Zebra-crossing Detection for the Partially Sighted

Stephen Se
Department of Computer Science, University of British Columbia
Vancouver, B.C. Canada V6T 1Z4
se@cs.ubc.ca

Abstract

Zebra-crossings are useful road features for outdoor navigation in mobility aids for the partially sighted. In this paper, zebra-crossings are detected by looking for groups of concurrent lines, edges are then partitioned using intensity variation information. In order to tackle the ambiguity of the detection algorithm in distinguishing zebra-crossings and stair-cases, pose information is sought. Three methods are developed to estimate the pose: homography search approach using an a priori model; finding normal using the vanishing line computed from equally-spaced lines and with two vanishing points. These algorithms have been applied to real images with promising results and they are also useful in some other shape from texture applications.

1 Introduction

The inadequacy of information provided by long canes and the very limited acceptability of guide dogs have prompted the development of electronic mobility aids for millions of partially sighted people worldwide. The problem we discuss here arose originally as part of the navigation function of a **T**echnological **A**id aimed at helping **P**artially **S**ighted (TAPS) [12, 13] which aims to provide a full mobility and navigation capability for partially sighted people.

Mobility has been defined by Foulke [7] as "the ability to travel safely, comfortably, gracefully, and independently through the environment". The ability to detect road features is an important component of any TAPS as this will facilitate independent outdoor navigation for the partially sighted.

Curbs, stair-cases and zebra-crossings are useful environmental landmarks that the partially sighted needs to be made aware of. First, a curb [18] is a hazard that could cause a partially sighted person to stumble, and it signifies the boundary between road and pavement. A stair-case [14, 19] may be a location by which to orient themselves or an important way point along the route that they wish to travel. Zebra-crossings are crucial for identifying intersections and crossing roads.

In this paper, we aim to detect the presence of zebra-crossings in road scenes. The detection algorithm is outlined in the next section. In Sections 3, 4 and 5, three methods are described to estimate the pose with results presented in Section 6.

2 Zebra-crossing Detection

Zebra-crossing consists of an alternating pattern of black and white stripes, which can be considered as a group of consecutive edges. Zebra-crossing edges are parallel to each other in 3D space. 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 Searching for Vanishing Point

There are two ways to find concurrent lines. One approach is to search for vanishing points using Hough Transform [1, 15, 20, 11, 3]. After obtaining straight lines using Hough Transform, we can apply another Hough Transform to find the intersection of these straight lines. The vanishing points are characterized as those points where most of the supporting line segment primitives intersect, so we can accumulate evidence provided by these line segments.

However, Collins and Weiss [4] considered vanishing point computation as a statistical estimation problem and observed that it is not reliable when not many lines are passing through that point. The accuracy level stays roughly the same as the number of lines drops from 100 down to 20, but degrades notably from 20 down to 5. In fact, any convergent group consisting of relatively small number of lines will be left undetected with this approach.

2.2 The Detection Algorithm

A different approach is proposed here where potential groups of candidate lines are generated and then tested for coincidence. Utcke [21] used a similar approach for grouping and recognizing zebra-crossing taking into account image feature uncertainties, cross-ratio and vanishing line constraints. This approach was employed to detect stair-cases in [14, 19].

Hough Transform line fitting is applied to the image first. Based on the projective property of structures with parallel lines, our algorithm picks out groups of nearly parallel lines and checks for concurrency (hence finding the vanishing point) as hypotheses for zebra-crossings. Then it seeks support from the other lines for these hypotheses, to determine the best hypothesis.

RANSAC [6] is employed to eliminate outlier edges and a least-squares procedure is then used to find the intersection of multiple lines [17].

2.3 Edges Partition

Using this technique based on the vanishing point constraint, we obtain a hypothesis for some structure containing parallel lines. To verify that the structure consists of an alternating pattern of black and white stripes, we add a further constraint that the edges can be partitioned into two sets of equally-spaced parallel lines. 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. Geometric constraints such as cross-ratios can be applied to refine the sets of edges.

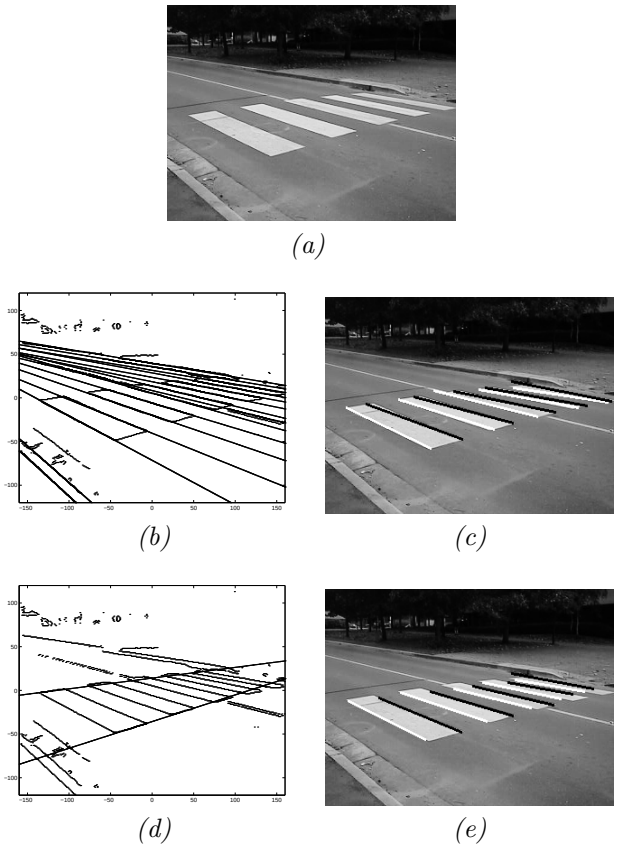
2.4 Detection Results

We are most interested in regular zebra-crossings whose centre-lines are perpendicular to the stripe pattern. Figure 1(a) shows a typical 320x240 zebra-crossing and the detection result is shown in Figure 1(b). Using the intensity variation, we partition the edges into two groups. The edges are overlaid on the image in Figure 1(c) 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 partly due to slight image distortion such that straight zebra-crossing edges may not project to perfectly straight lines. Moreover, interference from side edges also affects the Hough Transform lines.

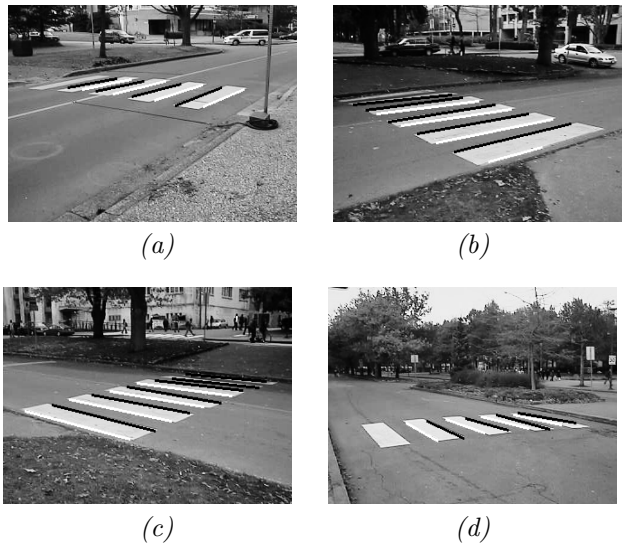
Therefore, we can find the two side lines first in the orthogonal quadrant direction and remove them from the original image before the detection. Figure 1(d) shows the two side lines found while Figure 1(e) shows the result using the new image in which the real edges are fitted much better. More detection results are shown in Figure 2.

However, the endpoints of the edges now are less accurate, as the side lines do not fit the actual side edges too well, therefore removing the side lines also discards



(a) *(b)* *(c)* *(d)* *(e)*

Figure 1: *A typical zebra-crossing. (a) The original image. (b) The detection result. (c) Detected edges are partitioned and overlaid on (a). (d) Side lines detection. (e) Same as (c) but with side lines removed.*



(a) *(b)* *(c)* *(d)*

Figure 2: *Other zebra-crossing images. Detected edges are partitioned and overlaid on the original images.*

Figure 3: *A stair-case image with concave and convex edges detected.*

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.

2.5 Zebra-crossing and Stair-case

The algorithm above 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. Stair-case edges can also be partitioned into concave and convex edges, as alternating intensity pattern can be observed, as shown in Figure 3. Using concurrent lines as the image feature with intensity variation is not sufficient to distinguish these two types of parallel structures. The stair-case edges form a virtual slanting plane, therefore, they both can be considered as planar.

Since all zebra-crossings lie on the ground, whereas stair-cases do not, therefore, 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 zebra-crossing. We will now look at some pose estimation techniques for these two types of structures 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 [22]. However, in the general shape from texture literature [9, 22, 10, 2, 8], 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 backprojection.

For both zebra-crossing 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:

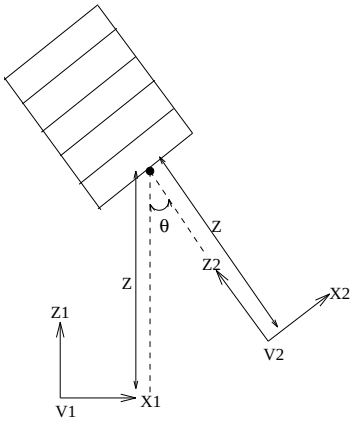


Figure 4: *Relationship between the two views V1 and V2 defining the rotation matrix* $\mathbf{R}$ *and translation vector* $\mathbf{t}$ *in the homography.*

one around the vertical axis (vertical rotation θ); the other around the horizontal axis (slope ϕ).

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 in one view to another view induced by a plane. Based on the initial world coordinates frame, the equation for a plane π is:

$$\mathbf{N}^\top \mathbf{X} \;=\; \mathbf{X} \cdot \mathbf{N} \;=\; d \tag{1}$$

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, we have

$$\mathbf{X}_2 = \mathbf{R}\mathbf{X}_1 + \frac{\mathbf{t}d}{d} = \mathbf{R}\mathbf{X}_1 + \frac{\mathbf{t}\mathbf{N}^\top \mathbf{X}_1}{d} = (\mathbf{R} + \frac{\mathbf{t}\mathbf{N}^\top}{d})\mathbf{X}_1$$

then this homography can be expressed as

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

where $\mathbf{x}_1$ and $\mathbf{x}_2$ are normalized image coordinates [5]. From Figure 4, in order to make the camera face the structure directly in the new view V2,

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

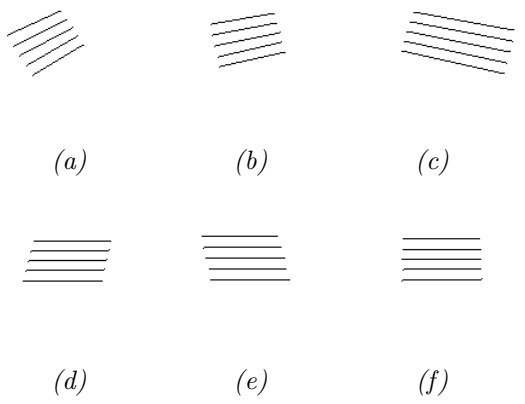


Figure 5: *Search criteria. (a) Original texture edges. (b) The transformed edges when the predicted vertical rotation is below the true value. (c) The transformed edges when the predicted vertical rotation is above the true value. (d) The transformed edges when the predicted slope is below the true 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.*

and $\mathbf{t} = (-X\cos\theta - Z\sin\theta, 0, X\sin\theta + Z - Z\cos\theta)^\top$ 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 $\mathbf{N}$ 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 de-normalized afterwards [17].

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.

Figure 5(a) shows a synthetic image of texture lines with a certain vertical rotation θ and slope ϕ. Figures 5(b), 5(c) and 5(f) indicate that 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 absolute values for the slopes of the image lines.

In fact, if the vertical rotation prediction is right, after the homography transformation, the camera will be facing the texture edges head-on and so horizontal edges are expected.

Similarly, Figures 5(d), 5(e) and 5(f) show that 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 midpoints of the texture lines all lie on a vertical line. We can quantify this by computing the standard deviation of the u-coordinates for all the midpoints of the image lines. That is, we search for ϕ which gives the lowest standard deviation.

We observe that only vertical rotation can make the image lines become not horizontal, therefore, how horizontal the lines are is 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 [17].

4 Vanishing Line

During the detection stage, we have partitioned the edges into two groups of equally-spaced edges 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

$$\mathbf{x} = \frac{f\mathbf{X}}{Z}$$

So, for a very distant point $\mathbf{X}$ on plane π given by Equation 1,

$$\mathbf{x}\cdot\mathbf{N} = \lim_{Z\to\infty}\frac{fd}{Z} = 0 \tag{3}$$

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,

$$\mathbf{l}_\infty = \mathbf{C}^{-\top}\mathbf{N} \tag{4}$$

Then we can determine the normal:

$$\mathbf{N} = [N_X, N_Y, N_Z]^\top = \mathbf{C}^\top\mathbf{l}_\infty \tag{5}$$

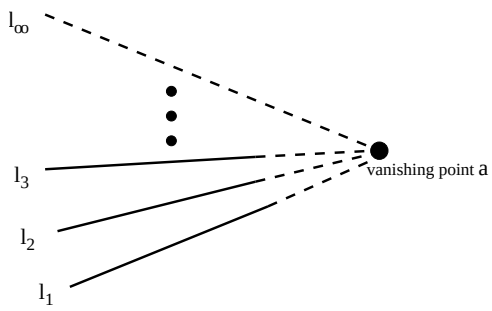


Figure 6: *The pencil of equally-spaced edges together with the vanishing line.*

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 γ, the estimated slope ϕ is given by:

$$\phi = \cos^{-1}(\frac{N_Y}{|\mathbf{N}|}) - \gamma \tag{6}$$

4.2 Finding the Vanishing Line

Here, we use three equally-spaced image lines to compute the vanishing line. Knowing that the three lines are equally-spaced lying on a plane in 3D, we can express the conjugate translation transformation [16] as

$$\mathbf{H} = \mathbf{I} + \lambda \mathbf{v} \mathbf{l}_\infty^\top \quad \text{with} \quad \mathbf{v} \cdot \mathbf{l}_\infty = 0$$

where $\mathbf{I}$ is the identity matrix, $\mathbf{l}_\infty$ is the vanishing line of the plane, $\mathbf{v}$ is the vanishing point of the translation direction and λ is a scalar representing the magnitude of the translation. Referring to Figure 6, this transformation can be used to obtain $\mathbf{l}_2$ and $\mathbf{l}_3$ from $\mathbf{l}_1$.

$$\mathbf{l}_2 \propto \mathbf{l}_1 + \lambda \mathbf{l}_\infty \mathbf{v}^\top \mathbf{l}_1 \tag{7}$$

$$\mathbf{l}_3 \propto \mathbf{l}_1 + 2\lambda \mathbf{l}_\infty \mathbf{v}^\top \mathbf{l}_1 \tag{8}$$

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, and the vanishing line also passes through $\mathbf{a}$. For this pencil of lines, we can express the vanishing line as a linear combination of any two lines, for instance,

$$\mathbf{l}_\infty = \alpha \mathbf{l}_2 + \beta \mathbf{l}_3 \tag{9}$$

using the concurrency property.

Taking the vector product of Equation 7 with $\mathbf{l}_2$ and Equation 8 with $\mathbf{l}_3$, we obtain:

$$\mathbf{l}_1 \wedge \mathbf{l}_2 + \mu \mathbf{l}_\infty \wedge \mathbf{l}_2 = \mathbf{0} \tag{10}$$

$$\mathbf{l}_1 \wedge \mathbf{l}_3 + 2\mu \mathbf{l}_\infty \wedge \mathbf{l}_3 = \mathbf{0} \tag{11}$$

where $\mu = \lambda \mathbf{v}^\top \mathbf{l}_1$.

Taking the vector product of Equation 9 with $\mathbf{l}_2$ and then with $\mathbf{l}_3$, we have:

$$\mathbf{l}_\infty \wedge \mathbf{l}_2 = \beta \mathbf{l}_3 \wedge \mathbf{l}_2 \tag{12}$$

$$\mathbf{l}_\infty \wedge \mathbf{l}_3 = \alpha \mathbf{l}_2 \wedge \mathbf{l}_3 \tag{13}$$

From Equations 11 and 13, similarly from Equations 10 and 12, we get:

$$\alpha = -\frac{1}{2\mu} \frac{(\mathbf{l}_1 \wedge \mathbf{l}_3) \cdot (\mathbf{l}_2 \wedge \mathbf{l}_3)}{|\mathbf{l}_2 \wedge \mathbf{l}_3|^2}$$

$$\beta = -\frac{1}{\mu} \frac{(\mathbf{l}_1 \wedge \mathbf{l}_2) \cdot (\mathbf{l}_3 \wedge \mathbf{l}_2)}{|\mathbf{l}_3 \wedge \mathbf{l}_2|^2}$$

Substituting these back into Equation 9, we obtain:

$$\mathbf{l}_\infty \propto [(\mathbf{l}_1 \wedge \mathbf{l}_3) \cdot (\mathbf{l}_2 \wedge \mathbf{l}_3)]\mathbf{l}_2 + 2[(\mathbf{l}_1 \wedge \mathbf{l}_2) \cdot (\mathbf{l}_3 \wedge \mathbf{l}_2)]\mathbf{l}_3 \tag{14}$$

4.3 RANSAC

We only need 3 lines to determine a vanishing line, but usually more than 3 edges from each group are found. We can apply RANSAC to select 3 lines at random, compute the vanishing line $\mathbf{l}_\infty$ and the parameter μ, then find the number of support from all the edges. Repeat and select the triple which has the maximum support to compute the vanishing line. This way can help eliminate possible outliers.

The 3 lines selected each time may not necessarily be consecutive, so we need to extend Equation 14 to deal with 3 arbitrary lines of known order. Given the lines $\mathbf{l}_i$, $\mathbf{l}_j$ and $\mathbf{l}_k$ corresponding to the i^{th}, j^{th} and k^{th} edges found and they are unequal, we have:

$$\mathbf{l}_\infty \propto (j-i)[(\mathbf{l}_i \wedge \mathbf{l}_k) \cdot (\mathbf{l}_j \wedge \mathbf{l}_k)]\mathbf{l}_j + (k-i)[(\mathbf{l}_i \wedge \mathbf{l}_j) \cdot (\mathbf{l}_k \wedge \mathbf{l}_j)]\mathbf{l}_k \tag{15}$$

and we can obtain equations to solve for μ like Equations 10 and 11. Then we can check if a line $\mathbf{l}_l$ supports the current triple, by computing

$$E = \mathbf{l}_i \wedge \mathbf{l}_l + (l-i)\mu \mathbf{l}_\infty \wedge \mathbf{l}_l$$

If $|E|$ is greater than some threshold value δ, then line $\mathbf{l}_l$ is considered as an outlier.

Moreover, we may further extend this so that the vanishing line is computed using all supporting lines instead of just using 3 lines to improve the accuracy.

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.

Since selecting a group of lines which converge in the image at a vanishing point is the criterion for our detection algorithm, we obtain the first vanishing point.

In Section 2.4, we have detected the two side lines of a regular zebra-crossing, which are parallel lines in 3D, converging in the image at another vanishing point. Hence we have two vanishing points for this plane.

Equation 3 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), the principal centre is at (u_0, v_0), the equation of the vanishing line across the two vanishing points is:

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

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:

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

Therefore, the normal of the plane is found and its slope can then be estimated using Equation 6.

6 Results and Comparison

For each zebra-crossing image shown above, we apply these three techniques to estimate its pose, in particular its slope. We can obtain its vertical rotation as well from the homography search approach. The results are tabulated in Table 1.

Images	VR: HS	Slope: HS	Slope: VL(s.d.)	Slope: 2VPs
Fig 1(e)	$-48°$	$-16°$	$-4.75°(2.69°)$	$-4.81°$
Fig 2(a)	$54°$	$-12°$	$-2.94°(9.53°)$	$-10.93°$
Fig 2(b)	$32°$	$-13°$	$-0.19°(2.17°)$	$-8.97°$
Fig 2(c)	$-34°$	$-10°$	$-3.53°(3.91°)$	$-9.46°$
Fig 2(d)	$-41°$	$-13°$	$-3.69°(3.24°)$	$-5.06°$
Fig 3	$45°$	$26°$	$24.58°(3.85°)$	-

Table 1: VR (vertical rotation) estimated from HS (homography search) and slope estimated from HS, VL (vanishing line) with standard deviation and 2VPs (two vanishing points) methods for the images

The orientation of the zebra-crossing is indicated by the vertical rotation estimate. A negative value means that one has to turn right appropriately to approach the zebra-crossing, whereas a positive value is for turning left.

Since zebra-crossings are horizontal approximately, the slope should be close to null, whereas stair-case slope is not. The results show that their slope estimated is significantly different from that of a stair-case, therefore, slope estimation will allow us to confirm the identity of zebra-crossings detected.

For Figure 3, since one side of the stair-case is outside of the view, the two vanishing points method cannot be used. The slope criterion in the homography search cannot apply but an equal spacing constraint is employed instead [19].

The two vanishing points method is most simple, but requires the vanishing point of the two side lines, and it would not be applicable if part of the structure is occluded. Although the first vanishing point is obtained as the least-squares intersection of many edges, the second vanishing point is obtained just from two 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 would not be applicable if part of it is occluded, as endpoints will be missing. This approach does not require partitioning the edges.

The vanishing line method does not need the side lines nor 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. It is less error-prone except when some lines are missing, as all lines are being used. In the case of zebra-crossing, it gives an estimate for each group. A weighed least-squares estimate is computed from the two estimates with the error analysis carried out [17].

From the results, we can see that the homography search and the two vanishing points methods are not as accurate as the vanishing line method. The former methods depend on the endpoints (lower accuracy as discussed in Section 2.4) while the latter does not. The road surface, which is not exactly flat, also contributes to the slope estimation errors of the zebra-crossings.

7 Conclusion

In this paper, we look into zebra-crossing detection by grouping lines and checking for concurrency using the vanishing point constraint. Intensity variation is used to partition the edges afterwards. However, this approach also works for stair-cases, therefore, pose information is required to distinguish them. Three techniques are presented for pose estimation based on homography search, vanishing line and two vanishing

points. The experimental results presented show that the detection algorithm with pose estimation allows us to identify zebra-crossings in road scene images.

Although the algorithms have been developed for TAPS, they are by no means limited to such applications. Zebra-crossing are important landmarks for outdoor mobile robots, for example in map-building applications or for navigation purposes.

The homography search approach can be used to estimate pose for other parallel texture lines planar structures. The vanishing line method is applicable when they consist of equally-spaced lines whereas the two vanishing points method is applicable when they consist of two independent groups of parallel lines, e.g. grid pattern. If the plane is not horizontal, the normal estimated from the vanishing line or the two vanishing points method can also provide the orientation.

These algorithms are working, though slow and far from real-time, and have not yet been integrated into our TAPS prototype system. Future works include optimization, further trials with different scenes to evaluate their robustness and performance, and trajectory planning to approach the zebra-crossing found.

Acknowledgements

We thank Michael Brady and Andrew Zisserman from University of Oxford for many helpful discussions.

References

[1] S.T. Barnard. Interpreting perspective images. *Artificial Intelligence*, 21(4):435–462, November 1983.

[2] A. Blake and C. Marinos. Shape from texture: Estimation, isotropy and moments. *Artificial Intelligence*, 45(3):323–380, 1990.

[3] B. Brillault-O'Mahony. New method for vanishing point detection. *Computer Vision, Graphics, and Image Processing*, 54(2):289–300, September 1991.

[4] R.T. Collins and R.S. Weiss. Vanishing point calculation as a statistical inference on the unit sphere. In *Proceedings of the Third International Conference on Computer Vision*, pages 400–403, Osaka, Japan, December 1990.

[5] O. Faugeras. *3 Dimensional Computer Vision - A Geometric Viewpoint*. MIT Press, 1993.

[6] M.A. Fischler and R.C. Bolles. Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography. *Commun. Assoc. Comp. Mach.*, 24:381–395, 1981.

[7] E. Foulke. The perceptual basis for mobility. *American Foundation for the Blind Research Bulletin*, 23:1–8, 1971.

[8] J. Garding. Shape from texture and contour by weak isotropy. *Artificial Intelligence*, 64(2):243–297, 1993.

[9] J.J. Gibson. *The Perception of the Visual World*. Houghton Mifflin, Boston, 1950.

[10] K. Kanatani. Detection of surface orientation and motion from texture by a stereological technique. *Artificial Intelligence*, 23:213–237, 1984.

[11] M.J. Magee and J.K. Aggarwal. Determining vanishing points from perspective images. *Computer Vision, Graphics, and Image Processing*, 26:256–267, 1984.

[12] N. Molton, S. Se, J.M. Brady, D. Lee, and P. Probert. Robotic sensing for the guidance of the visually impaired. In *International Conference on Field and Service Robotics FSR'97*, pages 236–243, December 1997.

[13] N. Molton, S. Se, J.M. Brady, D. Lee, and P. Probert. A stereo vision-based aid for the visually impaired. *Image and Vision Computing*, 16(4):251–263, 1998.

[14] N. Molton, S. Se, M. Brady, D. Lee, and P. Probert. Robotic sensing for the partially sighted. *Robotics and Autonomous Systems*, 26:185–201, 1999.

[15] L. Quan and R. Mohr. Determining perspective structures using hierarchical hough transform. *Pattern Recognition Letters*, 9:279–286, 1989.

[16] F. Schaffalitzky and A. Zisserman. Geometric grouping of repeated elements within images. In J.N. Cartor and M.S. Nixon, editors, *Proceedings of British Machine Vision Conference BMVC'98*, pages 13–22, Southampton, September 1998.

[17] S. Se. *Computer Vision Aids for the Partially Sighted*. PhD thesis, Department of Engineering Science, University of Oxford, 1998.

[18] S. Se and M. Brady. Vision-based detection of kerbs and steps. In A.F. Clark, editor, *Proceedings of British Machine Vision Conference BMVC'97*, pages 410–419, Essex, September 1997.

[19] S. Se and M. Brady. Vision-based detection of staircases. In *Fourth Asian Conference on Computer Vision ACCV 2000, Volume I*, pages 535–540, Taipei, January 2000.

[20] T. Tuytelaars, L. Van Gool, M. Proesmans, and T. Moons. The cascaded hough transform as an aid in aerial image interpretation. In *Proceedings of the Sixth International Conference on Computer Vision*, pages 67–72, Bombay, India, January 1998.

[21] S. Utcke. Grouping based on projective geometry constraints and uncertainty. In *Proceedings of the Sixth International Conference on Computer Vision*, pages 739–746, Bombay, India, January 1998.

[22] A.P. Witkin. Recovering surface shape and orientation from texture. *Artificial Intelligence*, 17(1-3):17–45, August 1981.