicle shadows. In fact, as under-vehicle shadows lie under

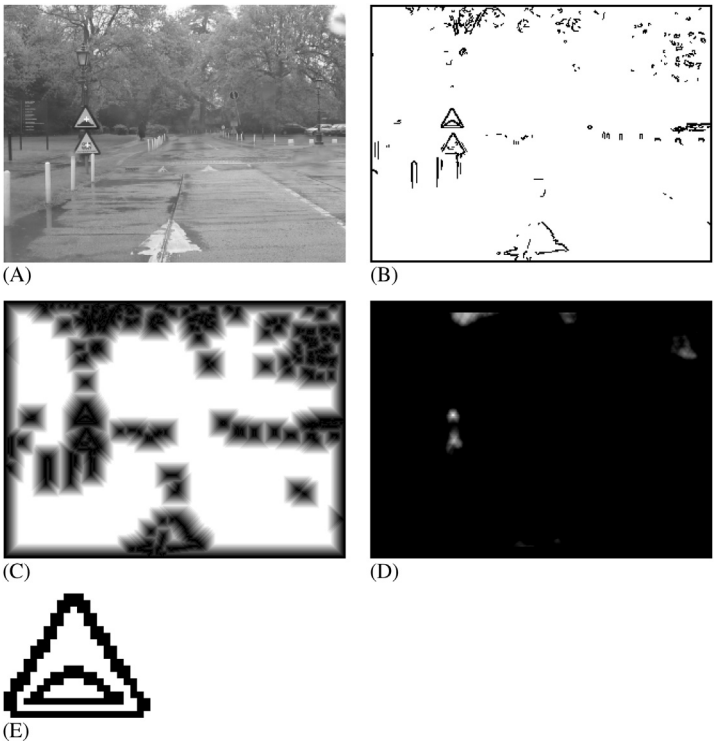

FIGURE 23.5

Locating road signs using chamfer matching. (A) Original image showing two triangular road signs (indicating a road bump and "GIVE WAY"): each of the signs is marked with a white cross where it has been located by the chamfer-matching algorithm. (B) Thresholded edge image after nonmaximum suppression. (C) Distance function image: note that with the display enhancement factor of 20 used here, the distance appears to saturate at 13 pixels. (D) The response obtained when moving the template (E) over the image.

vehicles, an excellent way of locating nearby vehicles is to move upward from the lowest part of the roadway until a dark entity appears: there is then a high probability that it will only locate vehicles. Note that in Fig. 23.1C, the other main candidates are trees, but these are discounted as being well above the road region—as indicated by the dotted triangle.

As pointed out earlier when considering methods for locating the road region, it is useful to have a number of methods available for locating objects such as vehicles, in case of peculiar illumination conditions or other factors. Following this line of analysis, we consider symmetry, which was first used for this purpose some years ago (e.g., Kuehnle, 1991; Zielke et al., 1993). Fig. 23.6 shows a

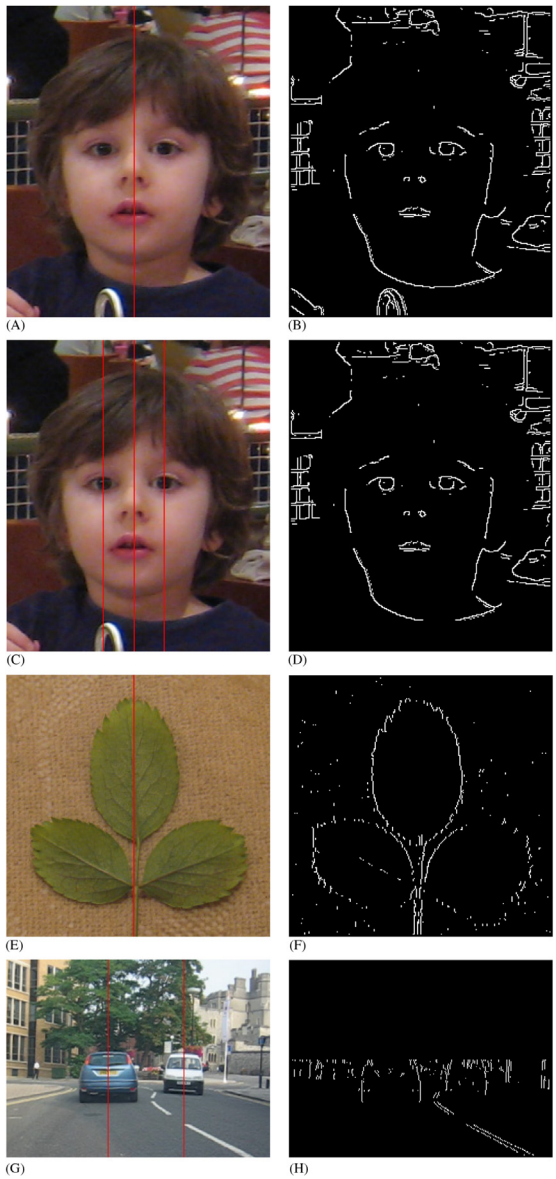

FIGURE 23.6

Searching for symmetry in images. (A) Original image of a face with a vertical axis of symmetry. (B) Edge image used for determining the axis of symmetry in (A). (C) Original image with symmetry axes of the eyes. (D) Slightly restricted edge image used for determining the symmetry axes of the eyes. (E) Original image of a leaf triplet, with symmetry axis. (F) Vertical edge image used to determine the symmetry axis in (E). (G) Original image of a traffic scene, with symmetry axes marked. (H) Vertical edge image used for determining the symmetry axes in (F). The slight bias of the left-most symmetry axis in (C) is not surprising in view of the few pixels involved and the interfering effects of other edge pixels in the image.

number of trials in which symmetry is applied to locate objects exhibiting a vertical axis of symmetry. The approach used is the 1-D Hough transform (HT), taking the form of a histogram in which the bisector positions from pairs of edge points along horizontal lines through the image are accumulated. When applied to face detection, the technique is so sensitive that it will locate not only the center-lines of faces but also those of the eyes. In the case of Fig. 23.6C, the algorithm was confused by the metal object at the bottom when locating the eye on the left, but when tested without that present, it was found without difficulty. Note that some bias occurs there because the algorithm is averaging the contribution of the whole eye, and the displacement between the iris and the rest of the eye becomes important. Similarly, the set of leaves in Fig. 23.6E is located without trouble, but the exact vertical axis that is located represents the combined peak signal from the lower two leaves and the uppermost leaf: in such a case, it would be better to identify each one separately. These sorts of problem are less important in Fig. 23.6G where both vehicles are located quite accurately—in spite of the fact that the car on the right is not exactly horizontal. Interestingly, both vehicles would also be found using the under-vehicle shadow method. The fact that they both lie within their respective lanes also aids positive identification.

In spite of these successful applications of symmetry, note that the approach needs to be used with caution. In particular, the building on the left in Fig. 23.6G gives a plethora of signals because of the multiple symmetries between its windows. An interesting lesson is that three equally spaced vertical lines at locations $x = 1, 3, 5$ will have a symmetry not only at $x = 3$ but also at $x = 2$ and 4.

Finally, rotation symmetries and reflection symmetries about nonvertical axes are not especially useful in the present context. However, just as a 1-D HT can be used to locate symmetries about vertical axes, so 2-D HTs can be used to locate symmetries about lines of arbitrary direction. Thus, one can build a single 2-D parameter space, each horizontal line of which represents the symmetry in a different direction in the image. Such a parameter space might be expected to have a minor amount of coherence in the vertical direction, but we do not consider this further here.

23.6 INFORMATION OBTAINED BY VIEWING LICENSE PLATES AND OTHER STRUCTURAL FEATURES

License plate location has already been covered in Section 22.10. In this section, we consider what can be deduced from an oblique view of a license plate of length R. We simplify the situation by assuming that both the image plane and the license plate are vertical and that they have their main axes aligned horizontally and vertically. Fig. 23.7A and B shows respectively the oblique and plan views of the license plate horizontal axis. The apparent horizontal projection (CQ) of the center-line of the license plate is $R \cos \alpha$ when viewed in the direction PT. Following Fig. 23.7C, its vertical projection (QT) is $R \sin \alpha \tan \beta$.

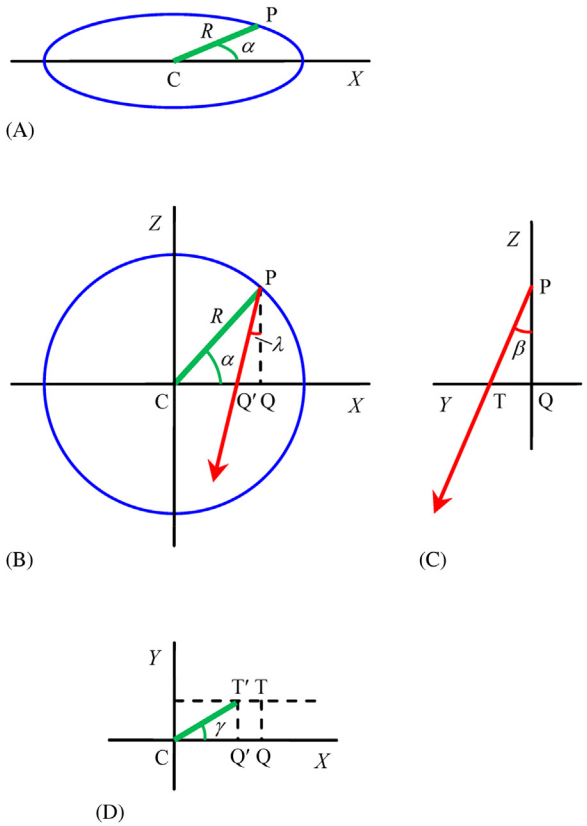

FIGURE 23.7

Horizontal line pose viewing geometry. (A) Oblique view of a horizontal straight line of length R rotated through an angle α from the X-axis. (B) Plan view of the line. (C) Side view showing the viewing direction, along PT′, with a lateral angle λ; the angle of elevation β is that of T, not of T′. (D) Front view in the X–Y plane, which is parallel to the image plane x–y. Note that the horizontal line CP in (B) appears to lie at an angle γ in (D): it has an apparent length (CT′) of R'.

However, when viewed in the more general direction PT′, with lateral angle λ, its horizontal projection is CQ′, which is equal to $R\cos\alpha - R\sin\alpha\tan\lambda$. From Fig. 23.7D, we deduce that its apparent angle γ and length R' are given by the equations:

$$\tan\gamma = \frac{\tan\alpha\tan\beta}{1-\tan\alpha\tan\lambda} \tag{23.1}$$

$$\begin{aligned} R' &= R\cos\alpha(1-\tan\alpha\tan\lambda)\sec\gamma \\ &= R\sin\alpha\tan\beta\,\mathrm{cosec}\,\gamma \end{aligned} \tag{23.2}$$

These formulae seem intuitively correct, as for example, $\gamma = 0$ if $\alpha = 0$ or $\beta = 0$. In addition, under nonoblique viewing, $\beta = 0$, $\lambda = 0$, and $\gamma = 0$, so Eq. (23.2) reverts to the standard result for nonoblique viewing, $R' = R \cos \alpha$.

Perhaps a more important case is that of $\alpha = \pi/2$, leading to $\tan \gamma = -\tan \beta / \tan \lambda$. We can interpret this result by taking image plane coordinates (x, y) and 3-D coordinates (X, Y, Z). Noting that $\tan \beta = y/f$ and $\tan \lambda = x/f$, we deduce that $\tan \gamma = -y/x = -Y/X$. This corresponds to viewing perspective lines on the roadway that are parallel to the optical axis of the camera. (Note that the minus sign in these equations corresponds to the fact that γ will be viewed in the range $\pi/2$ to π when $\alpha = \pi/2$.)

Finally, note that instead of obtaining the projection of the line as it would appear in the direction of viewing, we have determined its projection in the vertical plane $X-Y$, which is parallel to the image plane $x-y$. As a result, the equations correspond *exactly* to projective projection into the image plane, rather than merely to orthographic projection.

We now need to obtain an equation for α in terms of the other parameters. Solving Eq. (23.1) for α, we find:

$$\tan \alpha = \frac{\tan \gamma}{\tan \beta + \tan \gamma \tan \lambda} \tag{23.3}$$

Next, taking the projections of the center-line of the license plate along the image x and y axes to be $\delta x, \delta y$, we find that the parameters β, γ, λ are all measurable, so α can be estimated:

$$\tan \alpha = \frac{\delta y / \delta x}{(y/f) + (\delta y / \delta x)(x/f)} = \frac{f \delta y}{y \delta x + x \delta y} \tag{23.4}$$

Thus, we now know the orientation in space of the license plate. In principle, we can use Eq. (23.2) to estimate the range of the license plate. To achieve this, we need to know the value of R. In fact, for standard UK license plates, R is reasonably well defined (this assumes that the number of characters in the license plate is known), and so Eq. (23.2) can be used to estimate R'. Next, the ratio of R' to the apparent length r of the license plate gives the range Z:

$$Z = fR'/r = \frac{fR'}{[(\delta x)^2 + (\delta y)^2]^{1/2}} \tag{23.5}$$

If we had also made use of the apparent lengths and orientations of the shorter sides of the license plate, we could have eliminated dependence on the assumptions that the latter are vertical. However, it is unlikely that these short lines could be measured accurately enough to improve the situation significantly: instead we presume that the best that can be done is to use measurements on the longer sides to obtain preliminary estimates of the positions of vehicles, which can then be improved by other measurements.

Unfortunately, all the above theory is somewhat confounded by the variable camber of the road. But note that, while the camber will be considerably different on the opposite side of the road, its effects will tend to cancel when observing the

FIGURE 23.8

Vehicles viewed obliquely. More accurate information about orientation is often obtained from the side of the vehicle than from its rear.

license plates of vehicles on the same side of the road. Next, the size of γ depends on y, and hence on the height of the camera above the target feature: this means that the observed value of γ will be smaller for the license plate than for the rear wheels; hence, if the rear wheels are not occluded, it is likely that they will give a more accurate estimate of α than that from the license plate. Nevertheless, license plates are more satisfactory indicators than rear wheels both because they are less likely to be occluded and because they are uniquely recognizable: in fact, the rear wheels of one vehicle can sometimes be confused with those of other vehicles, and even the front wheels can cause confusion. Finally, another factor needs to be borne in mind—that we are attempting to estimate an often small quantity α from another small quantity γ when both are comparable to the interfering effect of the camber angle. Interestingly, this problem can be overcome more effectively by estimating $\tilde{\alpha} = \pi/2 - \alpha$ from $\tilde{\gamma} = \pi/2 - \gamma$ and applying these measures to views of the sides (particularly the sides of the wheels) of other vehicles. All this can be achieved by recalling that tan α and tan γ should, respectively, be replaced by cot $\tilde{\alpha}$ and cot $\tilde{\gamma}$ in Eqs. (23.1) and (23.2). Overall, it might be expected that side views of vehicles will be more valuable for estimating orientation than rear views, whether the latter use rear wheels or license plates as indicators (though, obviously, only the rear view of a vehicle will be relevant when driving directly behind it). Consideration of Figs. 23.1, 23.6, 23.8, and 23.9 will provide adequate confirmation of these observations.

Finally, it might be asked why so much emphasis has been placed on measurement of angles vis-à-vis distances. This is basically because angles represent ratios of distances, and thus they tend to provide scale-invariant information. In addition, they do not demand knowledge of absolute distances for interpretation.

23.7 LOCATING PEDESTRIANS

In principle, locating whole pedestrians would require many chamfer templates of varying shapes and sizes, to cover the many body profiles of moving people. The

FIGURE 23.9

Chamfer matching to locate pedestrians from their lower legs. Parts (A) and (B) show original images of road scenes containing pedestrians. The red dots are the peak signals after chamfer matching using an idealized binary U template. Note the plethora of false positives because of the number of vertical edges able to stimulate signals—as seen in (C) and (D).

alternative chosen here is to look for specific subshapes that would be more general and invariant. Possibilities include leg, arm, head, and body sections. Fig. 23.9 shows lower legs being located using an idealized "U" template with parallel sides. However, a plethora of false positives arises because of the large number of vertical edges that are able to stimulate signals. Their presence means that the distance functions do not have the ideal maximum values that might be expected because the spurious edges reset the distance functions to zero in many places. This does not affect the sensitivity of the method in the sense that the templates are bound to locate instances of the profiles they represent. However, it does affect the numbers of false positives that are detected. In fact, in the examples shown, the result is not disastrous, because the lowest objects found, once road markings are eliminated, are the feet of the pedestrians. However, the fact that the method does not give ideal results makes it essential to back it up using alternative methods.

The Harris operator provides a useful alternative approach. As Fig. 23.10 shows, it is able to locate a range of features, including feet and heads, as well as road lane markings. Note that in the case shown in Fig. 23.10A, the right foot has not been found as it is larger than the other foot, and the particular Harris operator

FIGURE 23.10

Alternative approach to pedestrian location using the Harris operator. Here, the operator has the effect of locating corners and interest points, some of which include pedestrian feet and heads: above all, road lane markings are also located with high probability. The operator has not been tuned in any way to recognize such features. In addition, it has no sense of polarity (preference for dark or light).

employed stretched over a range of only seven pixels. Note that the Harris operator has no sense of polarity (preference for black or white): in the case of pedestrians, this is useful as the clothing and shoes (or feet) are unpredictable and can appear dark on a light background or vice versa. (Lack of polarity also applies to chamfer matching, but for different reasons.)

Further approaches are useful to back up the two mentioned above and also to confirm detections that have already been made. In this respect, unique identification of human skin color can be useful. That this is possible is shown in Fig. 23.11, one of the main problems clearly being the rather small numbers of pixels in the face regions. To carry out skin detection rigorously, it is necessary to train the color classifier on a set of training images. This was carried out for Fig. 23.11E. While the method was highly successful (see Fig. 23.11F), it corresponded to supervised learning of skin color; in practice, with less tight control of the training images, this process could be compromised by the presence of sand, stone, cement, and a host of brown variants, which have colors close to those of darker or lighter people. Another important factor is that in-vehicle vision systems will not have sufficient time to gather enough training data, considering particularly that the whole point of a vehicle is to travel and thus adaptation from dark to light and other environmental factors are bound to be a source of serious problems. In this respect, in-vehicle systems are subject to far worse conditions than will be usual for surveillance systems.

Overall, we find that in-vehicle pedestrian detection systems involve a demanding set of pattern recognition problems. Earlier we emphasized the potential value of pattern recognition when moving objects are being detected from moving platforms: this approach to the subject was also useful for didactic reasons. However, we are now finding that there are limits to this. In fact, it would

FIGURE 23.11

Another approach to pedestrian location via skin color detection. Parts (A) and (B) show that a lot can be achieved via skin color detection, detecting not only faces but also neck, chest, arms, and feet: see also the detail in (C) and (D). With proper color classifier training, even more can be achieved, as shown in (E) and (F).

be an artificial restriction not to make use of motion by at least tracking features and grouping them according to velocity (a process that was already mentioned in Chapter 22: Surveillance). The problem with this approach is the large number of, for example, interest point features that exist in an entire image, where almost all the features are moving. If each of them (say N) is to be compared with all others in a pair of adjacent frames, then $O(N^2)$ operations will have to be undertaken. However, by acknowledging the individuality and different characteristics of the various features, and their spatial arrangements, this vast number can be cut down

(E)

(F)

FIGURE 23.11

(*Continued*).

to manageable proportions. In particular, feature points should only move a limited distance between frames, so there will only be a small number n of candidates that match a given feature as it moves from one frame to the next. This leaves us with O(Nn) pairs of feature points to consider, a number that can be further minimized by examining the relative strengths and colors of the various pairs (ideally, the final result will be O(N)). Here, some of the ideas of Section 6.7, where features were characterized by a great many descriptors, may prove useful, even though wide baseline matching is not relevant for frame to frame tracking.

23.8 GUIDANCE AND EGOMOTION

An important aspect of driver assistance systems is that of vehicle guidance. In fact, this aspect is important both for vehicles with human drivers and for autonomous robot vehicles. In either case, vehicle egomotion is handled by a controlling computer which has to be fully aware of the situation. Incoming images contain complex information and reliable cues have to be found to key into them. Amongst the most widely used such cues are VPs, which are often very evident in city scenes (e.g., Fig. 17.11).

One of the ways in which VPs are most useful is in helping to identify the ground plane, and a lot of other information follows from this. In particular, local scale can be deduced: for example, objects on the ground plane have width that is referable to, and a known fraction of, the local width of the ground plane; in addition, VPs permit an estimate to be made of distance along the ground plane, by measuring the distance from the relevant image point to the VP, as we shall see below. Thus, they are useful for initiating the process of recognizing and measuring objects, determining their positions and orientations, and helping with the task of navigation.

Here, a lot will depend on the type of environment and the type of vehicle. There are many possibilities such as vacuum-cleaning robots, window-cleaning robots, lawn-mowing robots, invalid chair robots, weeding and spraying robots, maze-running robots, not to mention vehicles running autonomously on roads, or cars that park themselves automatically. In some cases, robots will have to undertake mapping, path planning, and navigational modeling and engage in detailed high-level analysis: this sort of situation has been explored by Kortenkamp et al. (1998). This approach will be important if a path has obstacles such as bollards or pillars (Fig. 23.12), and it will be vital for a maze-running robot. In many such cases, vision or other sensors will provide only limited information about the working area, and knowledge will have to be augmented in a suitable representation: this makes a plan view model of the working area a natural solution. To proceed with this idea, we need to transfer the information from individual images into the plan view representation (see the algorithm of Table 23.2).

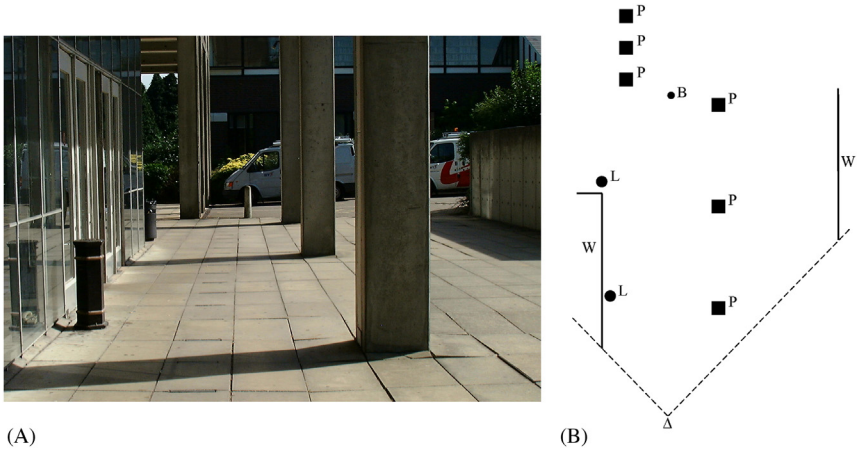

FIGURE 23.12

Plan view obtained for navigation. (A) View of a scene showing the obstacles to be avoided. (B) Plan view of the ground plane showing what is visible from viewpoint Δ (for clarity, the full areas of the pillars P, bollards B, and litter-bins L are shown). The walls are marked W.

Table 23.2 Computing Ongoing Plan Views of the Ground Plane

1. Detect all edges in the current frame
2. Locate all straight lines in the current frame: e.g., use an Hough transform
3. Locate all VPs: use a further HT, as described in Section 18.7
4. Find the VP closest to the direction of motion: eliminate all other VPs
5. Determine the closest section of G: this should be the part of the frame immediately in front of the robot
6. Use this and other information to determine which lines through the primary VP lie on G: eliminate all other lines
7. Segment objects on G
8. Eliminate object boundaries on G that are unrelated to lines passing though the primary VP
9. Tentatively identify as shadows any dark regions lying on G
10. Take the remaining object and shadow boundaries and check for consistency between frames: e.g., use the 5-point cross ratio values, as described in Section 18.3
11. Label all remaining feature points on G with their (X, Z) coordinates: use Eqs. (23.7) and (23.8)
12. Check for consistency with previous frames
13. Update list of objects with inconsistent boundaries as not lying on G, or as being otherwise unreliable: these could be due to moving shadows or noise
14. Update history of feature point coordinates on G

This table presents an algorithm showing how a plan view of the ground plane G may be computed. It is assumed that the robot sees a sequence of video frames, and that it has to update its knowledge base as each frame comes along. The algorithm is set up assuming that it is best to analyze each frame ab initio, and then to look for consistency with previous frames.

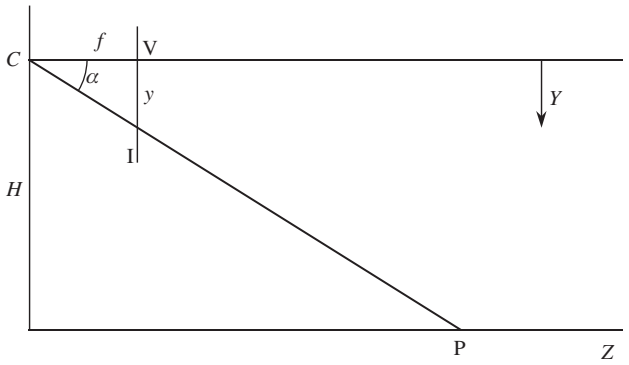

FIGURE 23.13

Geometry relating the image and the ground plane. C is the center of projection of the camera, I is the image plane, V is the vanishing point, and P is a general point on the ground plane. f is the focal length of the camera lens, and H is the height of C above the ground plane. The optical axis of the camera is assumed to be parallel to the ground plane.

Basically, to construct a plan view of the ground plane, we start with a single view of a scene in which the VP V has been determined and significant feature points on the ground plane (particularly regarding its boundaries) have been identified. Next, distance along the ground plane can be deduced as shown in Fig. 23.13. The angle of declination α of a general feature point P (X, H, Z) on the ground plane, seen in the image as point (x, y), is given by:

$$\tan \alpha = H/Z = y/f \tag{23.6}$$

The value of Z is therefore given by:

$$Z = Hf/y \tag{23.7}$$

After obtaining a similar formula giving the lateral distance X, we deduce that:

$$X = Hx/y \tag{23.8}$$

The world (plan view) coordinates (X, Z) have now been found in terms of the image coordinates (x, y). Note that y has to be measured from the VP V rather than the top of the image. Note also that as X and Z vary inversely with y, they vary rapidly when y is small, so digitization and other errors will markedly affect the accuracy with which far-away objects can be located from the plan view.

When the optical axis of the camera is not parallel to the ground plane, the calculations are best dealt with using homogeneous coordinates as shown in Chapter 19, Image Transformations and Camera Calibration.

23.8.1 A SIMPLE PATH-PLANNING ALGORITHM

In this subsection, we assume that a plan view of the environment has been built up using the methods of the previous section. While it is by no means clear that humans use an instantaneous plan view model to help them to walk or drive around an environment (an image-based representation seems more likely), it is clear that they use plan views for deductive, logical analysis of the situation and when reading maps. In any case, plan views probably constitute the most natural means for storing navigational knowledge and arriving at globally optimal routes. Here, we leave aside conjecture of exactly how humans juggle the information between the two representations and concentrate on how a robot might reasonably undertake path planning using a plan view it has built up. In fact, a maze-running robot would need to be provided with a suitable algorithm for this purpose.

Fig. 23.14A shows a simple maze in which the robot has to proceed from the entrance E to the final goal G (respectively, marked "↓" and "☺" in the figure). We assume that a plan view of the maze has been built up and that a systematic means is needed to find the optimum path to the goal G. The envisaged algorithm starts from G and propagates a distance function over the whole region, constrained only by the walls of the maze (Fig. 23.14). If a parallel algorithm is used, it is terminated when the distance function arrives at E; if a sequential algorithm is used, it must carry on until the whole maze has been covered—assuming that an optimal path is required. When the distance function has been completed, finding an optimum path necessitates proceeding downhill along the distance function until G is reached: at each point, the locally greatest gradient must be used (Kanesalingam et al., 1998). Connected components analysis could be used to confirm that a path exists, but a distance function has to be used to guarantee finding the shortest path. Note that the method will find only one of several paths of equal length: these arise because of the limitations of this type of method that assigns integer values to distances between adjacent pixels.

23.9 VEHICLE GUIDANCE IN AGRICULTURE

In recent years, there has been increasing pressure on farmers to reduce the quantities of chemicals used for crop protection. This cry has come both from environmentalists and from the consumers themselves. The solution to this problem lies in more selective spraying of crops. For example, it would be useful to have a machine which would recognize and spray weeds with herbicides, leaving the vegetable crops themselves unharmed: alternatively, the individual plants could be sprayed with pesticides. This case study relates to the design of a vehicle which is capable of tracking plant rows and selecting individual plants for spraying (Marchant and Brivot, 1995; Marchant, 1996; Brivot and Marchant, 1996; Sanchiz et al., 1996; Marchant et al., 1998). Interestingly, many of the details of

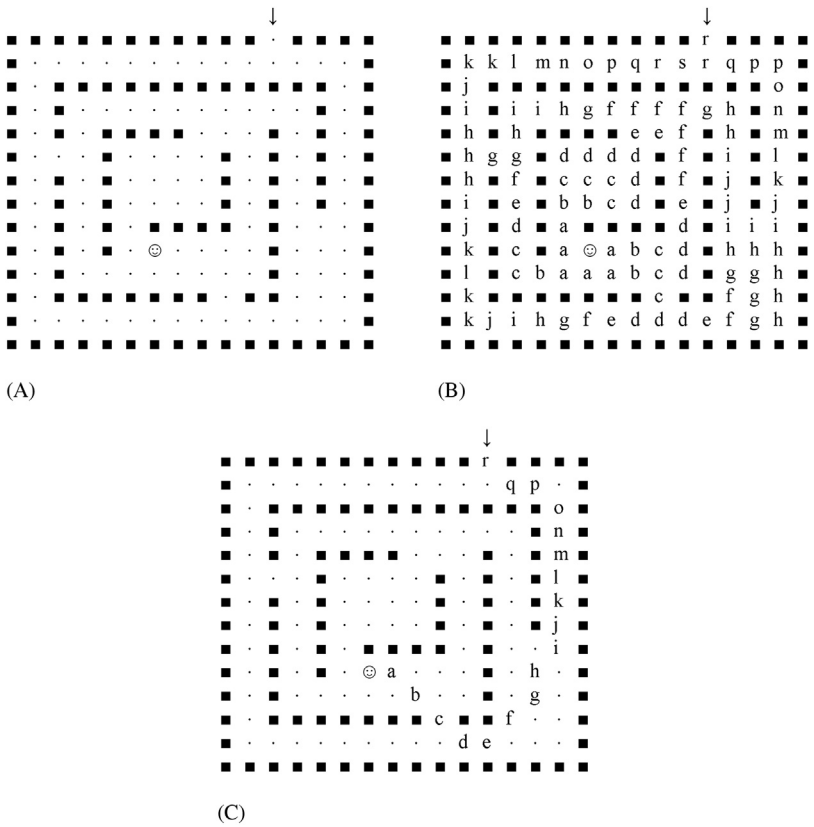

FIGURE 23.14

Method for finding an optimal path through a maze. (A) Plan view of maze. (B) Distance function of the maze, starting at the goal (marked ☺), and presenting distance values by successive letters, starting with $a = 1$. (C) Optimum path obtained by tracking from the maze entrance (marked ↓) along maximum gradient directions.

this work are remarkably similar to those for the totally independent project undertaken in Australia by Billingsley and Schoenfisch (1995).

The problem would be enormously simplified if plants grew in highly regular placement patterns, so that the machine could tell from their positions whether they were weeds or plants, and deal with them accordingly. However, the growth of biological systems is somewhat unpredictable and renders such a simplistic approach impracticable. Nevertheless, if plants are grown from seed in a greenhouse and transplanted to the field when they are approaching 100 mm high, they can be placed in straight parallel rows, which will be approximately retained as they grow to full size. There is then the hope (as in the case shown in Fig. 23.15)

FIGURE 23.15

Value of color in agricultural applications. In agricultural scenes, such as this, color helps with segmentation and with recognition. It may be crucial in discriminating between weeds and crops if selective robot weed killing is to be carried out.

that the straight rows can be extracted by relatively simple vision algorithms, and the plants themselves located and identified straightforwardly.

At this stage, the main problems are (1) the plants will have grown to one side or another and will thus be out of line; (2) some will have died; (3) weeds will have appeared near some of the plants; (4) some plants will have grown too slowly and will not be recognized as plants. Thus, a robust algorithm will be required to perform the initial search for the plant rows. The HT approach is well adapted to this type of situation: specifically, it is well suited to looking for line structure in images.

The first step in the process is to locate the plants. This can be achieved with reasonable accuracy by thresholding the input images (this process is eased if infrared wavelengths are used to enhance contrast). However, at this stage, the plant images become shapeless blobs or clumps (Fig. 23.16). These contain holes and lobes (the leaves, in the case of cabbages or cauliflowers), but a certain amount of tidying up can be achieved either by placing a bounding box around the object shape, or by performing a dilation of the shape which will regularize it and fill in the major concavities (the real-time solution employed the first of these methods). Then, the position of the center of mass of the shape is determined, and

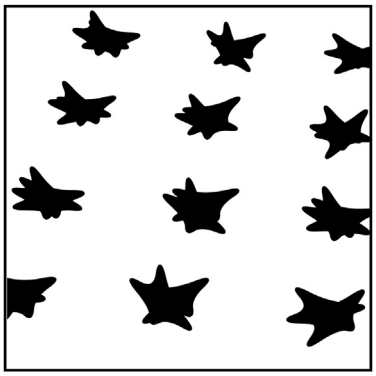

FIGURE 23.16

Perspective view of plant rows after thresholding. In this idealized sketch, no background clutter is shown.

it is this that is fed to the HT straight line (plant row) detector. In common with the usual HT approach, votes are accumulated in parameter space for all possible parameter combinations consistent with the input data. Here, this means taking all possible line gradients and intercepts for lines passing through a given plant center and accumulating them in parameter space. To help find the most meaningful solution, it is useful to accumulate values in proportion to the plant area. In addition, note that if three rows of plants appear in any image, it will not initially be known which plant is in which row, and therefore each plant should be allowed to vote for all the row positions: this will naturally only be possible if the inter-row spacing is known and can be assumed in the analysis. However, if this procedure is followed, the method will be far more resistant to missing plants and to weeds which are initially mistakenly assumed to be plants.

The algorithm is improved by preferentially eliminating weeds from the images before applying the HT. Weed elimination is achieved by three techniques—hysteresis thresholding, dilation, and blob size filtering. Dilation refers to the standard shape expansion technique described in Chapter 3, Image Filtering and Morphology and is used here to fill in the holes in the plant blobs. Filtering by blob area is reasonable as the weeds are seldom as strong as the plants, which were transplanted only when they had become well established.

Hysteresis thresholding is a widely used technique which involves use of two threshold levels. In this case, if the intensity is greater than the upper level t_u, the object is taken to be plant; if lower than the lower level t_l, it is taken to be weed; if at an intermediate level and next to a region classified as plant, it is taken to be plant; the plant region is allowed to extend sequentially as far as necessary, given only that there is a contiguous region of intensity between t_l and t_u connecting a given point to a true plant ($\geq t_u$) region. Note that this application is unusual in

that whole-object segmentation is achieved using hysteresis thresholding: more usually the technique is used to help create connected object boundaries (see Section 5.10).

Once the HT has been obtained, the parameter space has to be analyzed to find the most significant peak position. Normally, there will be no doubt as to the correct peak—even though the method of accumulation permits plants from adjacent rows to contribute to each peak. The reason for this is that with three rows each permitted to contribute to adjacent peaks, the resultant voting patterns in parameter space are as follows: 1,1,1,0,0; 0,1,1,1,0; 0,0,1,1,1—totaling 1,2,3,2,1—thereby making the true center position the most prominent (actually, the position is more complicated than this as several plants will be visible in each row, thus augmenting the central position further). However, the situation could be erroneous if any plants are missing. It is therefore useful to help the HT arrive at the true central position. This can be achieved by applying a Kalman filter (Section 20.8) to keep track of the previous central positions and anticipate where the next one will be—thereby eliminating false solutions. This concept is taken furthest in the paper by Sanchiz et al. (1996), where the individual plants are all identified on a reliable map of the crop field and errors from any random motions of the vehicle are systematically allowed for.

23.9.1 3-D ASPECTS OF THE TASK

So far we have assumed that we are looking at simple 2-D images which represent the true 3-D situation in detail. In practice, this is not so. The reason for this is that the rows of plants are being viewed obliquely and therefore appear as straight lines but with perspective distortions which shift and rotate their positions. The full position can only be worked out if the vehicle motions are kept in mind. In practice, vehicles moving along the rows of plants exhibit variations in speed and are subject to roll, pitch, and yaw. The first two of these motions correspond respectively to rotations about horizontal axes along and perpendicular to the direction of motion: these are less relevant and are ignored here. The last is important as it corresponds to rotation about a vertical axis and affects the immediate direction of motion of the vehicle.

To proceed, we have to relate the position (X, Y, Z) of a plant in 3-D with its location $(x,\ y)$ in an image. We can achieve this with a general translation:

$$T = (t_x, t_y, t_z)^{\mathrm{T}} \tag{23.9}$$

together with a general rotation:

$$R = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} \tag{23.10}$$

giving:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \tag{23.11}$$

The lens projection formulae are also relevant:

$$x = fX/Z \tag{23.12}$$

$$y = fY/Z \tag{23.13}$$

We shall not give a full analysis here, but assuming that roll and pitch are zero, and that the heading angle (direction of motion relative to the rows of plants) is ψ, and that this is small, we obtain a quadratic equation for ψ in terms of t_x. This means that two sets of solutions are in general possible. However, it is soon found that only one solution matches the situation, as the wrong solution is not supported by the other feature point positions. This shows the complications introduced by perspective projection—even when highly restrictive assumptions can be made about the geometrical configuration (in particular, ψ being small).

23.9.2 REAL-TIME IMPLEMENTATION

Finally, it was found to be possible to implement the vehicle guidance system on a single processor augmented by two special hardware units—a color classifier and a chaincoder. The latter is useful for fast shape analysis following boundary tracking. The overall system was able to process the input images at a rate of 10 Hz, which is sufficient for reliable vehicle guidance. Perhaps more important, the claimed accuracy was in the region of 10 mm and 1° of angle, making the whole guidance system adequate to cope with the particular slightly constrained application considered. A later implementation (Marchant et al., 1998) did a more thorough job of segmenting the individual plants (though still not using the blob size filter), obtaining a final 5 Hz sampling rate—again fast enough for real-time application in the field. All in all, this case study demonstrates the possibility of highly accurate selective spraying of weeds, thereby very significantly cutting down the amount of herbicide needed for crops such as cabbages, cauliflowers, and wheat.

23.10 CONCLUDING REMARKS

This chapter has considered the value of in-vehicle vision as part of the means for providing driver assistance systems. It has also considered the design of such systems. This process is rendered far from trivial because the camera is necessarily moving, so all objects in a scene will appear to be in motion. Hence, it becomes quite difficult to eliminate the background from consideration, and less easy to

rely on motion-based segmentation. This makes it natural to adopt the alternative approach of placing reliance on recognition of individual objects. Sections 23.2 and 23.3 showed how this concept can be applied to the location not only of the roadway but also of road markings and road signs. The principle also applied to location of vehicles, but as these vary in appearance, it proved necessary to have several distinct methods for locating them, including under-vehicle shadows, symmetry, wheels, and license plates (the latter acting not merely as unique vehicle identifiers but also as characteristics of vehicles in general). Curiously, license plates offered a possible means of finding the orientations of vehicles on roads as well as their locations, though the result was dependent on the relative heights of the camera and license plate under observation. This meant that, when they are not occluded, tire and wheel location will probably be more accurate indicators of vehicle orientation.

Pedestrian location was also seen to be a challenge—particularly as people are articulated objects, and walk with bobbing motions, and also because they tend to have unique appearances and clothing. This makes it natural to use specific templates for leg, arm, head, and body detection rather than whole-body templates. Here, symmetry is also a possible cue as well as skin color. All these approaches were studied in Section 23.7 and tallied with findings in the literature.

The chapter also included aspects of path planning consequent on projecting vehicles and other obstructions onto a plan view of the ground plane: this has some consequence for robot egomotion and navigation. It is also relevant for guidance of agricultural vehicles that are being used for cultivation, selective spraying, and so on. Here, it is also important to consider the much greater degrees of roll, pitch, and yaw that will be experienced by a tractor or other vehicle moving over plowed fields, and the visual compensation needed to cope with this. Some indication was given about how these factors have been coped with: because the principles are known, it seemed better for readers to refer to the original papers for further details.

Finally, we should remark on the almost explosive growth of interest in in-vehicle driver assistance systems, particularly since 2000. This is so important that the following section looks very closely at developments in this area and provides separate bibliographies relating to the various aspects. It was felt that it would be clearer presenting these separately once the principles of the subject had been dealt with, as has been done relatively didactically in the preceding sections.

In-vehicle vision systems necessarily deploy moving cameras, so the usual surveillance strategy of eliminating the stationary background becomes difficult to apply. However, considerable success can be achieved using the alternative strategy of directly locating the most relevant objects, such as the roadway, road signs, road markings, vehicles (e.g., via their symmetry, shadows, wheel, and license plates), and pedestrians (e.g., via their legs, arms, body, and head). Plan views of the ground plane form useful adjuncts to the information obtained in these ways.

23.11 MORE DETAILED DEVELOPMENTS AND BIBLIOGRAPHIES RELATING TO ADVANCED DRIVER ASSISTANCE SYSTEMS

As indicated earlier in the chapter, in recent years (and particularly since 2000), there has been an almost explosive growth of interest in in-vehicle vision systems. The prime though often unwritten underlying aim has been that of driver assistance—a general term that ultimately includes vehicle guidance. However, in 1998, it at first appeared that Bertozzi and Broggi (1998) had largely solved the problem. In fact, they had laid down many of the ground rules, including finding lane markings with the aid of morphological filters, locating obstacles without constraints on symmetry or shape, analyzing stereo images to find free space on the road ahead, removing the perspective effect, implementing the system on a rapidly operating software plus massively parallel hardware architecture, presenting feedback information to the driver via a TV monitor and control panel, testing the system on the road, and above all demonstrating robustness with respect to shadows, changing illumination conditions, varying road texture, and typical motions on the road. Nevertheless, the system was subject to basic assumptions such as the road being flat and road markings being visible; in addition, it placed a great deal of reliance on the stereo system, which had limited range; furthermore, it treated each pair of stereo images individually and was unable to exploit temporal correlations. Finally, while it never failed to detect vehicles on the road ahead, it sometimes detected false obstacles because of noise arising from the various image-remapping processes.

In the light of this work, other workers continued development with increased pace, pressing to eliminate deficiencies with the basic strategy; interestingly, many abandoned the stereo vision approach which brings with it many complications: in fact, appeal to the human vision system demonstrates all too clearly that stereo brings few real advantages for the restricted tasks involved in driving a vehicle (whatever is the case when assembling a gyroscope or other instrument on a workbench). We shall return to this point below.

First, it is worth outlining the findings of Connolly (2009) who has described in a general way the gains to be achieved by advanced driver assistance systems (ADASs). The main keys to success appear to be the provision of lane departure warnings, help for lane changing, collision avoidance, adaptive cruise control, and driver vigilance monitoring. However, it is important that the ADAS should not give too many warnings, or the driver may become annoyed and deactivate it: neither should it fail to act soon enough or give the driver too much confidence or too much freedom. In fact, it is vital for drowsiness to be detected because c. 30% of motorway accidents are caused by drivers undergoing micro or macrosleeps. While much work has been carried out on blink-rate analysis for detecting these conditions, the method has limited effectiveness in probing the state of the brain itself. Nevertheless, it is clear that vision systems can do much to monitor

the driver's behavior, and specifically to monitor his direction of gaze and state of *apparent* awareness. Overall, it is probably in the realm of lane departure warnings and of collision avoidance that an ADAS can do the most good, without annoying the driver. Indeed, in the event of the driver's unawareness of an impending collision, or incapability of acting soon enough, the ADAS should be permitted to act autonomously. While this could in principle be legally contentious, it is not without precedent, as antilock braking systems are in common use.

There are many causes of collision, and a large proportion of them are due to driver error, even when drowsiness is not a specific factor. Failure to see a vehicle or pedestrian because of preoccupation with other events on or off the road, failure to estimate speeds or trajectories of vehicles sufficiently accurately, failure to judge how rapidly braking can be performed in the prevailing conditions, and lack of awareness of what other drivers intend to do are all involved in causing accidents: this list does not include gross vehicle malfunctions such as unpredictable tire bursts. In fact, all these factors arise from or are exacerbated by lack of the right information being available soon enough. Thus, it is obvious that vision has a large part to play in overcoming the problems. While radar, lidar, ultrasonics, or other technologies may help, vision provides far more of the right sort of information with the right sort of response rates, and computer vision should be able to cope reliably and rapidly enough to make this possible. The main questions are as follows: What will be the cost? Where will the cameras be placed? Can enough of them be used to ensure that relevant information is made available? Fortunately, cameras are by now so cheap that cost—relative to that of a vehicle or of the damage caused in a crash—is no longer a serious problem. On the other hand, the real problems are the sophistication and speed of the associated software (or in the latter case, how the system is to be implemented in hardware—a topic for which the reader should refer to other works such as Bailey, 2011). For the remainder of the chapter, we therefore concentrate mainly on the sophisticated software aspects, and what has been achieved since the turn of the millennium.

23.11.1 DEVELOPMENTS IN VEHICLE DETECTION

One area of vital concern has been the detection of other vehicles, especially those overtaking (Zhu et al., 2004; Wang et al., 2005; Hilario et al., 2006; Cherng et al., 2009). The last of these papers considers patterns of driving, such as "cutting" in after overtaking, but more subtly how interactions between events involving more than two vehicles can cause distractions that prevent optimal actions being taken: this is because not all dynamic obstacles are predicable; in fact, multiple critical situations can occur simultaneously. The paper takes the line that the computer must follow attention patterns that emulate those of the human brain and concentrate cyclically on eliminating the various critical phases that are being experienced. The necessary dynamic visual model is in this case tackled using a

spatiotemporal attention neural network. The system of Kuo et al. (2011) concentrates on detecting vehicles on the road ahead but is also able to assess longitudinal distance information and thus to provide adaptive cruise control (albeit no indications of accuracy are given in the paper). Note that this system uses a monocular camera and thus avoids the difficulties of stereo systems mentioned earlier.

Sun et al. (2004, 2006) reviewed the methods used by various workers to detect vehicles. They reported knowledge-based methods using symmetry, color, shadow, corners, horizontal and vertical edges, texture, and lights. In addition, stereo and motion approaches have been used. They also reported template matching and appearance-based methods and noted that sensor fusion is needed to ensure that sufficient information is brought to bear to make vehicle detection reliable. They emphasized that hypothesis generation and verification are important for obtaining reliable solutions. Overall, they offered no silver bullet solution, apart from sensor fusion, though (looking at their conclusions as a whole) *method* fusion appears to be rather more important. Amongst the worst challenges, they found were those of "all hours−all weather" operation. In particular, bad illumination (especially at night) and the results of rain and snow will affect many well-known algorithms for vehicle detection, including those based on shadows. While in principle, vehicle lights should provide an easy way of detecting vehicles, in the dark, they can prove confusing, especially when rain-soaked roads cause reflections. Sun et al. therefore "believe that these cues have limited employability." However, there are bound to be conditions under which some methods will not work well, but by using method fusion in a dynamic way, giving different methods different weights in different conditions, viable solutions should in the end be obtainable. Whereas humans could be confused in dark situations where no information at all is available, it is difficult to imagine them not being able to solve vehicle detection problems because of rain, snow, or random reflections, and certainly not simply because no shadows are visible.

While the difficulties of dealing with the problems of driving at speed on a motorway can be hugely complicated, with vehicles overtaking on either side and sometimes cutting in, the solution is often to drive more slowly thereby minimizing risks and lowering the data rate to manageable levels. However, the problems of dealing with pedestrians are considerably more complicated. This is because, in contrast to the case of vehicles which travel at more or less constant speeds in constant directions for considerable periods of time—and also have a fair amount of free space immediately around them—pedestrians are unpredictable, sometimes running to get across roads between vehicles, sometimes jay-walking, and sometimes moving in groups having even more unpredictable behavior. A basic problem is that it is unknown when a stationary pedestrian might suddenly move into the roadway and with a temporary acceleration that exceeds that of most vehicles. Hence, a great many workers have been and are producing algorithms for pedestrian detection and tracking.

23.11.2 DEVELOPMENTS IN PEDESTRIAN DETECTION

Geronimo et al. (2010) have recently reviewed pedestrian detection systems for ADASs. As this is very thorough and contains 146 references, the reader is recommended to work carefully through it. Nevertheless, some useful points can be made here. They emphasize that pedestrians exhibit high variability in size, pose, clothing, objects carried, and so on; they appear in cluttered scenes, can be partially occluded, and may be in poor contrast regions; they have to be identified in dynamically varying scenes when both they and the camera are moving; they often appear radically different when viewed from different directions. Geronimo et al. note that silhouette matching, e.g., using the chamfer-matching technique, is widely used for detection, yet it needs to be augmented by an additional appearance-based step. (This is not an argument against silhouette matching, but one for using it as a cue, in accordance with the idea expressed above that method fusion is required—i.e., method redundancy is needed to cope robustly with *real* scenes containing substantial clutter.) Geronimo et al. (2010) underline the need for verification and refinement. Interestingly, they note that the Kalman filter is (still) by far the most heavily used tracking algorithm—a surprising fact considering that pedestrian motions along pavements, in precincts or crossing the road exhibit far from steady motion (in fact, their motions tend to be jerky and indecisive, as they find their way around obstacles and other people). Finally, Geronimo et al. emphasize the need for all hours−all weathers performance; here, they note that NIR (near infra-red) imaging gives pictures not dissimilar to visible light images, so similar algorithms can be used for analysis. This is less true for thermal (far infrared or FIR) images, which are commonly called "night vision." In any case, the latter respond to relative temperature, which is useful for distinguishing hot targets, including pedestrians for vehicles, but inappropriate for examining most of the background, or objects such as road signs. Thus, thermal cameras need to be backed up with visible light cameras in the day or NIR cameras in the night and so would generally constitute an unnecessary expense.

Gavrila and Munder (2007) describe a multicue pedestrian detection system: after extensive field tests in difficult urban traffic conditions, they reasonably claim it to be at the (2007) leading edge. The four main detection modules are sparse stereo-based ROI (region of interest) generation, shape-based detection, texture-based classification, and verification using dense stereo, these being complemented by a tracking module. In fact, the paper builds on earlier work (Gavrila et al., 2004), and its main contributions are the method of integration into a multicue system for pedestrian detection and a systematic ROC-based (receiver−operator characteristic based) procedure for parameter setting and system optimization. In part, the success of the system is due to the use of a novel mixture-of-experts architecture for shape and texture-based classification: here, the idea is to take the known shape information and to use texture to partition the feature space into regions of reduced variability—a process that matches well the types of clothing worn by humans. Importantly, the approach using a texture-

based mixture-of-experts weighted by the outcome of shape matching was found to outperform an approach based on single texture classifiers. Also notable is the (continued) use of chamfer matching for shape detection, prominent in much of Gavrila's earlier work.

It was remarked earlier that stereo adds considerable complication to a vision system, which may not be justified for an in-vehicle system when most of the objects being viewed will be many meters away. This makes it no surprise that the review article by Enzweiler and Gavrila (2009) concentrates on monocular pedestrian detection. The paper also included descriptions of a number of experimental comparisons of methods for pedestrian detection. Apart from temporal integration and tracking, methods that were tested included the following: (1) Haar wavelet-based cascades, (2) neural networks using local receptive fields, (3) histograms of orientated gradients (HOGs) together with linear SVM (support vector machine) classifiers, and (4) combined shape and texture-based approaches. The fourth of these was subsequently disregarded as its main advantage was processing speed, which was not considered relevant to the comparison. The investigation found that the HOG approach outperformed the wavelet and neural network approaches (Section 22.16 contains a brief outline of the HOG approach and also explains why it outperforms the wavelet approach in this type of application: see also Section 6.7.8). In particular, at a sensitivity of 70%, the respective false positive rates were 0.045, 0.38, and 0.86, representing huge reduction factors for false positives. [This assumes that the term "detection rate" used by the authors actually means "sensitivity" (or "recall"): see Chapter 14, Machine Learning: Probabilistic Methods.] Similarly, at a sensitivity of 60%, the precision rates were vastly improved for the HOG approach, particularly relative to the neural network approach. It should be emphasized that these results apply for intermediate resolutions with pedestrian images $\sim 48 \times 96$ pixels, while earlier low resolution work with pedestrian images $\sim 18 \times 36$ pixels led to Haar wavelets being the most viable option. Overall, there seemed to be slight doubt about what the critical factors actually are: in particular, the authors state "perhaps it is the data that matters most, after all," meaning that increased performance may be at least partly due to increases in the size of the training set. In addition, quite a bit depends on the processing constraints that are applied, and for tighter constraints, the Haar wavelet approach comes back into its own. However, as ever, it is difficult to standardize or specify image data, or a fortiori, image sequence data, so this paper is not able to tell the whole story. Finally, it should be noted that at this point in time, shape-based detection, and in particular, the chamfer-matching approach, has dropped out of sight because its main advantage was that of speed, and here recognition accuracy measures were the main performance criteria. In this paragraph, note that sensitivity gives a reverse measure of false negative rate, $1-FN/(TP + FN)$, whereas precision gives a reverse measure of false positive rate, $1-FP/(TP + FP)$.

Looking back to the work of Curio et al. (2000)—who use Hausdorff distance rather than chamfer matching for template matching—the attention is very much on analyzing limb movements, modeling human walking and observing human

gait patterns. However, they note that the upper body shows a high degree of variation in its appearance, so it is better to restrict pedestrian detection to the lower body: in fact, this strategy is both more reliable and more computationally efficient. They also point out that exact modeling is more complicated for women wearing skirts. (A similar situation must apply for men wearing robes or mackintoshes.) Overall, just as the driver is aware of motion and gait as well as the body models of pedestrians, these need to be incorporated into practical pedestrian detection algorithms in order to provide maximum reliability and robustness.

Zhang et al. (2007) performed tests on pedestrian detection in "IR (infra-red) images" (these were actually thermal images taken with a camera operating in the spectral range $7-14\ \mu m$). Their motivation was to make a system that was capable of working at night time, though they also noted that many undesirable activities occur at night or in relative darkness, so the methodology should be useful in other applications as well. They found that IR images are by no means dissimilar to visible light images, so similar algorithms can be used for analyzing them: i.e., there is no need to invent radically different methods for the IR domain. In particular, they found that edgelet and HOG methods (see Dalal and Triggs, 2005) could be adapted to work with IR images, and similarly for boosting and SVM cascade classification methods (Viola and Jones, 2001). Hence, they achieved detection performance for IR images comparable to state-of-the-art results for visible light. The underlying reason for this seemed to be that IR and visible light lead to similar silhouettes.

23.11.3 DEVELOPMENTS IN ROAD AND LANE DETECTION

Zhou et al. (2006) developed a lane detection and tracking system using a monocular monochromatic camera. They used a deformable template model to initially locate the lane markings, with tabu search for optimal location; then they used a particle filter for tracking the markings. Their experimental results showed that the resulting system was robust against broken lane markings, curved lanes, shadows, distracting edges and occlusions. Kim (2008) also used a particle filter for tracking lane markings but employed RANSAC for initial detection. Similarly, Mastorakis and Davies (2011) used RANSAC for detection but modified it for increased reliability, as described in Sections 10.4 and 23.3: see also Borkar et al. (2009). Finally, Marzotto et al. (2010) showed how a RANSAC-based system could be implemented in real time using an FPGA (field programmable gate array) platform.

While the above approaches are suitable for urban roads, which normally have well-defined lane markings, many roads, especially in rural regions, are unstructured and lacking in markings—and the road boundaries may be overgrown with vegetation. Cheng et al. (2010) devised a system with the ability to handle both structured and unstructured types of road using a monocular camera. To achieve this, they devised a hierarchical lane detection strategy which was able to achieve high accuracy using quite simple algorithms. First, environment classification of

pixels was carried out with high dimensional feature vectors using eigenvalue decomposition regularized discriminant analysis. For unstructured roads, mean-shift segmentation was used, and then road boundary candidates were selected from the region boundaries: Bayes rule was used to select the most probable of these as actual boundaries. When the vehicle moved from one type of road to another, the environment classifier indicated that a different algorithm should be used so that accuracy could be maintained.

There is one way in which road and lane mapping schemes are restricted—namely, by the view available from the chosen camera. Typically, this will give an overall viewing angle of up to $\sim 45^\circ$. In fact, ideally, a vehicle-borne camera should have a full 360° viewing angle, so that overtaking vehicles and pedestrians about to approach from the side can be seen clearly. Omnidirectional (catadioptric) cameras may be the best answer to this problem, and many workers are actively pursuing this possibility. Cheng and Trivedi (2007) tested a system which used an omnidirectional camera for the dual tasks of lane detection and monitoring the head pose of the driver (the reason for monitoring head pose is to check that the driver is aware of the situation on the road). Their tests showed that accuracy of lane detection is reduced by a factor of (only) $2-3$ because of the reduced resolution available with this sort of camera. Thus, it should prove possible to make savings in the numbers of sensors employed in practical implementations.

23.11.4 DEVELOPMENTS IN ROAD SIGN DETECTION

It is a sign of the seriousness with which ADASs are nowadays being taken that a good many papers describing research into the detection and recognition of road signs have been published since the turn of the millennium. Fang et al. (2003) describe a system that uses neural networks for detecting and tracking road signs by their color and shape. The shapes considered are circles, triangles, octagons, diamonds, and rectangles. Initial detection takes place at some distance, where the road signs appear small and relatively undistorted, and tracking is carried out by a Kalman filter. At each distance, due account is taken of changes in size and shape due to increasing projective distortions, and when a potential sign has become large enough the system verifies that it is a road sign or discards it. Actual recognition is not discussed in the paper, but detection and tracking are said to be accurate and robust: although speed was slow on a single PC (personal computer), the neural networks could conveniently be run in parallel on other processors. A related paper by Fang et al. (2004) describes the types of neural network used in this sort of application. Kuo and Lin (2007) describe a similar system, again involving use of neural networks. The latter paper makes use of greater amounts of structural analysis of the images at the detection stage, e.g., using corner detection, HTs and morphology. De la Escalara et al. (2003) describe a system which starts the analysis using color classification, uses genetic

algorithms for narrowing down the search, and employs neural networks for sign classification.

McLoughlin et al. (2008) describe practically orientated work on road sign detection and also on the detection of "cat's eyes." Their aim is to assess the road signage quality rather than to use it, and to this end, they relate the signs to GPS information. They focus particularly on reflectivity aspects of the signs and are able to detect defective road studs and road signs. Their system is fully autonomous and thus the methodology is largely transferrable to ADASs.

Prieto and Allen (2009) describe a vision-based system for detecting and classifying traffic signs using self-organizing maps (SOM)—a type of neural network. A two-stage detection process is adopted—of first detecting potential road signs by analyzing the distribution of red pixels within the image, and then identifying the road signs from the distribution of dark pixels in their central pictograms. The HT approach and other structural analysis approaches were eschewed because they were felt to operate too slowly for (efficient) real-time operation, so the SOM approach was adopted. To achieve recognition of the pictogram, it was divided into 16 blocks arranged in the form of a triangle (or whatever shape the particular sign was found to possess). It was found necessary to normalize brightness over the region of the sign. The hardware of the embedded machine vision used for this application was a hybrid consisting of an FPGA together with a digital implementation of a SOM. Experiments showed that the system had good performance, being able to tolerate substantial changes in position, scale, orientation and partial occlusion of the road signs, and also being trainable, at least to within the model of colored surround and black on white pictograms. For further details of the SOM and the hybrid implementation, see the original paper and the references mentioned therein.

Ruta et al. (2010) have developed a system not based on neural networks (as for many of the above) but on color distance transforms, coupled with a nearest neighbor recognition system. The color distance transform is actually a set of three distance transforms, one for each color (RGB). If a particular color is absent during testing, it is accorded a maximum distance value of 10 pixels to avoid confusing the system. The color distance transform was tested for dependence on a variety of conditions, such as strong incident light, reflections, and deep shade, and was found to be robust to substantial illumination changes. Perhaps more important, it was found to be reasonably invariant to the effects of affine transformations, which a moving camera would be subject to. This is almost certainly because chamfer matching is subject to graceful degradation as distortions occur, so the distances at any template (edge) locations will *gradually* increase with the changing levels of distortion. When compared with other methods, the method performed well, the percentages of correct classifications being 22.3 for HOG/PCA, 62.6 for Haar/AdaBoost, 74.5 for HOG/AdaBoost, and 74.4 for the new method using the color distance transform. The main competitor to the new method, HOG/AdaBoost, offers an elegant solution but is much more complex than the new method and did not outperform it in any real sense. Hence, the new method seemed well adapted to the task it was set.

23.11.5 DEVELOPMENTS IN PATH PLANNING, NAVIGATION, AND EGOMOTION

The subjects of vehicle guidance and egomotion date from as long ago as 1992 (Brady and Wang, 1992; Dickmanns and Mysliwetz, 1992), whereas automatic visual guidance in convoys dates from a similar period (Schneiderman et al., 1995; Stella et al., 1995). Mobile robots and the need for path planning were discussed by Kanesalingam et al. (1998) and by Kortenkamp et al. (1998), and later a survey was carried out by DeSouza and Kak (2002): see also Davison and Murray (2002). Guidance of outdoor vehicles, particularly on roads, has undergone increasingly rapid development: see for example Bertozzi and Broggi (1998), Guiducci (1999), Kang and Jung (2003), and Kastrinaki et al. (2003). Zhou et al. (2003) considered the situation for elderly pedestrians—though clearly such work could also be relevant for blind people or wheelchair users. Hofmann et al. (2003) showed that vision and radar can profitably be used together to combine the excellent spatial resolution of vision with the accurate range resolution of radar.

In spite of the evident successes, there is still only a limited number of fully automated visual vehicle guidance systems in everyday use. The main problem would appear to be *potential* lack of the robustness and reliability required to trust the system in "all hours−all weathers" situations—though there are also legal implications for a system that is to be used for control rather than merely for vehicle monitoring.

23.12 PROBLEM

1. Check that the path through the maze shown in Fig. 23.14C is optimal, (1) by a hand calculation, and (2) by a computer calculation. Confirm that several other paths are also optimal. Obtain a more accurate result by taking the horizontal and vertical neighbors of any pixel as being 2 units away, and taking diagonal neighbors as being 3 units away.