

Road feature detection and estimation

Stephen Se¹ and Michael Brady²

¹ MD Robotics, 9445 Airport Road, Brampton, Ontario L6S 4J3, Canada

² Department of Engineering Science, University of Oxford, Oxford OX1 3PJ, UK

Received: 20 January 2001 / Accepted: 5 October 2002

Published online: 23 April 2003 – © Springer-Verlag 2003

Abstract. Crosswalks and stair-cases are useful road features for outdoor navigation. In this paper, crosswalks and stair-cases are detected by looking for groups of concurrent lines, and edges are then partitioned using intensity variation information. To distinguish them, three methods are developed to estimate the pose: a homography search approach using an a priori model, and finding the normal using the vanishing line computed from equally spaced lines and likewise with two vanishing points. These algorithms, with error analysis carried out, have been applied to real images with promising results, and they are also useful in other shape-from-texture applications.

Key words: Crosswalk detection – Stair-case detection – Vanishing point – Vanishing line – Homography

1 Introduction

The problem we discuss here arose originally as part of the navigation function of a mobility aid [14, 15], which aims to provide a full mobility and navigation capability for partially sighted people. Crosswalks and stair-cases are useful environmental landmarks that the partially sighted need to be made aware of.

In this paper, we aim to detect the presence of crosswalks and stair-cases in road scenes. The detection algorithm is outlined in the next section. In Sects. 3, 4 and 5, three methods are described to estimate the pose from the texture lines detected, so that these two types of structures can be distinguished. Results and comparison among the three methods are presented in Sect. 6. Novel algorithms are developed to estimate the orientation of the structures from a set of parallel lines and compute the vanishing line from a set of equally spaced parallel lines. As a real application, it is crucial to know how reliable the estimation is, therefore, error analysis is carried out in Sect. 7.

Correspondence to: S. Se (e-mail: sse@mdrobotics.ca)

2 Crosswalk detection

A crosswalk consists of an alternating pattern of black and white stripes, which can be considered as a group of consecutive edges. Crosswalk edges are parallel to each other in 3D space. Weak perspective does not apply because there is considerable variation in depth between the top and bottom stripes. Therefore, when they are projected onto the image, these edges will intersect at a vanishing point (provided that they are not fronto-parallel to the image plane). It is logical to search for concurrent lines when looking for a structure that originally consists of parallel lines.

2.1 The detection algorithm

There are two approaches to finding concurrent lines. The first approach is to search for vanishing points using the Hough transform [1, 4, 17, 19]. After obtaining straight lines using the Hough transform, we can apply another Hough transform to find the intersection of these straight lines. However, Collins and Weiss considered the vanishing point computation as a statistical estimation problem and observed that this approach is not reliable when not many lines are passing through that point [6]. The accuracy level degrades notably as the number of lines drops from 20 down to 5.

The second approach, similar to Utcke [20], is employed here where potential groups of candidate lines are generated and then tested for coincidence. The Canny edge detector [5] is applied to the image to detect the edge points, followed by Hough transform line fitting to obtain lines. RANSAC (random sample consensus) [9] is employed to find a group of lines satisfying the vanishing point constraint, by eliminating outlier lines. A least-squares procedure is then used to find the vanishing point, i.e., the intersection point among this group of lines [18].

2.2 Edges partition

Using this technique based on the vanishing point constraint, we obtain a hypothesis for some structure containing parallel

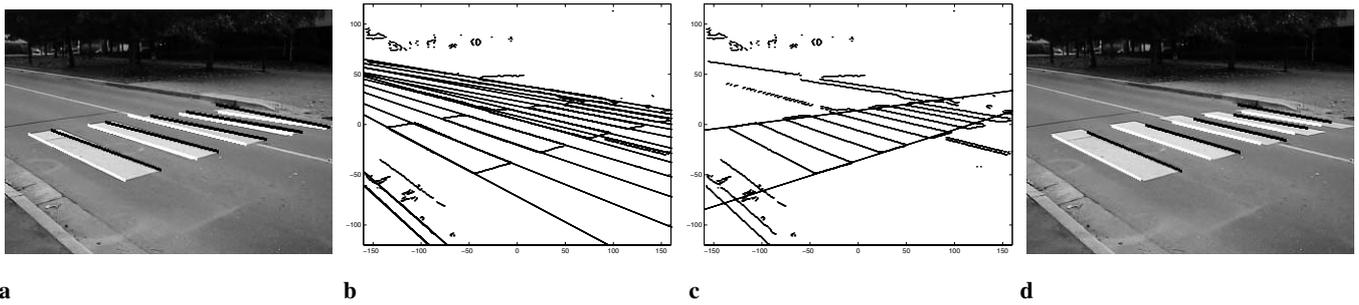


Fig. 1a–d. A typical crosswalk. **a** Original image with detected edges overlaid. **b** Line detection result. **c** Side lines detection. **d** Same as **a** but with side lines removed

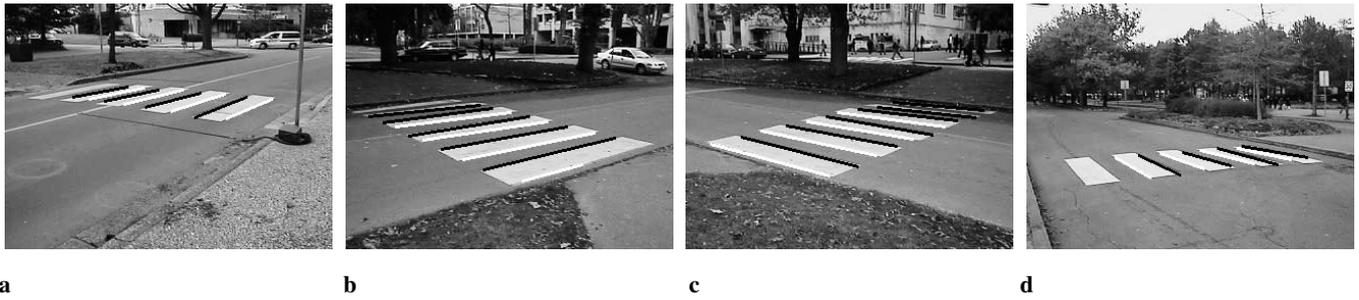


Fig. 2a–d. Other crosswalk images. Detected edges are partitioned and overlaid on the original images

lines. To verify that the structure is a crosswalk, we add a further constraint that it consists of an alternating pattern of black and white stripes. In fact, the edges can be partitioned into two sets of equally spaced parallel lines, corresponding to light-to-dark and dark-to-light transitions. This is a much stronger constraint compared to merely searching for structures with parallel lines. For instance, in some Legoland scenes where there are a lot of structures with parallel lines, many hypotheses will be found and this constraint is useful to eliminate the false ones.

Intensity variation is considered here as a cue on which to base the partition, i.e., to detect changes of intensity from white to black and from black to white. To facilitate the position extraction of these changes, we use the average intensity to threshold the image first. Afterward, the intensity profile across the crosswalk clearly exhibits a pattern of ups and downs, where changes can be localized easily.

2.3 Detection results

We are most interested in regular crosswalks whose centre-lines are perpendicular to the stripe pattern. Figure 1a shows a typical crosswalk of 320×240 image resolution, and the line detection result is shown in Fig. 1b. Using the intensity variation, we partition the edges into two groups. The edges are overlaid on the image in Fig. 1a with the dark-to-light transitions marked in white and the light-to-dark transitions marked in black.

It can be seen that some detected lines do not fit the actual edges too well. This is due to slight image distortion and interference from side edges. Therefore, we find the two side lines first in the orthogonal quadrant direction and remove them before the detection. Figure 1c shows the two side lines found

while Fig. 1d shows the new result where the real edges are fitted much better. More results are shown in Fig. 2.

However, the endpoints of the edges now are less accurate, because the side lines do not fit the actual side edges too well; therefore, removing the side lines also discards some edge points. It is because the road surface is not flat but slightly curved, with the curb sides a bit lower so that rain water drains into the gutters.

2.4 Stair-case detection

This algorithm also works for stair-cases because stair-cases are characterized as a sequence of steps, which can be regarded as a group of consecutive parallel edges. The detection algorithm above has been applied to various indoor and outdoor stair-case images, with some results shown in Fig. 3. We can see that stair-cases are detected successfully by finding groups of coincident lines.

Stair-case edges can also be partitioned into two groups of edges (concave and convex), as an alternating intensity pattern corresponding to the tread and riser of the stair-case steps can be observed.

We show in Fig. 4 an image sequence in which a user walks towards a stair-case. Using the algorithm above, concave and convex edges are found and overlaid on the images with convex edges marked in white and concave ones marked in black. It can be seen that the partition is stable, with most edges correctly classified.

The stair-case edges form a virtual slanting plane; therefore, both crosswalks and stair-cases can be considered planar. As a result, using concurrent lines as the image feature with intensity variation is not sufficient to distinguish these two types of parallel structure. They are both useful road features

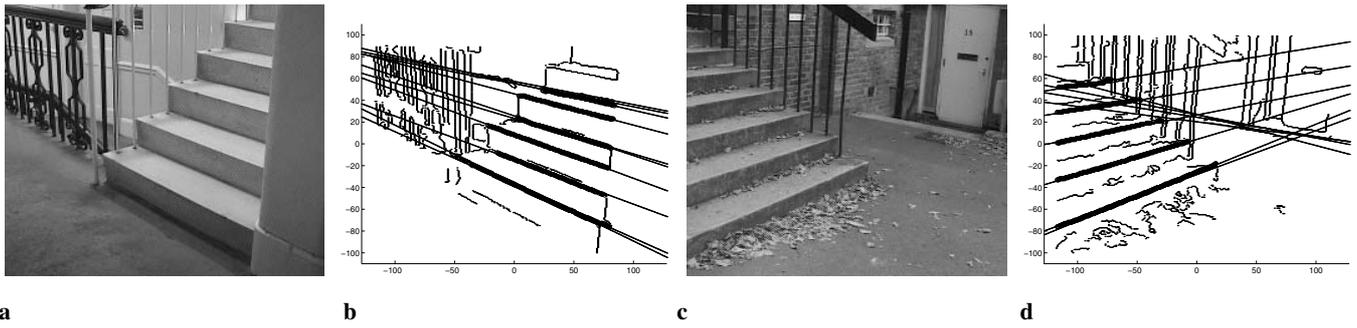


Fig. 3a–d. Stair-case detection results. **a** An indoor stair-case. **b** Detection result for **a**. **c** An outdoor stair-case. **d** Detection result for **c**

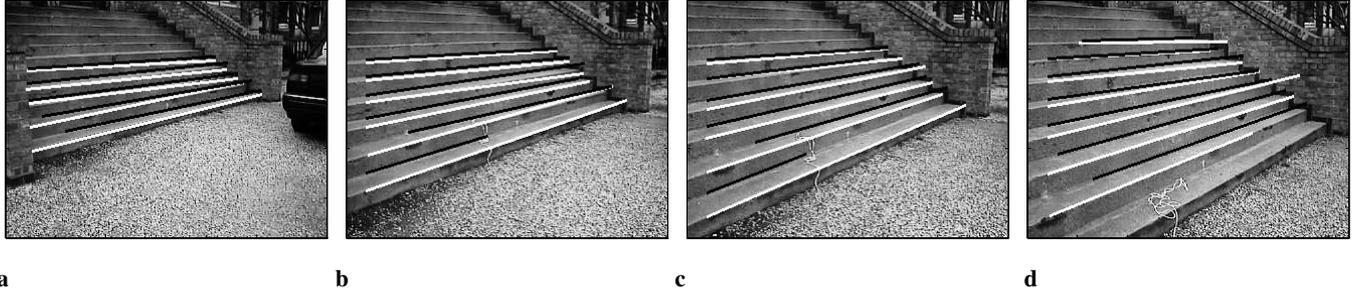


Fig. 4a–d. Image sequence overlaid with the concave and convex edges found on the stair-case. Convex edges are marked in *white* while concave ones are marked in *black*

for outdoor navigation, but it is necessary to distinguish them to react differently.

Since all crosswalks lie on the ground while stair-cases do not, pose information will allow us to differentiate them. If the parallel structure with alternating pattern also has a null slope, then it is confirmed as a crosswalk, otherwise it is a stair-case. We will now look at some pose estimation techniques for these two types of structure in the following sections.

3 Homography search

We use a search approach which is similar to Witkin’s search for tilt and slant from texture [21]. However, in the general shape-from-texture literature [3,10–12,21], isotropy of texture is assumed. In Witkin’s case, a maximum likelihood estimator is derived to compute the tilt and slant, which will give the best isotropy texture on back-projection. He did not assume natural textures to be particularly uniform, but assumed that their non-uniformity does not mimic projection. However, if the true texture is not isotropic, but has a preferred orientation, it mimics a projected image, and therefore it is impossible to detect the true orientation.

For both crosswalk and stair-case structures, we can consider the edges as a textured plane. Since the orientation of each edge is the same, the texture is anisotropic, and therefore an a priori model is required. The model we adopt here is a group of non-skewed parallel horizontal lines on the image when it is facing the camera.

The shape-from-texture for our textured plane is a more constrained problem than a general textured surface, as there are only two rotational pose components: one around the vertical axis (vertical rotation θ); the other around the horizontal axis (slope ϕ).

Once we have obtained the vertical rotation θ and slope ϕ , we can compute the tilt τ and slant σ parameters, which are often used in the shape-from-texture literature, by the following equations:

$$\tau = \tan^{-1} \left(\frac{1}{\tan \phi \sin \theta} \right); \quad \sigma = \cos^{-1}(\sin \phi \cos \theta).$$

Our aim is to transform the image to another view by a homography so that the camera in the new view will be facing the structure directly. We employ criteria based on our model while we search in a discretized space of (θ, ϕ) .

3.1 The homography

Here, we look at the transformation of an image of one view to another view induced by a plane. Based on the initial world coordinates frame, the equation for a plane π is

$$\mathbf{N}^\top \mathbf{X} = \mathbf{X} \cdot \mathbf{N} = d, \quad (1)$$

where \mathbf{N} is its normal. The relationship between the old view coordinates \mathbf{X}_1 and the new view coordinates \mathbf{X}_2 is given by

$$\mathbf{X}_2 = \mathbf{R}\mathbf{X}_1 + \mathbf{t},$$

where \mathbf{R} is the rotation matrix and \mathbf{t} is the translation vector. From this, we obtain

$$\begin{aligned} \mathbf{X}_2 &= \mathbf{R}\mathbf{X}_1 + \frac{\mathbf{t}d}{d} \\ &= \mathbf{R}\mathbf{X}_1 + \frac{\mathbf{t}\mathbf{N}^\top \mathbf{X}_1}{d} = \left(\mathbf{R} + \frac{\mathbf{t}\mathbf{N}^\top}{d} \right) \mathbf{X}_1, \end{aligned}$$

and this homography can be expressed as

$$\mathbf{x}_2 = \left(\mathbf{R} + \frac{\mathbf{t}\mathbf{N}^\top}{d} \right) \mathbf{x}_1 = \mathbf{H}\mathbf{x}_1, \quad (2)$$

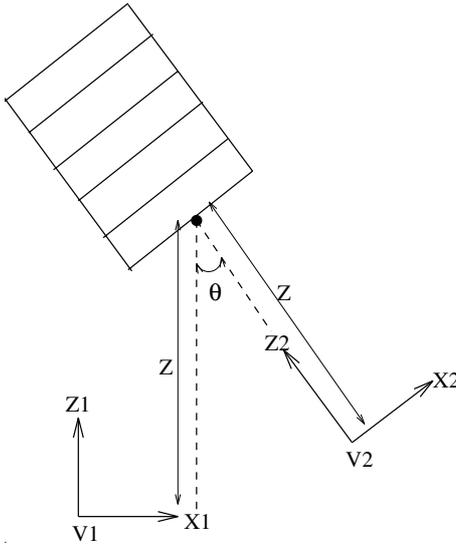


Fig. 5. Relationship between the two views V1 and V2 defining the rotation matrix R and translation vector t in the homography

where x_1 and x_2 are normalized image coordinates [8]. From Fig. 5, in order to make the camera face the structure directly in the new view V2,

$$R = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix},$$

and $t = (-X \cos \theta - Z \sin \theta, 0, X \sin \theta + Z - Z \cos \theta)^\top$, where (X, Y, Z) is the 3D position of the structure which has been detected and localized. For a planar structure with vertical rotation θ and slope ϕ , the surface normal N is $(\sin \phi \sin \theta, \cos \phi, -\sin \phi \cos \theta)^\top$. As its position (X, Y, Z) is found and lies on this plane, we have $d = X \sin \phi \sin \theta + Y \cos \phi - Z \sin \phi \cos \theta$.

This homography holds provided that we are working with normalized coordinates. Therefore, the coordinates should be normalized before applying the homography transformation and denormalized afterward. To normalize the coordinate system, we need to determine the matrix K which takes the old image coordinates m_{old} and gives the normalized coordinates m_{new} ; i.e., $m_{\text{new}} = K m_{\text{old}}$.

Similarly, we have the projective relationship between the image coordinates and the world coordinates M , i.e., $m_{\text{old}} = P_{\text{old}} M$. Therefore, we have

$$m_{\text{new}} = K P_{\text{old}} M = P_{\text{new}} M.$$

m_{new} will be the normalized coordinates if

$$P_{\text{new}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Referring to Fig. 6, to transform from the world coordinates (X_w, Y_w, Z_w) to the image coordinates (u, v) , we have

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = M_{\text{int}} M_{\text{proj}} M_{\text{rot}} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}, \quad (3)$$

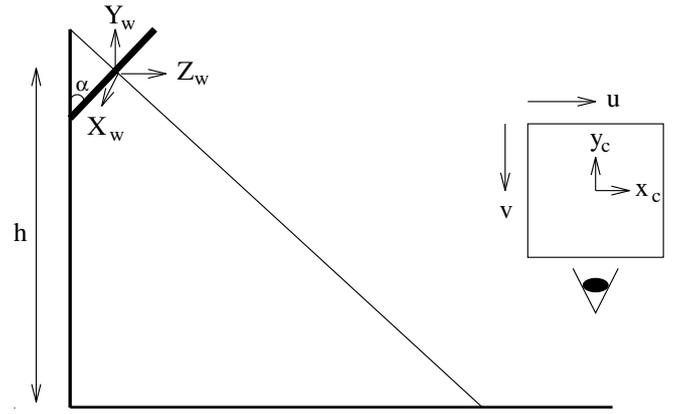


Fig. 6. Transformations between world coordinates and image coordinates

where

$$M_{\text{rot}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha & 0 \\ 0 & -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, M_{\text{proj}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

$$M_{\text{int}} = \begin{bmatrix} f k_u & 0 & u_0 \\ 0 & -f k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix},$$

where α is the camera tilt, $f k_u$ and $f k_v$ are the focal intrinsic parameters, and (u_0, v_0) are the intrinsic coordinates of the image centre. Expanding Eq. (3) above, we have

$$P_{\text{old}} = \begin{bmatrix} f k_u & -u_0 \sin \alpha & u_0 \cos \alpha & 0 \\ 0 & -f k_v \cos \alpha - v_0 \sin \alpha & v_0 \cos \alpha - f k_v \sin \alpha & 0 \\ 0 & -\sin \alpha & \cos \alpha & 0 \end{bmatrix};$$

and since $P_{\text{new}} = K P_{\text{old}}$, the normalizing matrix K is

$$K = \begin{bmatrix} \frac{1}{f k_u} & 0 & -\frac{u_0}{f k_u} \\ 0 & -\frac{\cos \alpha}{f k_v} & \frac{v_0}{f k_v} \cos \alpha - \sin \alpha \\ 0 & -\frac{\sin \alpha}{f k_v} & \frac{v_0}{f k_v} \sin \alpha + \cos \alpha \end{bmatrix}.$$

Hence, the homography for our coordinate system is expressed as

$$y_2 = K^{-1} H(\theta, \phi) K y_1, \quad (4)$$

where y_1 and y_2 are our image coordinates in the old and new views respectively.

3.2 Search criteria

There are two components of our model-based search criteria: one for the vertical rotation and the other for the slope.

From Fig. 7a, b, c and f, the criterion for the correct vertical rotation is based on how horizontal the transformed texture lines are. This can be expressed as searching for θ which gives the lowest sum of squares for the slopes of the image lines.

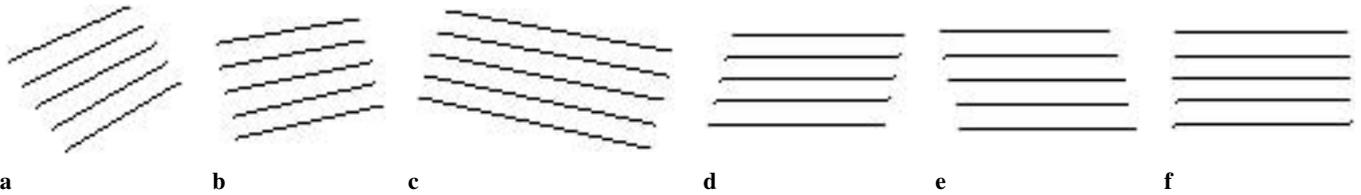


Fig. 7a–f. Search criteria. **a** Original texture edges. **b** The transformed edges when the predicted vertical rotation is below the true value. **c** The transformed edges when the predicted vertical rotation is above the true value. **d** The transformed edges when the predicted slope is below the true value. **e** The transformed edges when the predicted slope is above the true value. **f** The transformed edges when the predicted vertical rotation and slope are correct

At the correct vertical rotation, the camera will be facing the stair-case edges head-on and so horizontal edges are expected.

From Fig. 7a, d, e and f, the criterion for the correct slope is based on how skewed the transformed textures lines are. The correct one will correspond to the case when the mid-points of all the texture lines lie on a vertical line. We measure the standard deviation for the u -coordinates of the image line midpoints and search for ϕ which gives the lowest standard deviation.

We observe that how horizontal the lines are is only dependent on the vertical rotation but independent of the slope, so the algorithm can proceed in two stages. Firstly, we assume an arbitrary value for ϕ , and perform a one-dimensional search on θ using the vertical rotation criterion. Knowing θ , we then proceed as another one-dimensional search on ϕ using the slope criterion.

This reduces the complexity of the search algorithm as two one-dimensional searches are performed instead of one two-dimensional search. To reduce the complexity further, we can employ a coarse-to-fine search strategy [18]. Furthermore, instead of a full-range search, some gradient-descent scheme can be employed.

3.3 Ill-conditioned case

The search algorithm fails to determine the slope when the vertical rotation is 0° , i.e. when the camera is facing head-on to the structure. Looking at the homography in Eq. (2), when $\theta = 0^\circ$,

$$\mathbf{n} = \begin{bmatrix} 0 \\ \cos \phi \\ -\sin \phi \end{bmatrix}, \quad \mathbf{t} = \begin{bmatrix} -X \\ 0 \\ 0 \end{bmatrix},$$

$$\mathbf{H} = \mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The homography equals the identity matrix irrespective of the hypothesised slope ϕ , so the transformed lines are the same whatever ϕ is. The search always determines the vertical rotation but fails to find the slope if the vertical rotation turns out to be zero.

4 Vanishing line

During the detection stage, we have partitioned the edges into two groups corresponding to the two intensity transitions. For

each group, we can use the fact that they are equally spaced to compute the vanishing line of that plane to estimate its normal. The normal in general gives information about its slope and vertical rotation, but when the plane is horizontal, it only provides the slope.

4.1 Vanishing line and normal

A 3D point \mathbf{X} is projected to camera coordinates $\mathbf{x} = f\mathbf{X}/Z$. For a very distant point \mathbf{X} on plane π given by Eq. (1),

$$\mathbf{x} \cdot \mathbf{N} = \lim_{Z \rightarrow \infty} \frac{fd}{Z} = 0. \quad (5)$$

Its image pixel coordinates is given by $\mathbf{u} = \mathbf{C}\mathbf{x}$, where \mathbf{C} is the intrinsic camera parameters matrix. Substituting $\mathbf{x} = \mathbf{C}^{-1}\mathbf{u}$ into $\mathbf{x}^\top \mathbf{N} = 0$, we have $\mathbf{u}^\top \mathbf{C}^{-\top} \mathbf{N} = 0$.

As a vanishing line is the projection of a line at infinity, \mathbf{u} , the projection of \mathbf{X} , will be lying on the vanishing line; therefore, the vanishing line \mathbf{l}_∞ is defined as follows:

$$\mathbf{l}_\infty = \mathbf{C}^{-\top} \mathbf{N}. \quad (6)$$

Then we can determine the normal:

$$\mathbf{N} = [N_X, N_Y, N_Z]^\top = \mathbf{C}^\top \mathbf{l}_\infty. \quad (7)$$

A dot product with the normal of the ground plane $(0, 1, 0)$ allows us to compute the slope. As our camera system is tilted downwards by α , the estimated slope ϕ is given by

$$\phi = \cos^{-1} \left(\frac{N_Y}{|\mathbf{N}|} \right) - \alpha. \quad (8)$$

4.2 Conjugate translation transformation

Using homogeneous coordinates, the homography from Eq. (2) is of the form

$$\mathbf{H} = \mathbf{C} \left(\mathbf{R} + \frac{\mathbf{t}\mathbf{N}^\top}{d} \right) \mathbf{C}^{-1}$$

for non-normalized image coordinates.

For the pattern of parallel lines, we are only translating in direction \mathbf{t} perpendicular to the lines along the plane, so $\mathbf{R} = \mathbf{I}$,

$$\mathbf{H} = \mathbf{C} \left(\mathbf{I} + \frac{\mathbf{t}\mathbf{N}^\top}{d} \right) \mathbf{C}^{-1} = \mathbf{I} + \frac{1}{d} (\mathbf{C}\mathbf{t})(\mathbf{N}^\top \mathbf{C}^{-1}).$$

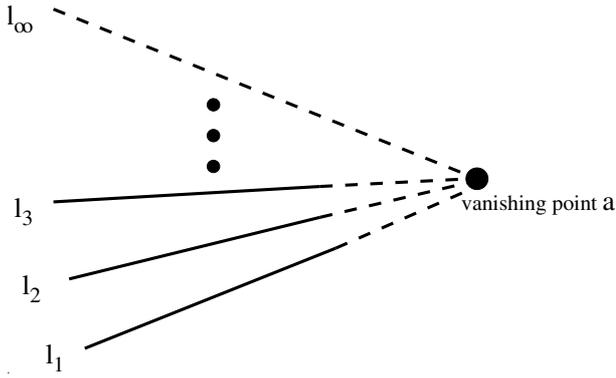


Fig. 8. The pencil of equally-spaced edges together with the vanishing line

Let $\mathbf{v} = \mathbf{C}\mathbf{t}$, where \mathbf{v} is the vanishing point of the translation direction, and from Eq. (6), $\mathbf{N}^\top \mathbf{C}^{-1} = \mathbf{l}_\infty^\top$, we have

$$\mathbf{H} = \mathbf{I} + \lambda \mathbf{v} \mathbf{l}_\infty^\top,$$

where λ is a scalar representing the magnitude of translation. Because the direction of translation is perpendicular to the plane normal,

$$\begin{aligned} 0 &= \mathbf{t} \cdot \mathbf{N} = \mathbf{t}^\top \mathbf{C}^\top \mathbf{C}^{-\top} \mathbf{N} \\ &= (\mathbf{C}\mathbf{t})^\top (\mathbf{C}^{-\top} \mathbf{N}) \implies \mathbf{v} \cdot \mathbf{l}_\infty = 0. \end{aligned}$$

4.3 Finding the vanishing line

For the three equally spaced lines in Fig. 8, the conjugate translation transformation derived above can be used to obtain \mathbf{l}_2 and \mathbf{l}_3 from \mathbf{l}_1 :

$$\mathbf{l}_2 \propto \mathbf{l}_1 + \lambda \mathbf{l}_\infty^\top \mathbf{l}_1, \quad (9)$$

$$\mathbf{l}_3 \propto \mathbf{l}_1 + 2\lambda \mathbf{l}_\infty^\top \mathbf{l}_1, \quad (10)$$

where \propto denotes ‘equal up to a scale’.

Since all these lines are parallel in 3D, they pass through some vanishing point \mathbf{a} in the image. As the vanishing line also passes through \mathbf{a} , we can express the vanishing line as a linear combination of any two lines using the concurrency property:

$$\mathbf{l}_\infty = \beta \mathbf{l}_2 + \gamma \mathbf{l}_3. \quad (11)$$

Taking the vector product of Eq. (9) with \mathbf{l}_2 and Eq. (10) with \mathbf{l}_3 , we obtain the following:

$$\mathbf{l}_1 \wedge \mathbf{l}_2 + \mu \mathbf{l}_\infty \wedge \mathbf{l}_2 = \mathbf{0}, \quad (12)$$

$$\mathbf{l}_1 \wedge \mathbf{l}_3 + 2\mu \mathbf{l}_\infty \wedge \mathbf{l}_3 = \mathbf{0}, \quad (13)$$

where $\mu = \lambda \mathbf{v}^\top \mathbf{l}_1$. Taking the vector product of Eq. (11) with \mathbf{l}_2 and then with \mathbf{l}_3 , we obtain the following:

$$\mathbf{l}_\infty \wedge \mathbf{l}_2 = \gamma \mathbf{l}_3 \wedge \mathbf{l}_2; \quad (14)$$

$$\mathbf{l}_\infty \wedge \mathbf{l}_3 = \beta \mathbf{l}_2 \wedge \mathbf{l}_3. \quad (15)$$

From Eqs. (13) and (15), and similarly from Eqs. (12) and 14, we further obtain the following:

$$\begin{aligned} \beta &= -\frac{1}{2\mu} \frac{(\mathbf{l}_1 \wedge \mathbf{l}_3) \cdot (\mathbf{l}_2 \wedge \mathbf{l}_3)}{|\mathbf{l}_2 \wedge \mathbf{l}_3|^2}; \\ \gamma &= -\frac{1}{\mu} \frac{(\mathbf{l}_1 \wedge \mathbf{l}_2) \cdot (\mathbf{l}_3 \wedge \mathbf{l}_2)}{|\mathbf{l}_3 \wedge \mathbf{l}_2|^2}. \end{aligned}$$

Substituting these back into Eq. (11), we obtain

$$\mathbf{l}_\infty \propto [(\mathbf{l}_1 \wedge \mathbf{l}_3) \cdot (\mathbf{l}_2 \wedge \mathbf{l}_3)] \mathbf{l}_2 + 2[(\mathbf{l}_1 \wedge \mathbf{l}_2) \cdot (\mathbf{l}_3 \wedge \mathbf{l}_2)] \mathbf{l}_3. \quad (16)$$

5 Two vanishing points

A vanishing point provides a constraint on the orientation of the plane. Two such independent constraints define a vanishing line and determine the orientation of the plane uniquely. This method is commonly used to obtain the vanishing line in various applications such as Criminisi et al. [7]. In [13], two sets of parallel lines are used to find the vanishing line followed by applying additional constraints for rectification of projective images.

Since selecting a group of lines which converge in the image at a vanishing point is the criterion for our detection algorithm, we have the first vanishing point. In Sect. 2.3, we detected the two side lines of a regular crosswalk, which are parallel in 3D, converging at another vanishing point. Hence we have two vanishing points for this plane. Equation (5) gives us the vanishing line on the image plane: $xN_X + yN_Y + fN_Z = 0$, where (x, y) are the camera coordinates, f is the focal length and $[N_X, N_Y, N_Z]^\top$ is the normal.

Assuming the first vanishing point is (u_1, v_1) , the second vanishing point is (u_2, v_2) and the image centre is at (u_0, v_0) , the vanishing line equation is

$$\frac{y - (v_0 - v_2)/k_v}{x - (u_2 - u_0)/k_u} = \frac{(v_0 - v_1)/k_v - (v_0 - v_2)/k_v}{(u_1 - u_0)/k_u - (u_2 - u_0)/k_u},$$

where k_u and k_v are the pixels-per-unit-length parameters for the u and v coordinates respectively. For square pixels ($k = k_u = k_v$), this gives

$$\begin{aligned} (v_2 - v_1)x + (u_2 - u_1)y \\ + f \frac{(u_2 - u_0)(v_1 - v_2) - (v_0 - v_2)(u_2 - u_1)}{fk} = 0. \end{aligned}$$

The normal of the plane is found and its slope is estimated using Eq. (8).

6 Results and comparison

For each crosswalk image shown above, we apply these three techniques to estimate its pose, in particular its slope. Since the crosswalks lie on the ground, their slopes are expected to be close to 0° . We can also obtain the vertical rotation with the homography search approach. The results are tabulated in Table 1.

The two-vanishing-points method is most simple, but requires the vanishing point of the two side lines, and it is not

Table 1. VR (vertical rotation) estimated from HS (homography search) and slope estimated from HS, VL (vanishing line) with either groups and 2 VPs (two vanishing points) methods for the crosswalk images

Images	VR: HS	Slope: HS	Slope: VL(1)	Slope: VL(2)	Slope: 2VPs
Fig 1(d)	-48°	-16°	-4.72°	-5.68°	-4.81°
Fig 2(a)	54°	-12°	-3.07°	-2.69°	-10.93°
Fig 2(b)	32°	-13°	-0.36°	-0.04°	-8.97°
Fig 2(c)	-34°	-10°	-5.47°	-0.76°	-9.46°
Fig 2(d)	-41°	-13°	-9.82°	3.90°	-5.06°

applicable if part of the structure is occluded. Moreover, the second vanishing point is obtained from just the two side lines, and hence it is error-prone. Edges do not need to be partitioned.

The homography search approach can estimate the orientation as well as the slope, whereas the other methods can only find the slope. However, to estimate the slope, it makes use of the edge endpoints, which are error-prone. It is not applicable if part of the structure is occluded. This approach does not require partitioning the edges. Points can alternatively be used to find the homography, but lines are employed for better robustness. Erroneous points in a point-based approach can lead to larger errors than in a line-based approach, as each line is fitted from multiple points.

The vanishing line method does not need the side lines or the endpoints, therefore it works under occlusion. However, it requires a group of equally spaced lines; therefore, it is feasible only if the parallel structure itself is equally spaced, or if its edges can be partitioned into groups of equally spaced lines. As the crosswalk edges are partitioned into two groups, this method gives an estimate for each group, as shown in Table 1.

For the homography search approach, reprojection error, limited by image resolution, could introduce errors of up to 3 degrees in the estimation. The road surface, which is not exactly flat, also contributes to the slope estimation errors. From the results, we can see that the vanishing line method is more accurate as it does not depend on the endpoints. Moreover, in practice, the full extent of the structure may not be always observed, therefore this method is preferable.

7 Error analysis

Only the homography search approach can estimate the orientation, and the comparison of the three methods above indicates that the vanishing line method estimates the slope better. Therefore, we carry out error analysis for these estimation, as reliability and quantitative measures are important.

7.1 Homography search

There are three stages in our search for vertical rotation: homography, line fitting, and then slope minimization. Firstly, all the points undergo the homography $\mathbf{H}(\theta)$:

$$\begin{bmatrix} u_{i,j} \\ v_{i,j} \end{bmatrix} \xrightarrow{\mathbf{H}(\theta)} \begin{bmatrix} u(\theta)'_{i,j} \\ v(\theta)'_{i,j} \end{bmatrix}, \quad (17)$$

where $i = 1, \dots, m$ denotes each line with n_i points and $j = 1, \dots, n_i$ denotes the points on the line.

This is followed by an orthogonal least-squares fitting on the points of each line i . With the line equation being $v = su + l$, where s is the slope and l is the intercept, we have

$$\min_{s(\theta)_i, l_i} C = \frac{1}{s(\theta)_i^2 + 1} \sum_{j=1}^{n_i} (s(\theta)_i u(\theta)'_{i,j} - v(\theta)'_{i,j} + l_i)^2. \quad (18)$$

Finally, we minimize the sum of squares of the slopes of all lines:

$$\min_{\theta} g = \sum_{i=1}^m s(\theta)_i^2.$$

We can combine the two minimizations in the second and third stages above by setting $s(\theta)_i$ in Eq. (18) to zero, and we have

$$\min_{\theta, l_i} f = \sum_{i=1}^m \sum_{j=1}^{n_i} (v(\theta)'_{i,j} - l_i)^2,$$

where $v(\theta)'_{i,j}$ can be expanded using the homography in Eq. (4) [18]. Taking partial derivatives with respect to θ and l_i and setting to zero, we obtain the function Φ which defines θ and l_i 's parameters (denoted by \mathbf{y}) implicitly in terms of input \mathbf{x} , which consists of $X, Y, Z, u_{i,j}$'s and $v_{i,j}$'s [8]. By the implicit function theorem, we have

$$Df(\mathbf{x}) = - \left[\frac{\partial \Phi}{\partial \mathbf{y}} \right]^{-1} \left[\frac{\partial \Phi}{\partial \mathbf{x}} \right],$$

and therefore the covariance matrix for \mathbf{y} is

$$\text{cov}_{\mathbf{y}} = Df(\mathbf{x}) \text{cov}_{\mathbf{x}} Df(\mathbf{x})^{\top},$$

where $\text{cov}_{\mathbf{x}}$ is the covariance matrix for $X, Y, Z, u_{i,j}$'s and $v_{i,j}$'s, which is a square matrix of dimension $(3 + 2 \sum_{i=1}^m n_i)$. Assuming the image points are independent and the variance σ^2 is the same for each point in both u and v directions, it is a diagonal matrix with σ_X^2, σ_Y^2 and σ_Z^2 in the first three positions of the diagonal and the rest being σ^2 .

Figure 9 shows the relationship between the estimated vertical rotation θ^* (degrees) and its standard deviation σ_{θ^*} (degrees) for a synthetic image scene. It can be seen that the error of the algorithm is higher when the vertical rotation is small. Therefore, shape from texture is not effective for small slant, which was also shown by Blake and Marinos [3].

7.2 Vanishing line

We only need 3 lines to determine a vanishing line, but usually more than 3 edges from each group are found. In order to eliminate outliers, we apply RANSAC to select 3 lines at random and compute the vanishing line l_{∞} . We repeat and select the triple with the most supporting edges to compute the vanishing line.

The 3 lines selected each time may not necessarily be consecutive, so we need to extend Eq. (16) to deal with 3 arbitrary

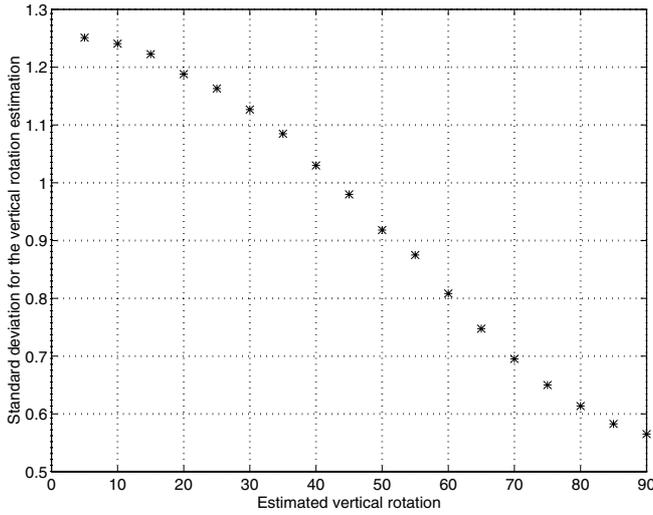


Fig. 9. Standard deviation of the vertical rotation estimation at different vertical rotation

lines of known order. Given the lines l_i , l_j and l_k corresponding to the i th, j th and k th edges found, and they being unequal, we have

$$l_\infty \propto (j-i)[(l_i \wedge l_k) \cdot (l_j \wedge l_k)]l_j + (k-i)[(l_i \wedge l_j) \cdot (l_k \wedge l_j)]l_k, \quad (19)$$

and we can obtain equations to solve for μ like Eqs. (12) and (13). Then we can check if a line l_l supports the current triple, by computing $E = l_i \wedge l_l + (l-j)\mu l_\infty \wedge l_l$. If $|E|$ is greater than some threshold value, then line l_l is considered as an outlier.

As Eq. (19) simply contains some cross products and dot products, we can use the standard error propagation formulae [2] to obtain the variances for the coefficients of the vanishing line. Assuming the calibration matrix C is accurate, from Eq. (7), we can compute the variance for the components of the normal. Rewriting Eq. (8) as

$$\phi = \tan^{-1} \left(\frac{\sqrt{N_X^2 + N_Z^2}}{N_Y} \right) - \alpha,$$

for a function $y = g(x)$, to a first-order approximation [16], the variance of y is given by $\sigma_y^2 \approx [g'(x)]^2 \sigma_x^2$. Therefore, in our case, we have

$$\sigma_\phi^2 \approx [\cos^2(\tan^{-1} x)]^2 \sigma_x^2 + \sigma_\alpha^2, \quad (20)$$

where

$$x = \frac{\sqrt{N_X^2 + N_Z^2}}{N_Y},$$

$$\sigma_x^2 = \frac{N_X^2 \text{var}(N_X) + N_Z^2 \text{var}(N_Z)}{N_Y^2 (N_X^2 + N_Z^2)} + \frac{(N_X^2 + N_Z^2) \text{var}(N_Y)}{N_Y^4},$$

and σ_α^2 is the variance for the camera tilt, which can be measured using the inclinometer on the system or estimated from the ground plane parameters [18].

Table 2. VR (vertical rotation) estimated from HS (homography search) and slope estimated from VL (vanishing line) with standard deviation for the crosswalk images

Crosswalk Images	VR with HS (s.d.)	Slope with VL (s.d.)
Fig 1d	$-48^\circ(4.8^\circ)$	$-4.8^\circ(2.7^\circ)$
Fig 2a	$54^\circ(8.7^\circ)$	$-2.9^\circ(9.5^\circ)$
Fig 2b	$32^\circ(7.4^\circ)$	$-0.2^\circ(2.2^\circ)$
Fig 2c	$-34^\circ(8.9^\circ)$	$-3.5^\circ(3.9^\circ)$
Fig 2d	$-41^\circ(21.0^\circ)$	$3.4^\circ(6.5^\circ)$

Table 3. VR (vertical rotation) estimated from HS (homography search) and slope estimated from VL (vanishing line) with standard deviation for the stair-case images

Stair-case Images	VR with HS (s.d.)	Slope with VL (s.d.)
Fig 4a	$38^\circ(0.5^\circ)$	$26.2^\circ(3.2^\circ)$
Fig 4b	$45^\circ(0.3^\circ)$	$24.6^\circ(3.9^\circ)$
Fig 4c	$49^\circ(0.3^\circ)$	$29.9^\circ(0.6^\circ)$
Fig 4d	$48^\circ(0.3^\circ)$	$25.2^\circ(2.6^\circ)$

From Eq. (20), σ_ϕ decreases as x increases. Therefore, the slope error decreases when the slope is less steep, i.e., when there is more perspective effect among the lines.

Since we have two groups of edges, we obtain two estimates of ϕ with their corresponding variances and then compute a weighed least-squares estimate.

7.3 Results

The pose estimation results with their standard deviation for the crosswalk and stair-case images are tabulated in Table 2 and Table 3 respectively.

The results show that all the crosswalk slopes estimated are close to 0° and are significantly different from the stair-case slopes; therefore, slope estimation allows us to distinguish between them. In addition, the vertical rotation estimation facilitates navigation to approach them.

8 Conclusion

In this paper, we look into crosswalk and stair-case detection by grouping lines and checking for concurrency using the vanishing point constraint. Intensity variation is used to partition the edges afterward. As crosswalks and stair-cases are inherently different 3D structures which have similar 2D properties, pose information is required to distinguish them. Three techniques are presented for pose estimation based on homography search, vanishing line and two vanishing points. Detailed error analysis indicates how reliable the estimation is, which is of utmost importance for applications used by human.

The contributions of this paper include the novel algorithms to estimate the vertical rotation from a set of parallel lines using the homography search approach, and to estimate the vanishing line from a set of equally spaced parallel lines. These requirements are easier to fulfill than using two sets of parallel lines [7, 13], when part of the structure may be occluded.

The experiments demonstrate that our detection and pose estimation algorithms can identify crosswalks and stair-cases successfully in real road scene images. The results show that slope can be estimated well with the vanishing line method and this allows differentiation between crosswalks and stair-cases. Vertical rotation can be estimated with the homography search approach and this facilitates navigation towards them.

The algorithms are by no means limited to mobility-aids applications. These road features are important landmarks for outdoor mobile robots, such as in map-building applications or for navigation purposes. The pose estimation techniques are also useful in other shape-from-texture applications.

These algorithms are working, though slow and far from real-time. Future work includes optimization, further trials with different scenes to evaluate their robustness and performance, and trajectory planning to approach the crosswalk/stair-case found.

Acknowledgements. We thank Andrew Zisserman for many helpful discussions.

References

- Barnard ST (1983) Interpreting perspective images. *Artif Intell* 21(4):435–462
- Bevington PR, Robinson DK (1992) *Data reduction and error analysis for the physical sciences*, 2nd edn. McGraw-Hill, New York
- Blake A, Marinos C (1990) Shape from texture: estimation, isotropy and moments. *Artif Intell* 45(3):323–380
- Brillault-O'Mahony B (1991) New method for vanishing point detection. *Comput Vision Graph Image Process* 54(2):289–300
- Canny J (1986) A computational approach to edge detection. *IEEE Trans Pattern Anal Mach Intell* 8(6):679–698
- Collins RT, Weiss RS (1990) Vanishing point calculation as a statistical inference on the unit sphere. In: *Proceedings of the third international conference on computer vision*, Osaka, Japan, December 1990. IEEE Computer Society Press, Los Alamitos
- Criminisi A, Zisserman A, Van Gool L, Bramble S, Compton D (1998) A new approach to obtain height measurements from video. In: *Proceedings of SPIE*, vol 3576, Boston, Mass., November 1998. The Society, Bellingham, Wash.
- Faugeras O (1993) *Three-dimensional computer vision: a geometric viewpoint*. MIT Press, Cambridge, Mass.
- Fischler MA, Bolles RC (1981) Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography. *Commun Assoc Comput Mach* 24:381–395
- Garding J (1993) Shape from texture and contour by weak isotropy. *Artif Intell* 64(2):243–297
- Gibson JJ (1950) *The perception of the visual world*. Houghton Mifflin, Boston, Mass.
- Kanatani K (1984) Detection of surface orientation and motion from texture by a stereological technique. *Artif Intell* 23:213–237
- Liebowitz D, Zisserman A (1998) Metric rectification for perspective images of planes. In: *Proceedings of IEEE conference on computer vision and pattern recognition*, June 1998. IEEE Computer Society Press, Los Alamitos
- Molton N, Se S, Brady JM, Lee D, Probert P (1998) A stereo vision-based aid for the visually impaired. *Image Vision Comput* 16(4):251–263
- Molton N, Se S, Brady M, Lee D, Probert P (1999) Robotic sensing for the partially sighted. *Robotics Auton Syst* 26:185–201
- Papoulis A (1965) *Probability, random variables, and stochastic processes*. McGraw-Hill, New York
- Quan L, Mohr R (1989) Determining perspective structures using hierarchical Hough transform. *Pattern Recogn Lett* 9:279–286
- Se S (1998) *Computer vision aids for the partially sighted*. PhD thesis, Department of Engineering Science, University of Oxford
- Tuytelaars T, Van Gool L, Proesmans M, Moons T (1998) The cascaded Hough transform as an aid in aerial image interpretation. In: *Proceedings of the sixth international conference on computer vision*, Bombay, India, 4–7 January 1998. Narosa Publishing House, New Delhi
- Utcke S (1998) Grouping based on projective geometry constraints and uncertainty. In: *Proceedings of the sixth international conference on computer vision*, Bombay, India, 4–7 January 1998. Narosa Publishing House, New Delhi
- Witkin AP (1981) Recovering surface shape and orientation from texture. *Artif Intell* 17(1–3):17–45



Stephen Se is currently with the research and development department at MD Robotics in Canada, developing computer vision systems for space and terrestrial applications. He completed a BEng degree with first class honours in Computing at Imperial College of Science, Technology and Medicine, London in 1995 and a DPhil degree in the Robotics Research Group at the University of Oxford in 1999. His DPhil thesis is on computer vision aids for the partially sighted. His research interests include computer vision, robotics, image processing and artificial intelligence. He has published in international journals and conference proceedings.



Michael Brady FRS, FREng BP Professor of Information Engineering at the University of Oxford. His degrees are in mathematics (BSc and MSc from Manchester University, and PhD from the Australian National University). He combines his work at Oxford University, where he founded the Robotics Laboratory and the Medical Vision Laboratory, with a range of entrepreneurial activities. He is the author of over 275 articles and the author/editor of nine books. He has been awarded honorary doctorates by the universities of Essex, Manchester, Liverpool, Southampton, Paul Sabatier (Toulouse), the IEE Faraday Medal for 2000 and the IEEE Third Millennium Medal for UK.