PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY B BIOLOGICAL SCIENCES

Vision for Mobile Robots

Michael Brady and Han Wang

Phil. Trans. R. Soc. Lond. B 1992 **337**, 341-350
doi: 10.1098/rstb.1992.0112

Vision for mobile robots

MICHAEL BRADY AND HAN WANG

Robotics Research Group, Department of Engineering Science, University of Oxford, Oxford OX1 3PJ, U.K.

SUMMARY

Localized feature points, particularly corners, can be computed rapidly and reliably in images, and they are stable over image sequences. Corner points provide more constraint than edge points, and this additional constraint can be propagated effectively from corners along edges. Implemented algorithms are described to compute optic flow and to determine scene structure for a mobile robot using stereo or structure from motion. It is argued that a mobile robot may not need to compute depth explicitly in order to navigate effectively.

1. INTRODUCTION

This paper is concerned with the information that can be computed efficiently and reliably by a mobile robot to support effective navigation, including obstacle avoidance. Object recognition, and the generation of collision avoiding trajectories are discussed elsewhere (Reid & Brady 1992; Hu *et al.* 1991). Though we are concerned primarily with mobile robots, the issues we discuss have also figured in our work on model-based (reduced bandwidth) image coding, medical image processing, and attentional control, for example to support automated surveillance systems.

Mobile robots are used increasingly in flexible manufacturing systems, as a programmable alternative to fixtured material transfer schemes such as conveyor belts. The mobile robot we work on at Oxford is functionally equivalent to a fielded commercial system that maintains a software model of the layout of a factory, and determines its position by retroreflecting infrared illumination from bar-coded targets placed in accurately known locations in the environment. Using this data, the mobile robot can calculate its position to better than a centimetre over 40 m, and its orientation to a few degrees. In most practical applications this is more than sufficient. Paths are planned using the software model of the factory. However, the fielded mobile robot cannot detect unexpected obstacles and plan detours around them, nor can it recognize objects to be acquired, for example pallets from a loading bay. These two problems have been the foci of work at Oxford. Successful systems have been developed using sonar sensing for obstacle detection (Hu *et al.* 1991) and a laser range scanner for recognising objects such as palletised goods that vary parametrically (Reid & Brady 1992). It is far from obvious that the additional power offered by visual guidance of vehicles is necessary for many industrial purposes, particularly inside a factory, while the computational cost and unreliability of visual processing currently makes it unattractive to production engineers. Outdoor applications are, however, a different proposition, and vision may well be put to cost-effective, practical use in applications such as automating work in stockyards, mining, agriculture, and for a number of military purposes.

In the next section we argue that the information provided at the different locations in an image provide vastly different local information, hence differing constraint. This is most obviously the motivation for edge detection; but applies equally to the detection, as primitives, of distinctive feature points, which, rather loosely, we call 'corners'. Section 3 presents a number of techniques for computing such feature points robustly and reliably, and shows they may be used for segmenting an optic flow field. Section 4 shows two approaches to computing scene structure from the motion of such feature points. In each case, the depth is computed explicitly at each feature point, then a depth map for the environment is interpolated from the discrete set of values. To do even this in real time requires a powerful parallel computer architecture; the one that has been assembled in Oxford is sketched in § 5. In § 6, we discuss the problem of propagating constraint from 'corner' feature points along contours defined by significant intensity changes ('edges'), a process that refines Hildreth's influential early work (Hildreth 1984) on edge-based optic flow. Finally, in § 7, we ask whether depth does after all need to be made explicit, present some work that shows that information that is useful for navigation and collision avoidance can be computed without precise camera calibration, and say why such a scheme is to be preferred.

2. THE INEQUALITY OF INFORMATION

Not all pixels in an image carry the same amount of information, even as a constant signal carries no information. Our ability as humans to readily interpret line drawings made from images, suggest that an

Phil. Trans. R. Soc. Lond. B (1992) **337,** 341–350
Printed in Great Britain

important early processing step is to extract intensity changes, a suggestion reinforced by physiological studies of simple cells in visual cortex. Indeed, under certain weak conditions, an image can be reconstructed from its intensity changes. A (static) image can be regarded as a surface, in which case the extraction of intensity changes amounts to developing numerical approximations to the gradient ∇I and image (local) surface curvatures. Several approximations to first and higher order derivatives of the image intensity function have been proposed typically preceded by convolving with a smoothing filter such as a Gaussian G_σ. A number of edge detection schemes have been proposed, extensively evaluated, and implemented in real time, for example using recursive filters (Canny 1986; Deriche 1987). The Canny filter, for example, serves to emphasize the importance of nonlinear filtering. Over the past five years three novel nonlinear filters have been developed: Fleck's (1988) application of finite-resolution cell complexes to compute high frequency texture edges; Owens & Venkatesh's (1989) local energy model based on Hilbert quadrature pairs and local phase; and morphological filtering.

3. COMPUTING 'CORNERS'

'Corners' carry even more information than edges, though they are more difficult to extract accurately and reliably. In the next section, we describe a system that computes structure from motion by tracking corners over a sequence of images, and in § 6 we use corners as seed points in an algorithm to compute optic flow. We first address the question of how corners can be modeled and extracted. One approach considers a corner to be a point on an intensity change contour (edge) where the curvature along the edge attains a maximum. More formally, let $\boldsymbol{n}$ be the unit vector in the direction of the intensity gradient. The edge curvature $\kappa_{\boldsymbol{n}\perp}$ can be measured by the image surface normal curvature in the tangential direction orthogonal to $\boldsymbol{n}$. Elementary differential geometry shows that:

$$\kappa_{\boldsymbol{n}\perp} = \frac{1}{1 + I_x^2 + I_y^2}\left(\frac{I_y^2 I_{xx} + I_x^2 I_{yy} - 2I_x I_y I_{xy}}{I_x^2 + I_y^2}\right).$$

This formula was discretized and used in an early corner finder by Kitchen & Rosenfeld (1982). A later implementation by Zuniga & Haralick (1983) worked out the expression analytically for the coefficients of a bicubic polynomial fit locally to the image surface. Both implementations were highly sensitive to noise. Recently, the authors introduced a novel algorithm of this sort, which we review following a discussion of an alternative formulation for a corner.

The most attractive alternative model is to identify corners with pixels whose intensities are significantly different from those surrounding neighbouring pixels. One way to do this is to use the local image autocorrelation function; but this tends to place corners at some remove from the ideal location, and although this does not affect tracking, it does affect the estimation of three-dimensional depth. A recent variant to this theme is the Harris corner finder (Harris & Pike 1987), which has been used in the DROID system discussed below. The algorithm consists of three steps.

1. Compute the partial derivatives of the image I_x and I_y, and compute the Gaussian smoothed products:

$$\langle I_x^2\rangle, \quad \langle I_y^2\rangle, \quad \text{and} \quad \langle I_x I_y\rangle.$$

2. Compute the eigenvalues λ_1, λ_2 of the matrix

$$\begin{bmatrix} \langle I_x^2\rangle & \langle I_x I_y\rangle \\ \langle I_x I_y\rangle & \langle I_y^2\rangle \end{bmatrix}$$

3. Mark points as corners where the product $\lambda_1\lambda_2$ divided by the sum $\lambda_1 + \lambda_2$ is greater than a threshold.

Noble's (1989) thesis presents a Taylor's series analysis of the grey level function $I(x,y)$ in the vicinity of a Harris corner. It shows that the Harris algorithm combines linearly an estimate of the intensity gradient and an estimate of the image curvature. Both need to be large for a pixel to be marked by Harris' algorithm.

Recently, Wang & Brady (1991) developed an algorithm that estimates the tangential curvature $\kappa_{\boldsymbol{n}\perp}$ using the linear interpolation and non-maximum suppression techniques used originally in Canny's edge finding algorithm. Refer to figure 1. Suppose that the image gradient $|\nabla I|$ is above a threshold at the point C shown, and let the edge tangent $\boldsymbol{t} = \boldsymbol{n}\perp$ be in the direction shown. The intensity function I is interpolated at locations I_B, I_b, I_C, I_f, I_F, and the second derivative $D_{\boldsymbol{t}}^2 = \kappa_{\boldsymbol{n}\perp}$ is estimated from the finite difference approximation:

$$D_{\boldsymbol{t}}^2 I = \frac{1}{r}(-2I_B - I_b + 6I_C + I_f + 2I_F),$$

where r varies spatially, e.g. $r = \nabla I / I_x$ when the tangential direction is less than $\pi/4$. Corner points are marked where (i) $D_{\boldsymbol{t}}^2 I > \omega|\nabla I|$, and (ii) non-maxima are suppressed. Figure 2 shows one synthetic image and the corners found by Wang & Brady's corner finder. The corner finder has been implemented on the parallel architecture PARADOX, described in § 5, and computes corners at 14 Hz. The Wang & Brady algorithm exhibits interesting scale space behaviour,

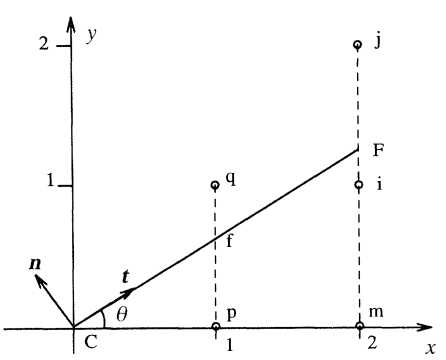


Figure 1. Linear interpolation for intermediate pixel addressing.

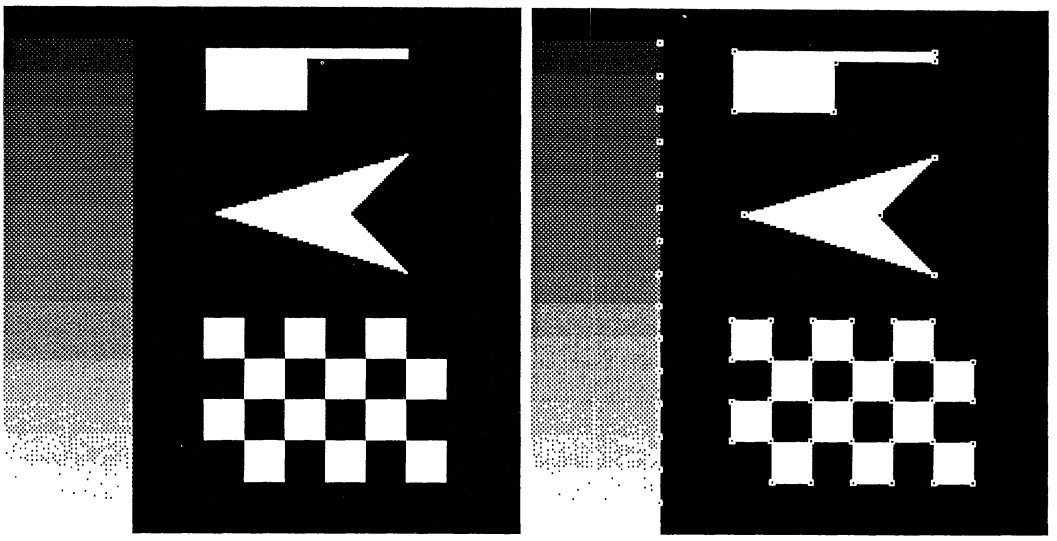


Figure 2. Left, a synthetic image; right, corners located by Wang & Brady's corner finder.

reminiscent of the early algorithm of Asada & Brady (1986); see Wang & Brady (1992) for further details.

The quest for feature-finding algorithms continues. Since Wang & Brady's algorithm was invented, Smith (1992) has devised a scheme that is a hybrid of an autocorrelation and morphological filter. It has been used to generate a fast, though approximate, segmentation of the moving objects in motion sequences. Torr *et al.* (1991) report initial experiments with a system that consists of a number of such modules each of limited scope but which are combined in a robust motion estimation system.

4. COMPUTING SCENE STRUCTURE USING CORNERS

This section reports two schemes for determining scene structure using the feature points computed for each image in a sequence by the algorithms described in the previous section. First, we review the structure from motion algorithm DROID originally developed by Harris & Pike (1987). The authors have worked on an Esprit grant VOILA with the DROID team to develop a revised version to run on the parallel architecture PARADOX. Second, we describe an algorithm based on disparity gradient to match the feature points in a stereo pair of images.

DROID determines the three-dimensional structure of a scene by tracking image features over a sequence of frames. It works as follows.

1. The state of the system at time t consists of a set of feature points, each of which has an associated vector $[x_i(t)y_i(t)z_i(t)\varepsilon_x^i(t)\varepsilon_y^i(t)\varepsilon_z^i(t)\lambda_i]^{\mathrm{T}}$. The first three components are the estimated three-dimensional position of the feature. The second three components are the semi-axes of an ellipsoidal representation of uncertainty. Initially, the uncertainty is low in the image coordinates, but large in depth. As the vehicle circumnavigates a feature (assuming that it does), the uncertainty in depth reduces by intersecting ellipsoids. The final component λ is a flag that says whether the feature is temporarily invisible (e.g. because it is occluded).
2. Given an estimate of the motion between the imaging position at time t and time $t+1$, the state prediction at time $t+1$ can be made, and, after perspective projection using a pinhole model, the positions of the feature points in the $t+1$st image can be predicted.
3. Features can be extracted from the $t+1$st image. Originally, features computed using the Harris algorithm were used. Now we use the Wang-Brady or Smith algorithms. (This is by far the most computationally intensive step of the computation.)
4. The correspondence is computed between the predicted locations of points in step 2 and the feature points found in step 3. Ambiguous matches are decided by matching descriptions of the predicted and found points that involve the intensity value and gradients.
5. The state estimate is updated using a Kalman filter algorithm, and the algorithm loops to step 2.
6. Optionally, at each loop, the image locations of the feature points are triangulated, using the Delaunay triangulation that minimises long thin triangles (Floriani 1987). The triangles are deprojected into the scene (recall that the z component of each feature point has been estimated). The surface normal of each deprojected triangular facet is computed. Driveable regions are computed as the transitive closure of the facets lying close to the horizontal plane, starting just in front of the vehicle.

Wang *et al.* (1992) report a number of careful experiments with the DROID algorithm using cameras mounted on the robot vehicle described in the first section and which can compute its position using the on-board infrared sensor. There is no integral action in the controller, but subject to this limitation, velocity and position estimates are resonably accurate.

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY B BIOLOGICAL SCIENCES

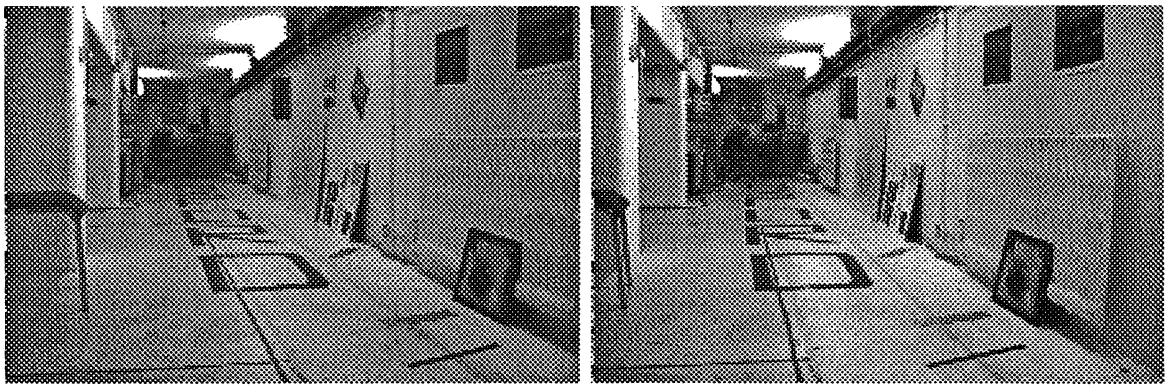


Figure 3. Stereo images of the laboratory as seen from cameras mounted on the mobile robot.

However, precise camera calibration is required, and, in view of the inevitable vibrations of the mobile robot, this is the most sensitive part of the system. We return to this point below; but note in passing that the system has more in common with photogrammetry than human vision.

The authors have explored an alternative method for computing scene structure by stereo matching the corner feature points in a pair of images (Wang & Brady 1992). (We have also integrated the stereo algorithm and the structure from motion algorithm DROID.) The stereo algorithm consists of two steps: correlation matching; followed by a verification step that enforces the disparity gradient limit (DGL) constraint (Pollard *et al.* 1985), uncovered for human stereopsis (Burt & Julesz 1980).

1. Matching. The match operation proceeds in two passes. First, regions of a fixed size are identified in the left and right images surrounding each feature point. For each feature point in the left image, a set of candidate matching feature points in the right image are determined to satisfy the epipolar constraint. For each feature point in the left image, the right image feature point whose surrounding region maximises the normalized cross correlation with that of the left image point (see Wang *et al.* (1990) for more details) is noted. This procedure is repeated for matches from right to left and pairs that maximize each other are recorded as candidate matches.
2. Verification. The DGL is used to confirm matches. For point pairs that are in sparse regions of the image, the disparity gradient is computed as the difference in disparity divided by the Euclidean distance. The mean of the local disparity gradient provides a statistical measure of the local support for a match.

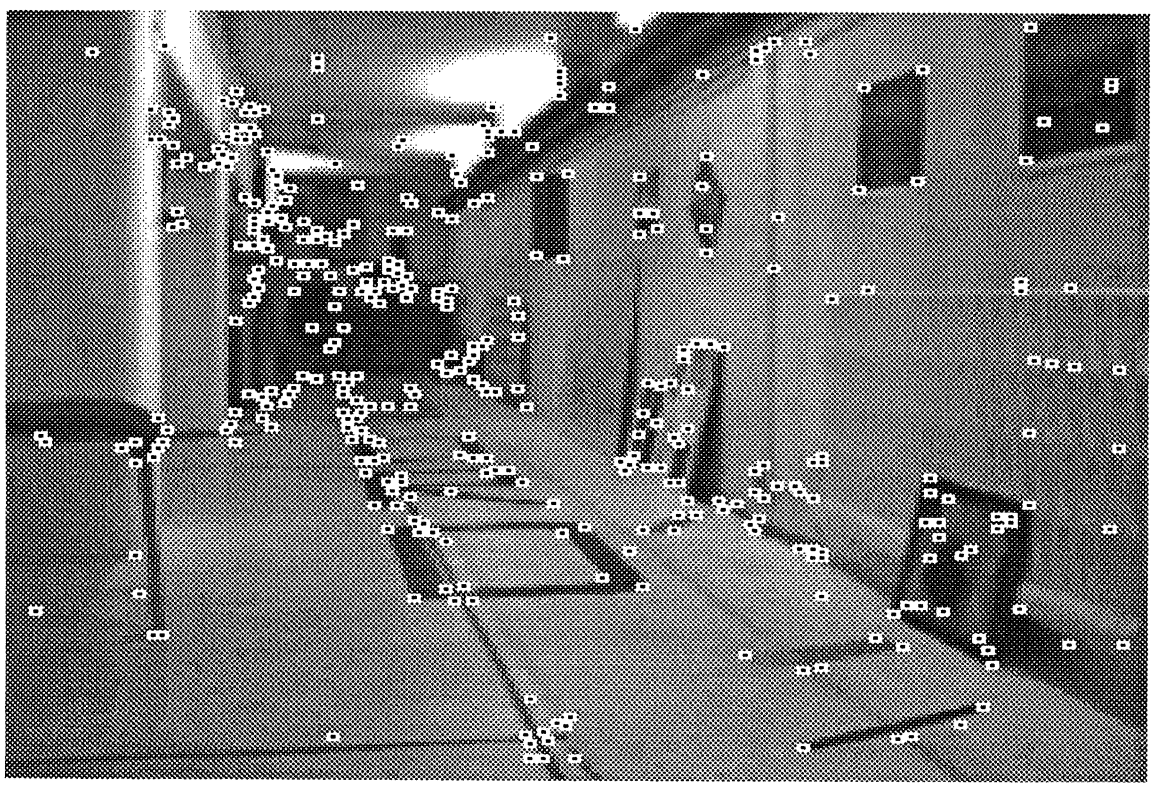


Figure 4. Corners detected in the left image of the stereo pair.

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY B BIOLOGICAL SCIENCES

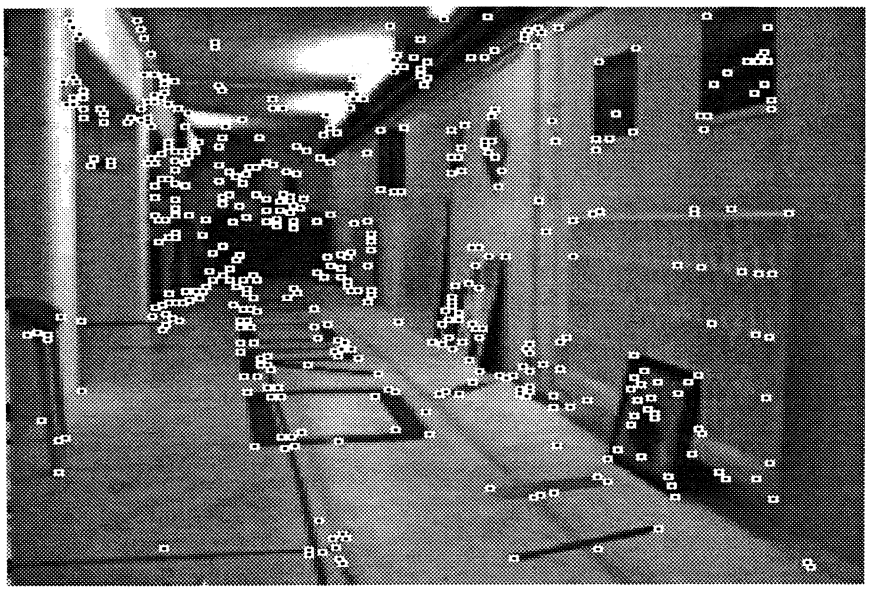


Figure 5. Corners detected in the right image of the stereo pair.

Figure 3 shows a stereo image pair of the laboratory as seen by a pair of CCD cameras mounted on the mobile robot. The cameras were carefully calibrated. Figures 4 and 5 show the corner features detected in the left and right images (they are computed to subpixel accuracy).

Figure 6 shows the corners matched in the stereo pair and indicates the disparities computed for each. Finally, figure 7 shows the three-dimensional surface reconstructed from the sparse disparity values. Thacker has also reported an approach of matching corners using correlation scheme, although there is not a verification stage for discarding ambiguous matches on noise data (Thacker 1990).

To summarize this section: it is possible to compute the depths from images to the scene points that give rise to image feature points using structure from motion or stereo. The algorithms rely on precise camera calibration. They more closely resemble photogrammetric mapping procedures than human stereopsis or motion parallax. It is not clear that the accurate depth maps that result from such algorithms are in fact required for successful navigation by a mobile robot (or animal).

5. REAL-TIME ARCHITECTURE

PARADOX is a hybrid architecture, designed and configured especially for vision- and image-processing algorithms. It consists of three major functional parts: a commercial (Datacube) pipelined image-processing system, a MIMD network of Transputers, and a

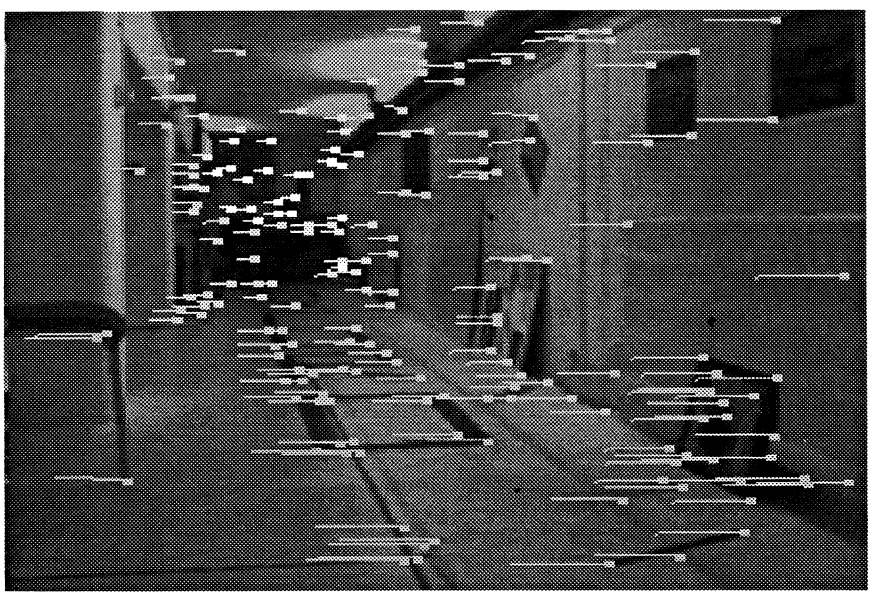


Figure 6. Matched corner pairs are shown as the disparity vectors. The cameras have been calibrated so that epipolars align with camera scan lines, so that disparity, a vector valued quantity, can be given by a scalar value.

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY B BIOLOGICAL SCIENCES

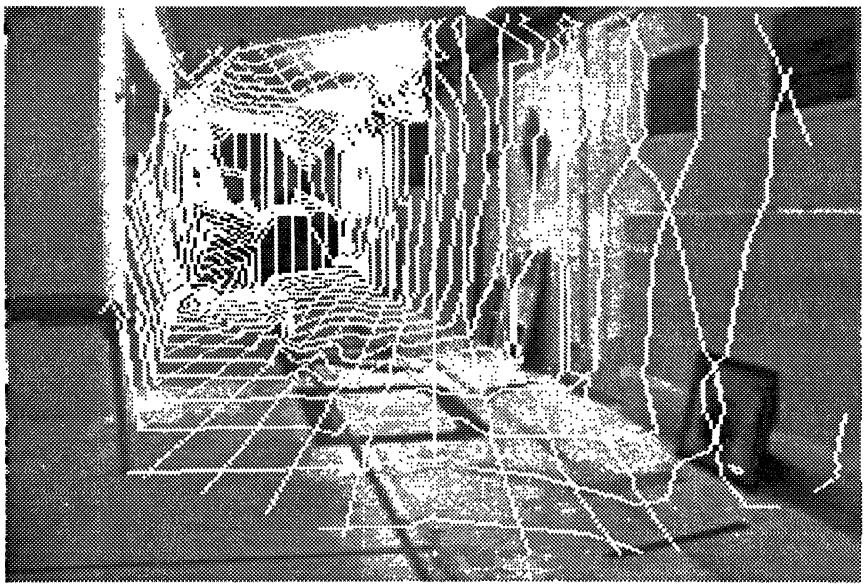


Figure 7. The three-dimensional surface reconstructed from the sparse disparity information.

controlling workstation. The architecture of PARADOX, used in the current parallel implementation of DROID, is shown in figure 8.

The commercial system contains more than 20 types of VME-based pipelined processing and input–output modules which can perform a range of image processing operations at video rates. Image data is passed between modules via a video bus. This system can be used for the digitization, storage and display of images, and also for a wide range of video frame-rate pixel-based processing operations. For example, convolution and morphological operations can be supported directly in the hardware. System control is by means of the VME bus from the controlling workstation.

The MIMD network consists of a board that contains 32 T800 Transputers, each with one Mbyte of RAM, together with both hardwired and programmable switching devices to allow the network topology to be altered. A wide range of network topologies can be implemented including parallel one-dimensional arrays (Wang *et al.* 1991), a two-dimensional array or a ring structure. This board delivers a peak performance of 320 MIPS. The connection between the image processing system and the Transputer network is by way of an interface board designed by the British Aerospace Sowerby Research Centre (Sheen 1988).

In the parallel implementation of DROID (Wang & Bowman 1991), the Datacube is used to digitize, store and display the image sequence and graphics overlays; corner detection is carried out by the Transputer array; whereas the relatively undemanding structure from motion filtering is carried out on the workstation.

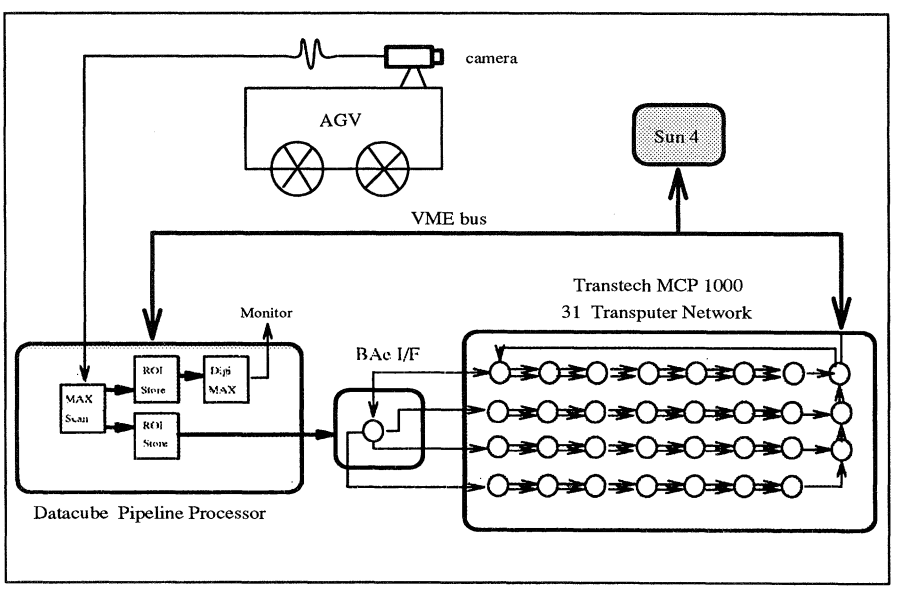


Figure 8. Machine architecture PARADOX (DROID incarnation).

It is quite straightforward to do the filtering on the Transputer network.

6. PROPAGATING CONSTRAINT FOR OPTIC FLOW

In this section we review the work of Brady, Gong, and Wang aimed at improving in certain respects Hildreth's edge-based optic flow algorithm, whose salient points we review. In our variant, we compute the full optic flow at corners lying on edges, then propagate the flow along the edge between such locations using a mixed wave–diffusion process.

As a viewer moves relative to the scene, the image brightness pattern changes. More precisely, let $I(x,y,t)$ denote the image at time t. Truncating to first order a Taylor's series approximation to the expansion of the image function at time and position $I(x+\delta x, y+\delta y, t+\delta t)$, which effectively amounts to assuming a constant brightness distribution locally, leads to the motion constraint equation (Horn 1986), given by:

$$\boldsymbol{n}\cdot\boldsymbol{\mu} = \frac{-I_t}{||\nabla I||},$$

where $\boldsymbol{n}$ is a unit vector in the direction of the image gradient $\boldsymbol{n} = \nabla I/||\nabla I||$, and $\boldsymbol{\mu} = [\dot{x}\ \dot{y}]^{\mathrm{T}}$ is the optic flow. This equation shows that so far as a first order analysis is concerned, only the component of optic flow in the direction of the image gradient can be computed, an observation that is known as the aperture problem.

Hildreth (Hildreth 1984) proposed to compute the optic flow only at edge points of an object. The underlying reason is that the magnitude of the gradient of the image function $||\nabla I||$ appears in the denominator of the motion constraint equation, hence the result will be poorly conditioned if this quantity is small. But large $||\nabla I||$ corresponds to edge points. Hildreth suggested a variational principle that attempted to estimate the optic flow $\boldsymbol{\mu}(s)$ along a (closed zero-crossing) image contour $\boldsymbol{r}(s)$ given the local (aperture problem) estimates of the normal component $\boldsymbol{\mu}^{\perp}(s)$. More precisely, Hildreth suggested computing the optic flow field $\boldsymbol{\mu}(s)$ that minimizes:

$$\oint \mathrm{d}s \left(\frac{\partial \boldsymbol{\mu}_x}{\partial s}\right)^2 + \left(\frac{\partial \boldsymbol{\mu}_y}{\partial s}\right)^2 + \lambda(\boldsymbol{\mu}(s)\cdot\boldsymbol{n}(s) - \boldsymbol{\mu}^{\perp}(s))^2.$$

The first two of these terms enforces a smooth flow field, the latter enforces agreement with the data. Hildreth tested her algorithm primarily on simulated data, for which she demonstrated remarkable agreement with psychophysical observations. We may say that Hildreth's work represents an important step in understanding human visual motion competence, but her algorithm certainly is not practical, for two major reasons.

The first is that her algorithm is inevitably slow even if it is implemented on a parallel computer (she did not consider how this might be done). The reason is that the optic flow value finally computed at a location s_1 along the curve depends both in principle and in practice on the values computed (including the initial value) at every other curve location s_2. In fact, if the curve has N points, then it takes $\mathcal{O}(N^2)$ iterations to converge. Because in many practical cases $\mathrm{N} \approx 1000$, this can be significant. The solution to this problem is somehow to reduce N. Nagel's early work suggested (Nagel 1987) that optic flow might be computed at image locations that he called 'image corners', though his mathematical definition of a corner location can be challenged (see Noble's thesis).

The second reason is that the edge points suggested by Hildreth are the zero-crossings of a Laplacian of a Gaussian filtered version of the image: $\nabla^2 G_\sigma * I$, which is often approximated by the difference of Gaussian (blurred) versions of the image. A feature of this set of putative edge points is that they form closed image contours. If image edges were always two-connected, this would in principle be satisfactory; but it simply cannot cope with multiway junctions typified by occlusion. If the wrong occlusion interpretation is made by the edge filter (and it often is in practice), then inevitably the optic flow estimates of (say) a moving object and static background are interpreted as a single moving object and the results intermingle and degrade badly. The solution is only partly alleviated by moving to better edge detectors, such as that proposed by Deriche (Deriche 1987), since it requires that optic flow estimates be inhibited from being confused across two independently moving objects.

We have investigated a different approach in which the Taylor's series approximation to the image function is truncated to second order in the image variables and first order in time. This is not elegant mathematically, but reflects the massive difference in sampling in space and time of images. Gong & Brady (1990) observe that a second order expansion of the image function along a curve yields the constraint:

$$(\boldsymbol{t}^{\mathrm{T}} H \boldsymbol{n})(\boldsymbol{t}^{\mathrm{T}} H \boldsymbol{\mu} + \nabla I_t \cdot \boldsymbol{t}) = 0,$$

where $\boldsymbol{t}$ is the unit tangent vector to the curve, $\boldsymbol{n}$ the unit normal, and H the image Hessian matrix as used in Nagel's oreinted smoothness algorithm. Gong & Brady (1990) developed an algorithm that improves upon Hildreth's by minimizing

$$\oint \mathrm{d}s \left(\frac{\partial \boldsymbol{\mu}_x}{\partial s}\right)^2 + \left(\frac{\partial \boldsymbol{\mu}_y}{\partial s}\right)^2 + \lambda(\boldsymbol{\mu}(s)\cdot\boldsymbol{n}(s) - \boldsymbol{\mu}^{\perp}(s))^2 + \frac{\det H}{\varepsilon}(\boldsymbol{t}\cdot\boldsymbol{\mu} - \boldsymbol{\mu}^{\mathrm{T}})^2,$$

where $\varepsilon = \det H \det H^{-1}$.

The effect of this is to reduce substantially the computation time of Hildreth's algorithm by reducing the distance over which information has to be propagated. However, it still turns out to be too slow for real-time implementation.

In essence, the problem is as follows. Initially, the full optic flow $\boldsymbol{\mu}(s_1)$ and $\boldsymbol{\mu}(s_2)$ is computed at successive corners along a piecewise smooth contour, and the normal component $\boldsymbol{\mu}^{\perp}(s), s_1 \leqslant s \leqslant s_2$ is computed in between. The difficulty is to propagate efficiently the additional constraint provided at the end points along the curve. Yuille (1988) proposed a 'motion coherence algorithm', which essentially amounts to a diffusion

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY B BIOLOGICAL SCIENCES

Figure 9. Two images from a sequence where the head rotates about an axis passing through the neck and pointing forwards.

process which propagates the end point constraints at a rate that is proportional to the square root of the number of iterations, hence requires time proportional to the square of the distance separating the corners. An alternative scheme, introduced for shape analysis by Scott *et al.* (1988) of Oxford, combines a diffusion with a wave process.

More precisely, a corner constraint is propagated in both directions from the corner at a constant speed, with half the strength in each direction. The wave transmits the tangential component of the flow, but does not conserve its value. To reduce the sensitivity to the velocity measure, we smooth sharp changes using a Gaussian (diffusion). The coupled diffusion process reduces its value. Clearly, contour points near a corner are more strongly influenced by the value of the tangenial component at the nearby corner than they are by the value at the adjacent corner point further away. The algorithm runs in time proportional to the arclength separating adjacent corners, which, given the restriction to propagation along the contour, is hard to improve on. Subsequent work by Wang *et al.* (1990) has described a parallel implementation of this process on a network of Transputers. Figure 9 shows a pair of images from a sequence (from an investigation of model-based image coding) in which the head rotates in the plane of the shoulder blades from the right shoulder towards the left. Figure 10 shows the optic flow computed by the Gong–Brady–Wang algorithm.

Figure 10. Propagated flow.

7. MUST WE COMPUTE DEPTH EXPLICITLY?

An important step in the development of machine vision was the demonstration that three-dimensional scene layout could be computed by processes such as stereo, structure from motion, or shape from shading, contour, or texture. An analysis of the viewing geometry in a small number of images (e.g. two in the case of stereo) led, via camera calibration, to a sparse depth map, and then, via surface interpolation, to a dense depth map. The quality of the resulting depth map was evaluated in a number of ways, most stringently by requiring robot arms to pick and place objects in the field of view. Many recent systems have exploited parallel hardware simply to execute more quickly multistep algorithms devised originally for serial processors. More interestingly, parallel hardware enables more images to be processed; a system can always take another look, and can close the loop between sensing and action.

However, closed loop control is no panacea. A key issue concerns the choice of control variable, and this straightway leads back to calibration, which we earlier identified as the Achilles' heel of three-dimensional vision systems. Recall, for example, the error ellipsoids used in the DROID system. The errors in the image plane directions are small, since corner feature points can be localised accurately; but the error in depth is large and circumnavigation of the corresponding scene point is required to give accurate depth values. More generally, the distance to a point in the scene is related nonlinearly to image quantities (observables) such as disparity that can be computed accurately. The nonlinearity corresponds to the calibration parameters, which are notoriously difficult to estimate accurately. From a control theoretic standpoint, filtering on depth involves a measurement equation that is a linear approximation to the hard-to-estimate nonlinear imaging geometry, a situation that mitigates against robustness.

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY B BIOLOGICAL SCIENCES

Is there an alternative? Can we, for example, compute information that is useful to a mobile robot directly from observables such as optic flow, disparity, and rate of change of disparity (or invariant quantities computed from these measures), and altogether avoid camera calibration. Recent work at Oxford, particularly by Cipolla (1992) and Blake & Cipolla (1990), suggests that we can. First, Blake & Cipolla show formally that if an observer makes deliberate motions relative to a scene containing a planar closed curve, then the surface normal of the plane containing the curve can be estimated from the symmetric component of the deformation of the image of the curve, and the time to contact with the curve can be estimated from the divergence. Algorithms based on moments of the sequence of image curves are developed. Second, but again making deliberate motions, an observer can distinguish bounding contours of an object (where the surface normal turns smoothly away from the viewer) from edges fixed in space (for example surface creases or reflectance changes). Furthermore, the surface curvature along a bounding contour can be estimated accurately from the relative motions of surface features. Blake *et al.* (1991) discussed how this might be used by a moving observer to navigate among smooth objects, and the scheme has been refined and implemented by Blake *et al.* (1992).

In our current work we are investigating this proposition further as we are developing an algorithm that, without knowing the motion of the observer, groups coplanar sets of features points, then determines the orientation relative to the observer of each planar group, and computes the time to contact it. A test for coplanarity based on projective invariants has been developed.

8. CONCLUSIONS

Edge and 'corner' feature points can be computed rapidly and reliably in images, and they are stable over image sequences. Corner points provide more constraint than edge points, which themselves provide more constraint than image locations where the intensity varies slowly. Constraint can be propagated effectively from corners along edges and from edges to the interior of the region that it bounds. Algorithms to compute optic flow and scene structure using stereo or structure from motion have been demonstrated. It seems that a mobile robot may not need to compute depth explicitly in order to navigate effectively, indeed there are good reasons why it should not.

Current work aims to develop a high-performance stereo platform (Murray *et al.* 1992) as part of an active vision system that uses low level visual information of the sort described in this paper to control the attention of the mobile robot.

We thank Andrew Blake, Bernard Buxton, Paul Beardsley, Roberto Cipolla, Olivier Faugeras, Chris Harris, Huosheng Hu, Steve Maybank, John Mayhew, Phil McLauchlan, David Murray, Dave Sinclair, Phil Torr, and Andrew Zisserman for helpful discussions in the work leading up to this paper. We particularly acknowledge collaborations with J.M.B.'s graduate students: Shaogang Gong, Alison Noble, Larry Shapiro, and Steve Smith. This research was supported by the SERC ACME Directorate and the EC under grants VOILA and FIRST.

REFERENCES

Asada, H. & Brady, J.M. 1986 The curvature primal sketch. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-8** (1), 2–14.

Blake, A. & Cipolla, R. 1990 Robust estimation of surface curvature from deformation of apparent contours. In *Proc. 1st European Conf. on Computer Vision* (ed. O. Faugeras), pp. 465–474. Berlin: Springer-Verlag.

Blake, A., Brady, J.M., Cipolla, R., Xie, Z. & Zisserman, A. 1991 Visual navigation around curved objects. In *Proc. IEEE Int. Conf. Robotics and Automation.*

Blake, A., Curwen, R. & Zisserman, A. 1992 Towards visually guided object manipulation. In *Proc. 2nd Euro. Conf. on Computer Vision, Genova* (ed. G. Sandini). Springer Verlag.

Burt, P. & Julesz, B. 1980 A disparity gradient limit for binocular fusion. *Science, Wash.* **208,** 615–617.

Canny, J. 1986 A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-8** (6), 679–698.

Cipolla, R. 1992 Active visual inference of surface shape. Ph.D. thesis, University of Oxford.

Deriche, R. 1987 Using canny's criteria to derive a recursively implemented optimail edge detector. *Int. J. Comput. Vis.* 1 ((2)):167.

Fleck, M.M. 1988 Boundaries and topological algorithms. Ph.D. thesis, MIT Artificial Intelligence Laboratory.

De Floriani, L. 1987 Surface representation based on triangular grids. *Visual Computer* **3**, 27–50.

Gong, S. & Brady, J.M. 1990 Parallel computation of optic flow. In *Proc. 1st European Conf. on Computer Vision* (ed. O. Faugeras), pp. 124–133. Berlin: Springer-Verlag.

Harris, C.G. & Pike, J.M. 1987 3D positional integration from image sequences. In *Proc. 3rd Alvey Vison Conference, Cambridge.*

Hildreth, E.C. 1984 Computations underlying the measurement of visual motion. *Artif. Intell.* **23**, 309–354.

Horn, B.K.P. 1986 *Robot vision.* McGraw-Hill.

Hu, H., Brady, J.M. & Probert, P.J. 1991 Coping with uncertainty in control and planning for a mobile robot. In *Proc. of IEEE Int. Workshop on Intelligent Robots and Systems*, pp. 1025–1030. Osaka, Japan.

Kitchen, L. & Rosenfeld, A. 1982 Grey-level corner detection. *Pattern recogn. Lett.* **1**, 95–102.

Murray, D.W., Du, F., McLauchlan, P.F., Reid, I.D., Sharkey, P.M. & Brady, J.M. 1992 Design of stereo heads. In *Active vision* (ed. A. Blake & A. Yuille), pp. 155–172. Cambridge Massachusetts: MIT Press.

Nagel, H.H. 1987 On the estimation of optical flow: relations between different approaches and some new results. *Artif. Intell.* **33**, 299–324.

Noble, J.A. 1989 Description of image surface. Ph.D. thesis, University of Oxford.

Owens, R. & Venkatesh, S. 1989 A robust scheme for isolating and classifying features. In *Proc. of Int. Conf. Artificial Intelligence in Industry and Government*, pp. 664–676. Hyderabad, India.

Pollard, S.B., Mayhew, J.E.W. & Frisby, J.P. 1985 PMF: a stereo correspondence algorithm using a disparity gradient limit. *Perception* **14,** 449–470.

Reid, I. & Brady, M. 1992 Model-based recognition and range imaging for a guided vehicle. *Image Vision Comput.*

Scott, G.L., Turner, S. & Zisserman, A. 1988 Using a

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY B BIOLOGICAL SCIENCES

mixed wave/diffusion process to elicit the symmetry set. In *Alvey Vision Conference*.

Sheen, J.A. 1988 A parallel architecture for machine vision. In *Colloquium on Practical applications of signal processing*. Institution of the Electrical Engineers. Digest no: 1988/111.

Smith, S. 1992 *A new class of corner finder*. Technical Report, Department of Engineering Science, Oxford University.

Thacker, N.A. 1990 *Corner detection and matching*. Technical Report AIVRU 58, University of Sheffield.

Torr, P.H.S., Wong, T., Murray, D.W. & Zisserman, A. 1991 Cooperating motion processes. In *Proceedings of the 2nd British Machine Vision Conference* (ed. P. Mowforth), pp. 145–150. Glasgow: Springer-Verlag.

Wang, H. & Bowman, C.C. 1991 The Oxford distributed machine for 3D vision system. In *IEE colloquium on Parallel Architectures for Image Processing Applications*, pp. 1/2–5/2. London.

Wang, H. & Brady, M. 1991 Corner detection for 3D vision using array processors. In *Bar-NAIMAGE 91*. Barcelona: Springer-Verlag.

Wang, H. & Brady, J.M. 1992 *Corner detection with subpixel accuracy*. Technical Report, Department of Engineering Science, Oxford University. OUEL No. 1925/92, submitted to CVGIP: Image Understanding.

Wang, H., Brady, J.M. & Page, I. 1990 A fast algorithm for computing optic flow and its implementation on a transputer array. In *Proc. of the British Machine Vision Conference* (ed. A. Zisserman), pp. 175–180. Oxford: Brit Machine Vis. Assoc.

Wang, H., Dew, P.M. & Webb, J.A. 1991 Implementation of Apply on a transputer array. *CONCURRENCY: Practice and Experience*, **3** (1), 43–54.

Wang, H., Bowman, C., Brady, J.M. & Harris, C. 1992 A parallel implementation of a structure-from-motion algorithm. In *Proc. ECCV'92, Second European Conference on Computer Vision* (ed. G. Sandini), pp. 272–276. Genova, Italy: Springer Verlag.

Yuille, A. 1988 A motion coherence theory. In *IEEE Inter. Conf. on Computer Vision*. Tampa, Florida.

Zuniga, O.A. & Haralick, R.M. 1983 Corner detection using the facet model. In *Proc. of IEEE Conference on Computer Vision & Pattern Recognition*, pp. 30–37. Washington DC.

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY B BIOLOGICAL SCIENCES

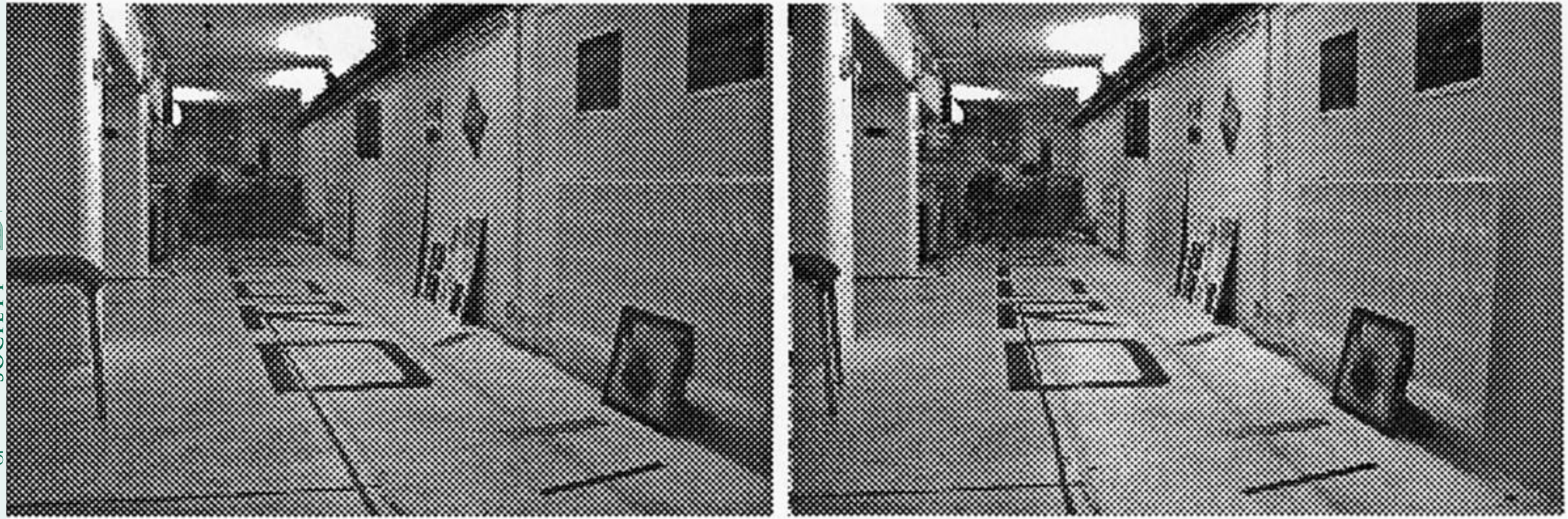


Figure 3. Stereo images of the laboratory as seen from cameras mounted on the mobile robot.

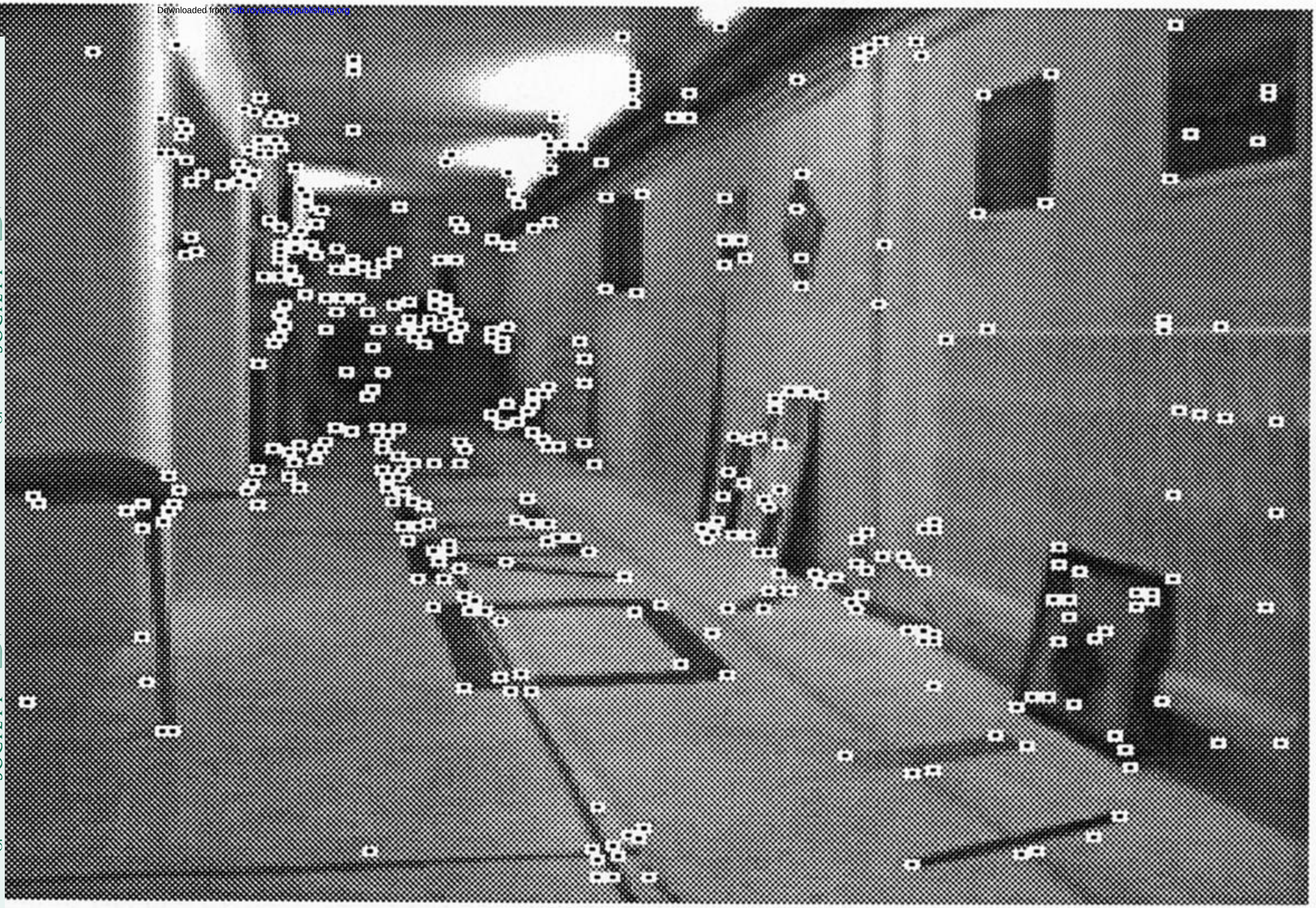


Figure 4. Corners detected in the left image of the stereo pair.

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY B BIOLOGICAL SCIENCES

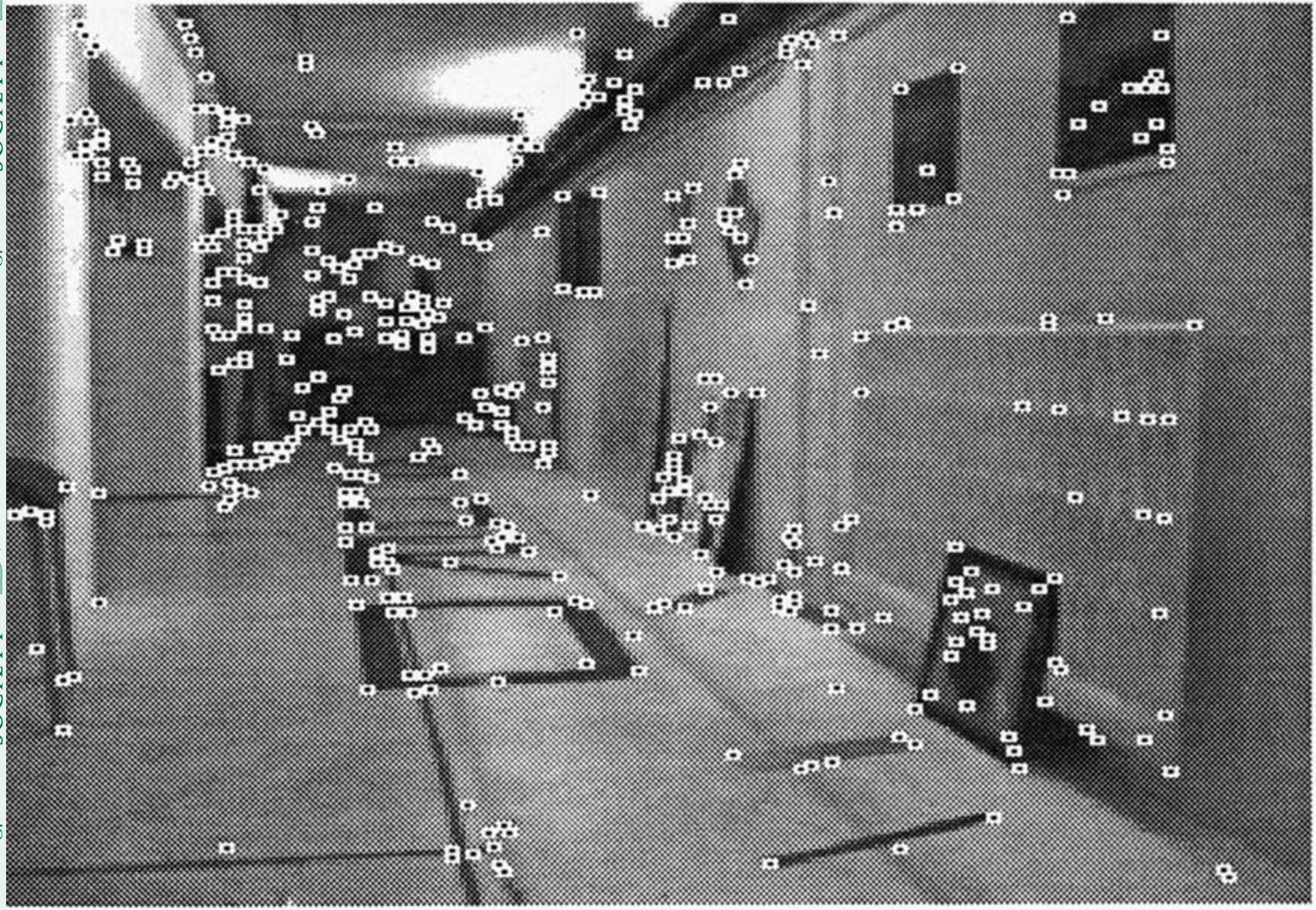


Figure 5. Corners detected in the right image of the stereo pair.

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY B

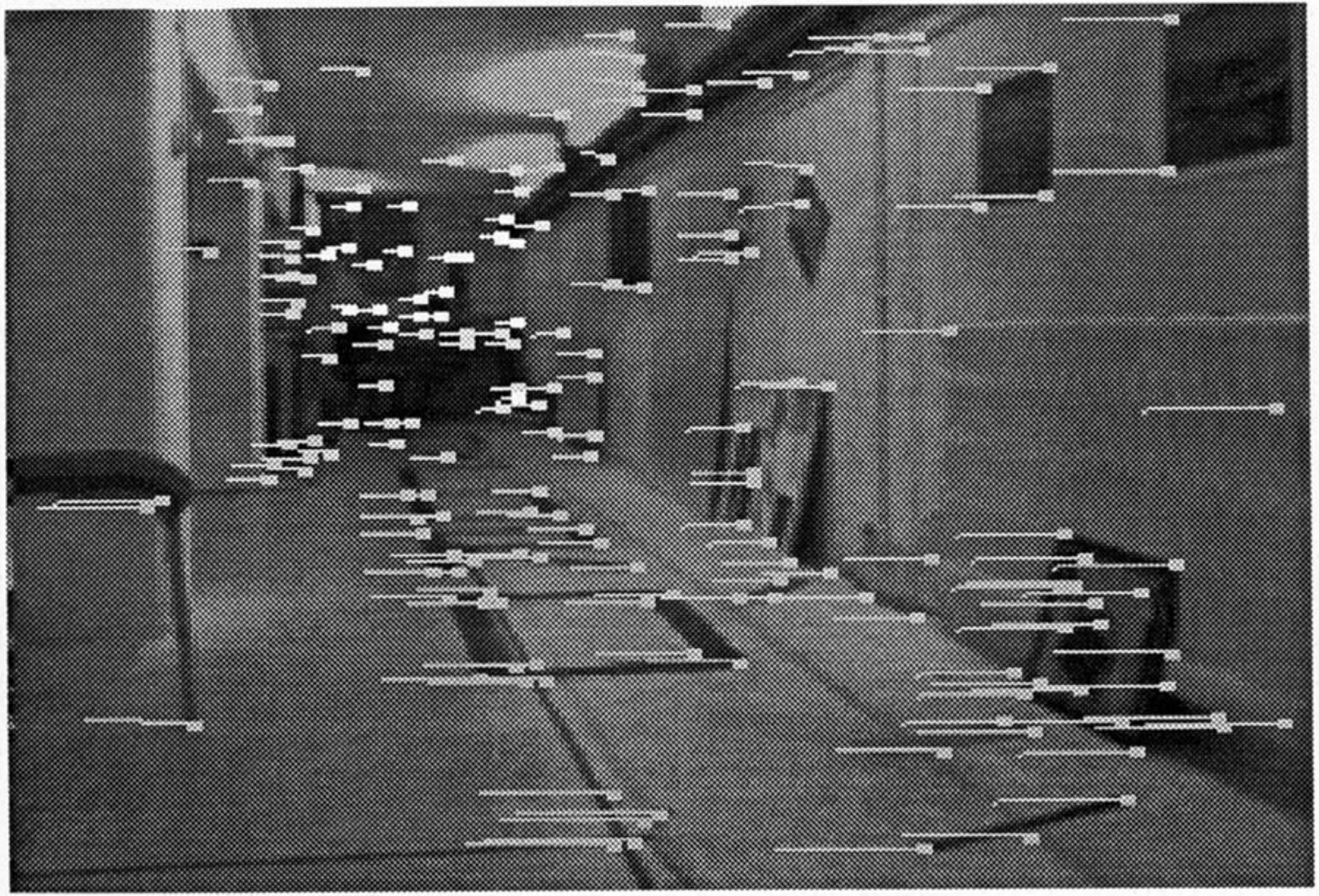
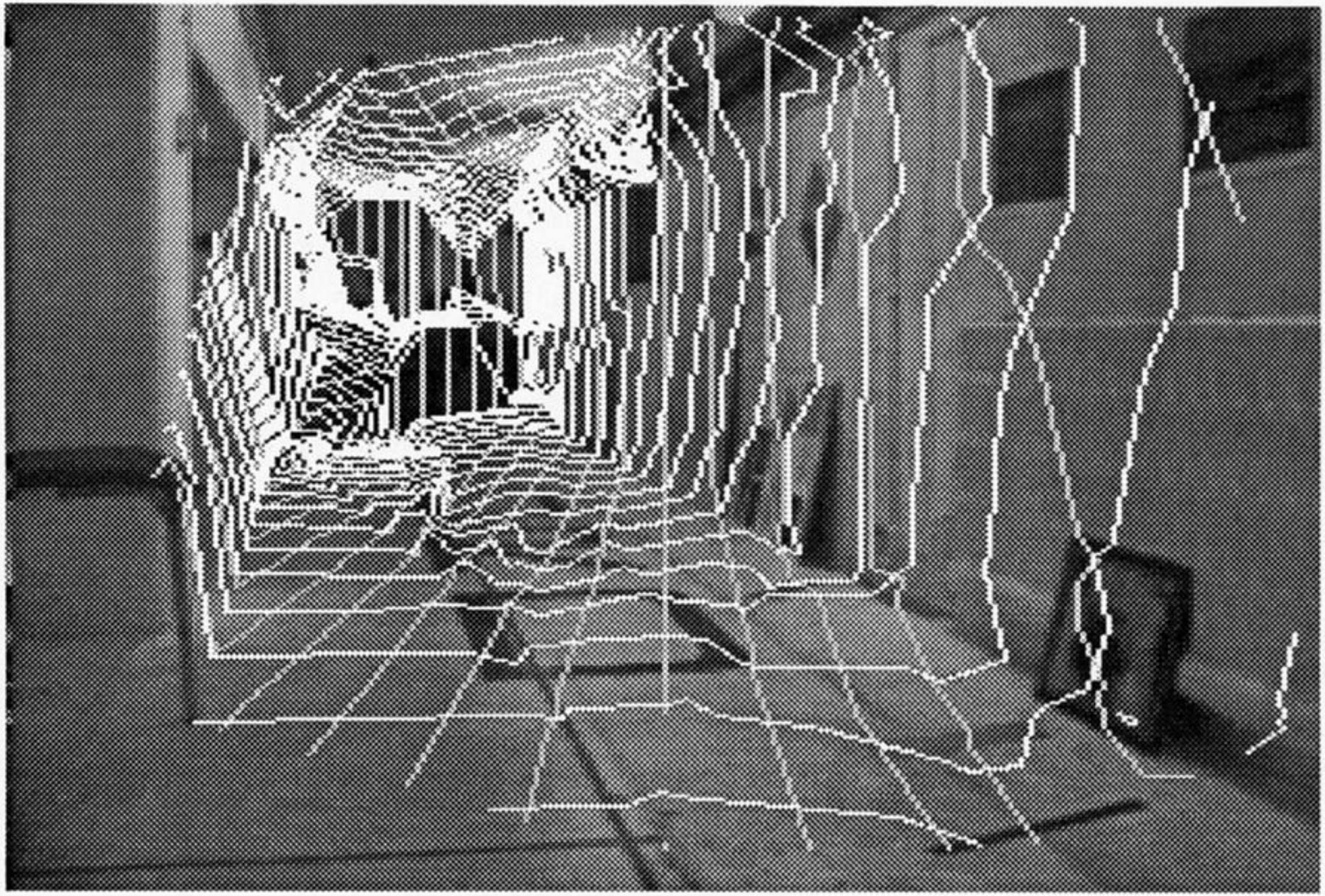


Figure 6. Matched corner pairs are shown as the disparity vectors. The cameras have been calibrated so that epipolars align with camera scan lines, so that disparity, a vector valued quantity, can be given by a scalar value.



igure 7. The three-dimensional surface reconstructed from the sparse disparity information.

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY B BIOLOGICAL SCIENCES

gure 9. Two images from a sequence where the head rotates about an axis passing through the neck and pointing forwards.