Bibliography

[1] Grosso, E. and G. Sandini and C. Frigato, 1988. 'Extraction of 3D Information and Volumetric Uncertainty from Multiple Stereo Images', Proc. of ECAI-88, Munchen, Germany.

[2] Baker H.H. 1982. 'Depth from edge and intensity based stereo', AI Memo No. 347, Stanford University.

[3] Nishiara H.K. 1984. 'PRISM: A practical real-time imaging stereo matcher', Opt. Eng., Vol. 23, No. 5, pp. 536-545.

[4] Marr D. and Poggio, T. 1976, 'A cooperative computation of stereo-disparity', Science, No. 194, pp. 283-287.

[5] Pollard B.P., Mayhew, J.E.W, and Frisby, J.P., 1985. 'PMF: A stereo correspondence algorithm using a disparity gradient limit', Perception, Vol 14, pp. 449-470.

[6] Grimson W.E.L. 1985, 'Computational Experiments with a Feature Based Stereo Algorithm', IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 7, No.1.

Chapter 5

Visual Motion and *VIS* $\vec{a}$ *VIS*.

Giulio Sandini, Massimo Tistarelli, and David Vernon

> Mathematics – let us introduce the refinement and rigor of mathematics into all sciences as far as this is at all possible, not in the faith that this will lead us to know things but in order to *determine* our human relation to things. Mathematics is merely the means for general and ultimate knowledge of man.
>
> *Friedrich Nietzsche*

5.1 Introduction.

The analysis of time-varying images can provide a great deal of information on the motion of viewed objects, on the segmentation of the image in different moving objects, on the $3D$ structure of the scene and on the egomotion of the viewing camera. Normally, an intermediate step is considered necessary. This is the determination of the $2D$ motion field, i.e., the perspective projection onto the image plane of the true $3D$ velocity field.

It is usually understood that local detectors, of any kind, can recover only one component of the $2D$ motion field, that which is parallel to the direction of the gradient of the image function. This notion, which can be found in many books and papers on machine vision, constitutes the essence of the *aperture problem*. It has been shown that the aperture problem is a false problem and that by simple local techniques it is possible to compute a dense $2D$ vector field (i.e. an optical flow) which is very close to the true $2D$ motion field. This is true if it is possible to characterise, locally, the second order (directional) variations in the grey level distribution of the images. One possible method to accomplish this is to track image contours in time. Edges in the image correspond to changes in the image brightness and therefore they convey information about the grey level distribution and its variation. The tracking of contour points on successive images of a sequence has much in common with the correspondence problem in stereo matching. The basic differences between the two computational problems arise from the fact that the motion of the camera can be actively controlled in case of egomotion whereas in stereo vision it is fixed by the stereo geometry (it is a virtual motion). For this reason motion can be analyzed by a *tracking* strategy while stereo must be based on *matching* procedures. The tracking strategy, which is possible in the case of egomotion, can be simplified by actively controlling the navigation parameters (this is equivalent to a reduction of dimensionality of the parameter space.) In fact from the knowledge of the egomotion parameters the two-dimensional direction of motion (i.e. the projection of the direction of motion on the image plane) can be computed uniquely. The amplitude of displacement (or the amplitude of the velocity vector) must be computed from the visual data. In order to increase the reliability of the measures the computation of displacement can be made only at the edge points.

In this chapter, the continuous nature of motion information is used to derive the amplitude component of the two-dimensional velocity vector perpendicular to the contours, while the knowledge of the egomotion parameters is used to compute its direction. By combining these measures an estimate of the displacement can be obtained for each image of the sequence. The value obtained is local in space, because it is computed at each edge point, and in time, because it is computed for each instant of time. In order to perform the tracking of image points this local procedure must be performed for each image of the sequence and each of the local values obtained must be linked to its predecessor and its successor in time. This is obtained by performing an explicit matching of the edges of successive images of the sequence, using the value of displacement computed locally, as an initial estimate for matching. This local matching procedure is re-iterated (obtaining the trajectories of each edge point) until the displacement of each point is considered sufficient for the accuracy of the final step of the algorithm, namely, the computation of depth (which is computed with a triangulation procedure whose accuracy is directly proportional to the length of the displacement vectors).

5.2 Camera motion and optical flow.

In our approach, the camera motion is controlled so as to track a point in the environment which lies on, or intersects, the optical axis of the camera (this is referred to as the *fixation point*). In the case of a static scene this is equivalent to stabilizing the fixation point during egomotion. It is worth noting that the sensor can perform any trajectory in space, with the advantage of reducing the dimensionality of the parameter space by one since the translation parameters of the camera are related to its rotation angles.

Without lack of generality we assume a null rotation around the optic axis (which coincides with the direction of gaze); in fact a rotation of the image around its center does not give any useful information about the scene.

In our opinion, the stabilization of the fixation point has at least two advantages: firstly, camera motion is simpler to control and secondly, the measurement of the parameters of the camera motion is easier than with other motion strategies.

The complexity of active control of camera motion in a mobile robot is, in fact, directly related to the constraints imposed on the motion pattern. For example, if the camera must be moved while keeping the optical axis parallel (i.e. a pure translation like the motion strategy proposed by Lawton), the control of movement must be based only on proprioceptive skills, or, alternatively, on the knowledge of a point in space at infinity. If this point is not available (for example in the case of indoor environments) or not known, the control must be based only on motor knowledge and cannot take advantage of the visual information. In case of a pure rotation, on the other hand, even if the control is much simpler, the visual information that can be computed from motion does not provide any information about depth: motion of the camera must be also translational. By using a control strategy similar to that adopted by the human visual system the constraints imposed on the trajectory of the camera are not so 'strong' as in the case of a pure translation but, nevertheless, can still provide information about depth. The visual feedback used to stabilize the fixation point (this procedure is similar to the tracking of a single image point or a small area of the image) requires the knowledge of a reference point (in our case the fixation point) but this is a general requirement to navigate in an unknown environment.

5.3 Computing optical flow and building the velocity image.

Regarding the computation of image displacement vectors, it is obvious that the matching of corresponding points on successive images is simpler if the images do not differ very much. On the other hand, if the successive images are only slightly different, the computation of the trajectory parameters, of image points, is subject to truncation errors. A way to optimize these two contrasting needs is to use a local procedure to match contour points on slightly different successive images, and a global one *linking* successive matched pairs to obtain a segment of trajectory long enough to compute the trajectory parameters with the appropriate accuracy.

The optical flow is computed, at the contour points, by means of the following steps:

- Computation of the velocity component perpendicular to the local orientation of the contour; the measurement of $V^{\perp}$ is performed at the zero crossing points[1].

[1]The contour extraction procedure is based on the theory of Marr and Hildreth on edge detection and it is performed by extracting the zero crossings in the image after the convolution with a Laplacian of Gaussian operator (see Chapter 3).

- Computation of the *true direction of motion.*
- Matching of corresponding contours of successive image pairs (i.e. computing the *instantaneous optic flow*).

These local computational steps are re-iterated over successive images and the instantaneous optical flows are linked to obtain the global optic flow relative to the entire sequence. At each step, an image data-structure containing velocity information, i.e. the velocity image outlined in Chapter 2, at each contour point is updated. In effect, the velocity vector amplitude and direction and other information used to link corresponding contour points over time. Contour descriptors are used to link contours in the same image.

The perpendicular component of velocity $V^{\perp}$ is computed as the ratio between the time derivative and the intensity gradient at the edge points:

$$V^{\perp} = -\frac{\frac{\partial I^g}{\partial t}}{|\nabla I^g|} \tag{5.1}$$

I^g represents a zero crossing point of the image I filtered with the $\nabla^2 G$ operator and $|\nabla I^g|$ is the edge slope. At this stage the velocity image is effectively built. For each contour point, the value of $V^{\perp}$ is stored in the amplitude field of the velocity image, while the inverse of the image gradient direction is stored in the phase field. It is worth noting that all possible orientations are quantized with eight-bit resolution, i.e., as one of 255 different values.

For the particular shape of the intensity edge filtered with the $\nabla^2 G$ operator, the maximum value of $V^{\perp}$ measurable with a given size of the mask is equal to half the width of the positive lobe of the mask (which is approximately equal to the standard deviation of the mask). Above this value $V^{\perp}$ will be underestimated. Assuming that the images are changing slowly in comparison with the sampling frequency, it is possible to use small values of σ, the standard deviation of the Gaussian function G in the $\nabla^2 G$ operator, for the filtering operator without loosing measurement accuracy. On the other hand, high resolving power requires a greater precision in the computation of the time derivative; for this reason the time derivative is computed using a five-point approximation formula applied to five frames of the filtered sequence of images.

5.3.1 Computing the true direction of motion.

In the general case of a *tracking observer* the translational velocity of the camera $\vec{W}$ can be computed from the parameters D_1, D_2, ϕ, θ and ψ:

$$W_X = \frac{D_2 \cos\phi \sin\theta}{\Delta t} \tag{5.2}$$

$$W_Y = \frac{D_2 \sin\phi}{\Delta t} \tag{5.3}$$

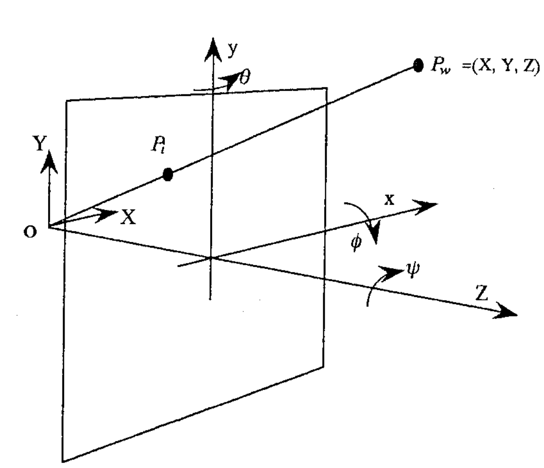

Figure 5.1: Coordinate reference frames for camera geometry

$$W_Z = \frac{D_1 - D_2 \cos\phi \cos\theta}{\Delta t} \tag{5.4}$$

D_1 and D_2 are the distances of the camera at time T and $T + \Delta t$ from the fixation point, respectively; ϕ is the rotational angle of the camera around the X axis; θ is the rotational angle of the camera around the Y axis and W_x, W_y and W_z are the components of the translational velocity of the camera along the X, Y and Z axes respectively, referred to the coordinate system of the camera at time T. The computed egomotion parameters are then used to determine the image velocity vector for every projected point on the camera.

Consider a reference world point in space P_w, with coordinates (X, Y, Z), as shown in in figure 5.1, defined with respect to the camera coordinate system **O**. Using the inverse perspective transform, the coordinates of the projected point P_i on the image plane are:

$$\vec{P_i} = (x, y) = (X, Y)\,\frac{F}{Z} \tag{5.5}$$

where F is the focal length of the camera measured in pixels. The apparent velocity of the point P_i on the image plane, due to the motion of the camera, is obtained differentiating (5.5) with respect to time. The resulting velocity vector can be decomposed into two parts: one due to camera translation $\vec{W} = (W_X, W_Y, W_Z)$ and the other to camera rotation $\vec{\omega} = (\omega_x, \omega_y, \omega_z)$

$$\vec{V_t} = \left(\frac{x\,W_Z - F\,W_X}{Z}, \frac{y\,W_Z - F\,W_Y}{Z} \right) \tag{5.6}$$

$$\vec{V_r} = \frac{x\,y\,\omega_x - \left[x^2 + F^2\right]\omega_y + y\,\omega_z}{F}, \quad \frac{\left[y^2 + F^2\right]\omega_x - x\,y\,\omega_y - x\,\omega_z}{F} \tag{5.7}$$

$$\vec{V} = \vec{V_t} + \vec{V_r} \tag{5.8}$$

where $\omega_x, \omega_y, \omega_z$ are the rotational velocities of the camera around the X, Y and Z axis respectively and W_x, W_y, W_z are the translational velocities of the camera in space. It is worth noting that from $\vec{V_t}$ it is possible to derive the position of the Focus Of Expansion (FOE) or the Focus of Contraction (FOC) on the image plane, which, in turn, depends upon the translational velocity of the camera $\vec{W}$ [2]:

$$FOE_x = \frac{W_X}{W_Z} F \tag{5.9}$$

$$FOE_y = \frac{W_Y}{W_Z} F \tag{5.10}$$

On the other hand, from the knowledge of the position of the FOE (FOC) on the image plane, which is determined through (5.9) and (5.10) if the translational velocity of the camera $\vec{W}$ can be computed, it is possible to determine only the direction of $\vec{V_t}$.

Because of the constraints posed on the motion of the camera, both rotation and translation must be considered; substituting the values of the translational velocity components of the camera $\vec{W}$ in (5.6), the expression of the velocity of point P_i on the image plane, due to camera translation, can be computed:

$$\vec{V_t} = \left(\frac{x\,[D_1 - D_2 \cos\phi \cos\theta] - F\,D_2 \cos\phi \sin\theta}{Z\,\Delta t}, \frac{y\,[D_1 - D_2 \cos\phi \cos\theta] - F\,D_2 \sin\phi}{Z\,\Delta t} \right) \tag{5.11}$$

Similarly, for the rotational component of velocity, if ω_x and ω_y are the rotation angles of the camera around the X and Y axes respectively (the rotation ω_z around

[2]Remember that the *focus* represents the intersection of an imaginary straight line, along the direction of translation and passing through the *convergence point* of the optical system, with the image plane. In particular we have a FOE if the camera is approaching the fixation point, and the velocity vectors are radiating from the focus, a FOC if the camera is moving away, and the velocity vectors collapse on it.

the optical axis is zero in our controlled motion) then $\omega_x = \frac{\phi}{\Delta t}$ and $\omega_y = \frac{\theta}{\Delta t}$, substituting the values in (5.7):

$$\vec{V_r} = \frac{x\,y\,\phi - \left[x^2 + F^2\right]\theta}{F\,\Delta t}, \quad \frac{\left[y^2 + F^2\right]\phi - x\,y\,\theta}{F\,\Delta t} \tag{5.12}$$

the vectorial sum of the velocity components represented by (5.11) and (5.12) is the whole velocity vector of the image point P_i.

Let us suppose the egomotion parameters D_1, D_2, ϕ, θ, and the focal length F of the camera to be perfectly known and assume Δt to be unity, then the only unknown variable to determine uniquely the image velocity $\vec{V}$ is the distance Z of the point $\vec{P_w}$ from the camera. On the other hand Z is a scale factor for $\vec{V_t}$ (see eq. (5.11)), therefore only the direction of $\vec{V_t}$ can be computed from the known parameters.

In summary, in this formulation the known (measured) terms are the egomotion parameters D_1, D_2, ϕ, θ, and the focal length of the camera F; from equation (5.11) and (5.12) the direction of $\vec{V_t}$ and the vector $\vec{V_r}$ are computed. The direction of velocity can be computed from a set of non-linear equations which incorporate the motion parameters and the constraints. This system can be solved, in closed form, for the direction of the velocity vector $\vec{V}$

$$\tan(\beta - \alpha) = \frac{[D_1 - D_2 \cos\phi \cos\theta]\left[V^{\perp}y - \cos\beta\left(yV_{rx} - xV_{ry}\right)\right]}{[D_1 - D_2 \cos\phi \cos\theta]\left[V^{\perp}x + \sin\beta\left(yV_{rx} - xV_{ry}\right)\right]} - \frac{FD_2\left[V^{\perp}\sin\phi - \cos\beta\left(V_{rx}\sin\phi - V_{ry}\cos\phi\sin\theta\right)\right]}{FD_2\left[V^{\perp}\cos\phi\sin\theta + \sin\beta\left(V_{rx}\sin\phi - V_{ry}\cos\phi\sin\theta\right)\right]} \tag{5.13}$$

where $V_{rx} = \frac{x\,y\,\phi - \left[x^2 + F^2\right]\theta}{F\,\Delta t}$ and $V_{ry} = \frac{\left[y^2 + F^2\right]\phi - x^y\,\theta}{F\,\Delta t}$ are the X and Y components of the vector $\vec{V_r}$ which represents the component of the flow field due to the rotation of the camera, and α and β are the directional angles of the vectors $\vec{V}$ and $\vec{V}^{\perp}$.

The computed direction of the velocity vectors (in radians) is converted into an integer value between 0 and 254, and then stored in the vector phase field in the velocity image. The velocity amplitude is also computed, for each contour point, and stored in the appropriate field in the velocity image.

It is worth noting that both the FOE and the direction of $\vec{V}$ can be computed by using only the ratio $R = \frac{D_1}{D_2}$. In fact:

$$FOE_x = F\,\frac{D_2 \cos\phi \sin\theta}{D_1 - D_2 \cos\phi \cos\theta} = F\,\frac{\cos\phi \sin\theta}{R - \cos\phi \cos\theta} \tag{5.14}$$

$$FOE_y \quad = \quad F \quad \frac{\sin\phi}{R \; - \; \cos\phi\cos\theta} \tag{5.15}$$

$$\begin{aligned}\tan(\beta - \alpha) & \\ = & \frac{[\mathrm{R} - \cos\phi\cos\theta]\left[\mathrm{V}^{\perp}\mathrm{y} - \cos\beta\left(\mathrm{yV}_{\alpha} - \mathrm{xV}_{\eta}\right)\right]}{[\mathrm{R} - \cos\phi\cos\theta]\left[\mathrm{V}^{\perp}\mathrm{x} + \sin\beta\left(\mathrm{yV}_{\alpha} - \mathrm{xV}_{\eta}\right)\right]} \\ - & \frac{\mathrm{F}\left[\mathrm{V}^{\perp}\sin\phi - \cos\beta\left(\mathrm{V}_{\alpha}\sin\phi - \mathrm{V}_{\eta}\cos\phi\sin\theta\right)\right]}{\mathrm{F}\left[\mathrm{V}^{\perp}\cos\phi\sin\theta + \sin\beta\left(\mathrm{V}_{\alpha}\sin\phi - \mathrm{V}_{\eta}\cos\phi\sin\theta\right)\right]}\end{aligned} \tag{5.16}$$

Let us consider the simplified case of a motion of the camera where the rotational velocity of the camera ω_x is zero (i.e. a motion trajectory which lies on the ZX plane, with reference to the camera coordinate system, while tracking the fixation point), as shown in figure 5.2: θ represents the rotational angle around an axis orthogonal to the plane of motion, whereas γ represents the direction of the translational axis between position 1 and 2. Using basic trigonometric relations it is possible to express the distance D_2 as a function of D_1 :

$$\begin{aligned} \mathrm{D}_2 &= D_1 \left[\frac{\cos\theta \; - \; \cos\gamma\,\cos(\theta+\gamma)}{\sin^2(\theta+\gamma)}\right] \\ &= D_1 \left[\frac{\sin\gamma\,\sin(\theta+\gamma)}{\sin^2(\theta+\gamma)}\right] \\ &= D_1 \left[\frac{\sin\gamma}{\sin(\theta+\gamma)}\right] \end{aligned} \tag{5.17}$$

$$R \quad = \quad \frac{\sin\gamma}{\sin(\theta+\gamma)} \tag{5.18}$$

If the camera performs also a rotation ϕ around its X axis, and the angle γ has been measured with respect to the Y axis of the camera in the first position, then we obtain:

$$D_2 \quad = \quad D_1 \left[\frac{\sin\gamma}{\sin(\theta+\phi)\,\cos\gamma}\right] \tag{5.19}$$

$$R \quad = \quad \frac{\sin\gamma}{\sin(\theta+\phi)\,\cos\gamma} \tag{5.20}$$

This computation can be done every time it is possible to approximate the real camera trajectory piecewise with straight lines of known spatial orientation.

Once the direction of velocity is computed, a first approximation of the magnitude of the vector $\vec{V}$ is obtained from $V^{\perp}$; then the value is refined searching for the first zero-crossing in the successive image along the computed direction.

Starting from a contour point $({x_0}^T, {y_0}^T)$ on the image at time T, the corresponding point in the successive image is searched, considering all the image points which lie

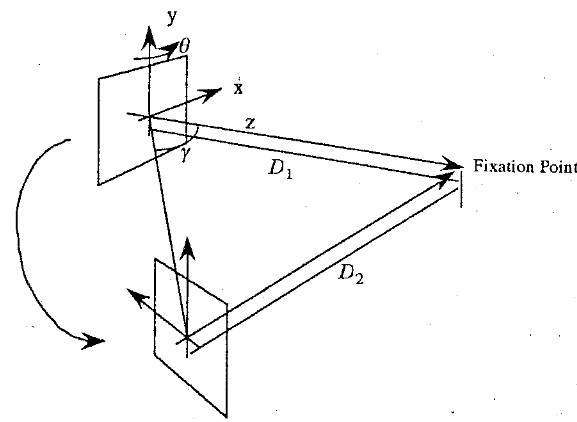

Figure 5.2: Tracking a fixation point

along the computed direction of $\vec{V}$. The first contour point found in the image at time $T + \Delta t$ is taken as the corresponding point of $({x_0}^T, {y_0}^T)$. The search is limited to a distance proportional to:

$$\left|\vec{V}\right| \; = \; \frac{V^{\perp}}{\cos\alpha} \tag{5.21}$$

where α is the angle between the velocity vector $\vec{V}$ vec and the perpendicular component. This kind of search is motivated by the fact that, due to the high sampling frequency and the smoothing effected by the Gaussian filtering, it is unlikely that more than one contour between corresponding edges will be found, particularly along direction of $\vec{V}$.

In order to match edges in case of underestimated velocities, a value equal to the sigma of the operator used for convolution is added to the displacement $\left|\vec{V}\right|$ to set the limit for the search. This correction factor should compensate the localization errors of edge points. A more accurate analysis should modify this value taking into account the uncertainty of the computed velocity; in such case the direction of the vector should also be modified.

Through the matching of corresponding contours an approximation of *instantaneous optic flow* is determined, relative to each pair of frames. In order to achieve a sufficient range of velocity for distance computation, the instantaneous optic flows are joined together obtaining a *global optic flow*, relative to a part of the sequence, from image T to $T+N$, where N is the time span (with an attendant baseline between viewpoints) which is used for triangulation.

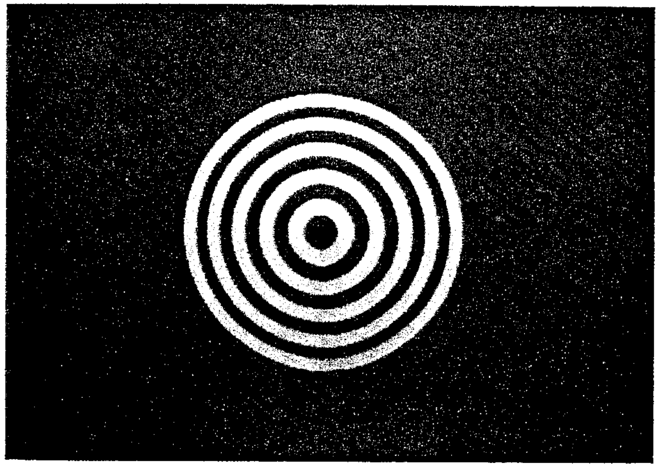

Figure 5.3: An intensity image of a very simple object: a cone with black and white stripes

During the matching process other fields of the velocity image is updated. In particular, for each contour point, the image, contour and contour point number of the corresponding (matched) point in the successive image are recorded. These data allow a linkage of contour points over time and, consequently, the representation of the cumulative trajectory over time.

Figures 5.3 through 5.8 and 5.11 through 5.16 illustrate the approach. For a given scene (figures 5.3 and 5.11), a sequence of nine images is generated (figures 5.4 and 5.12) as a camera mounted in a robot arm is rotated about a fixation point. Each of the constituent images in these image sequences is then convolved with a Laplacian of Gaussian mask (standard deviation of the Gaussian function = 4.0 pixels) and the zero-crossings contours are extracted (figures 5.5 and 5.13). An adaptive thresholding technique is employed to identify the most relevant contours (see figures 5.6 and 5.14) and the instantaneous optic flow relative to images 3 and 4 is computed (figures 5.7 and 5.15). The vectors shown represent a sampling of the flow field along the contours. Finally, the global optic flow relative to images 3 through 8 is computed by linking the instantaneous optic flows relative to each image pair, computed at the contour points (figures 5.8 and 5.16).

5.4 Inferring Depth from Camera Motion.

The computation of depth is done straightforwardly from the knowledge of the optic flow, by a triangulation using corresponding points of the first and last frame of the

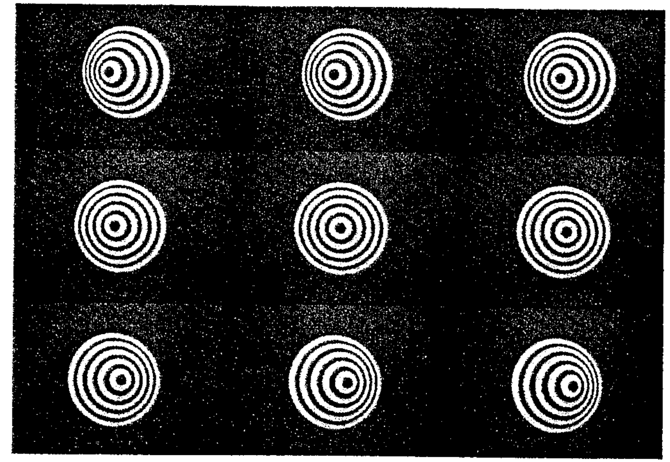

Figure 5.4: A sequence of nine views of the object

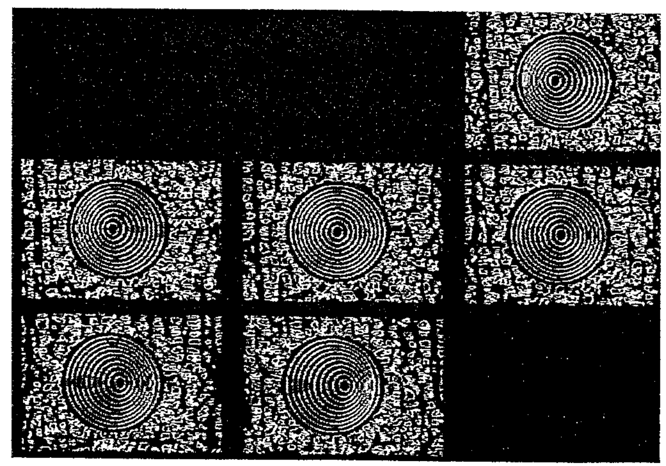

Figure 5.5: Zero-crossing images 3 through 8 of the sequence

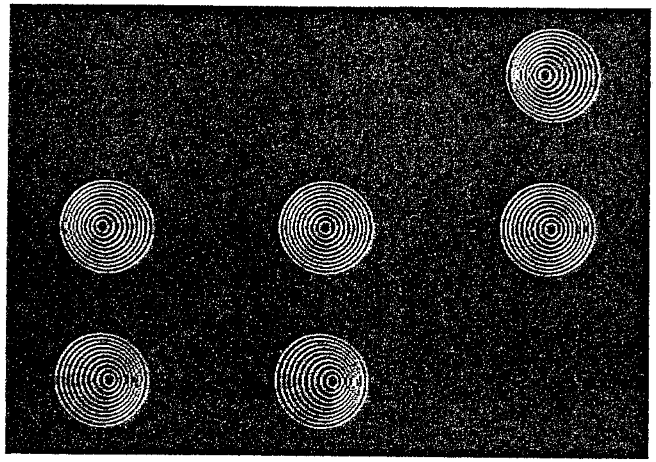

Figure 5.6: Adaptively-selected zero-crossings

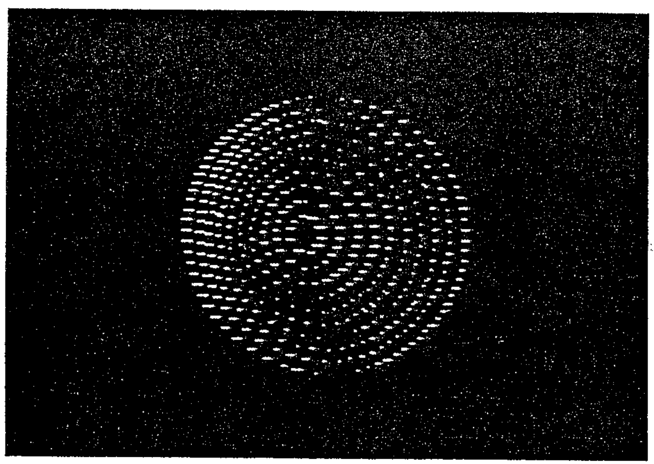

Figure 5.7: Instantaneous optic flow relative to images 3 and 4

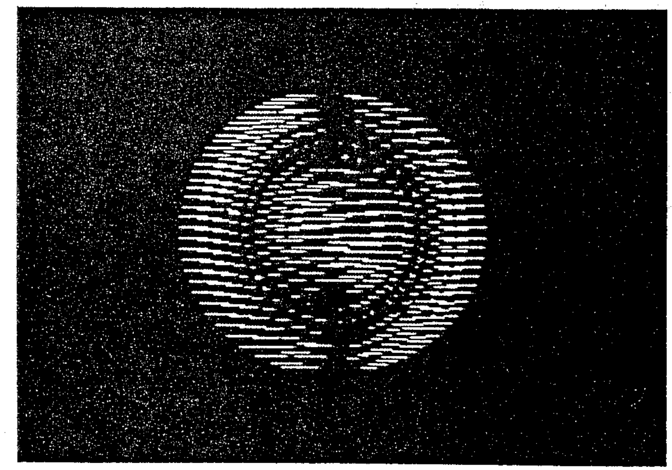

Figure 5.8: Global optic flow relative to images 3 through 8

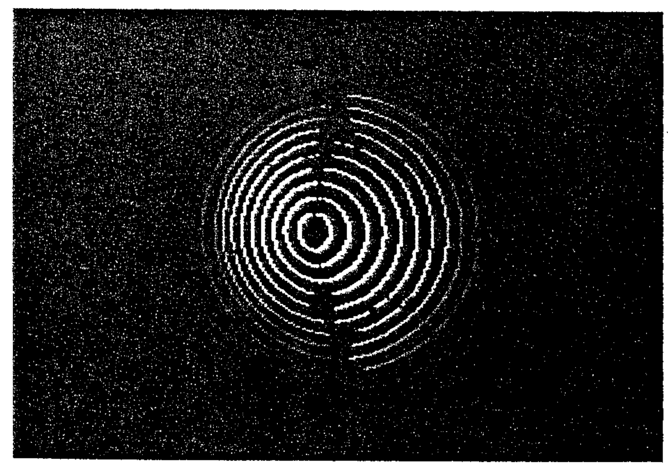

Figure 5.9: Depth image derived from the optic flow field (brightness is proportional to distance)

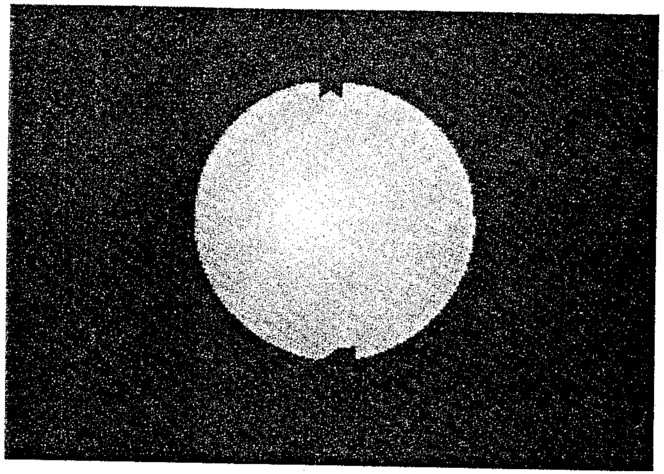

Figure 5.10: Range image derived by interpolation from the depth image (brightness is proportional to distance)

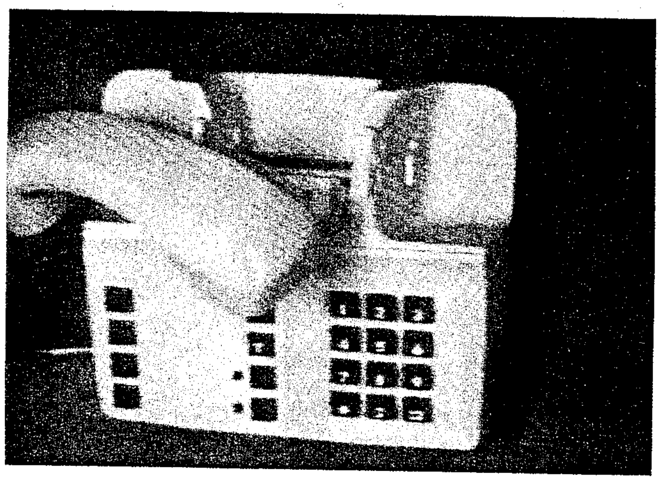

Figure 5.11: An intensity image of a more complex object: a telephone

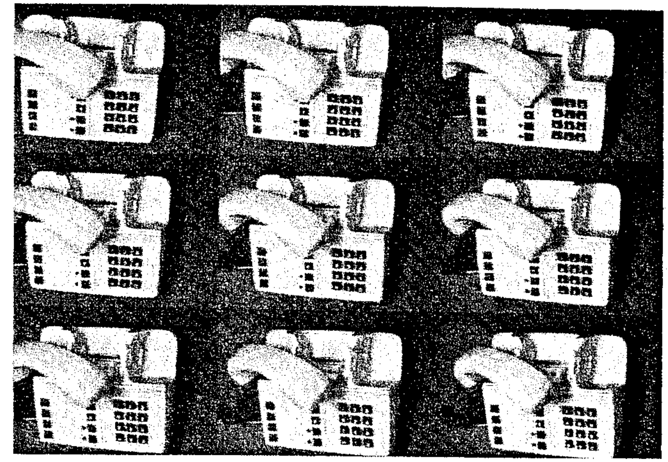

Figure 5.12: A sequence of nine views of the object

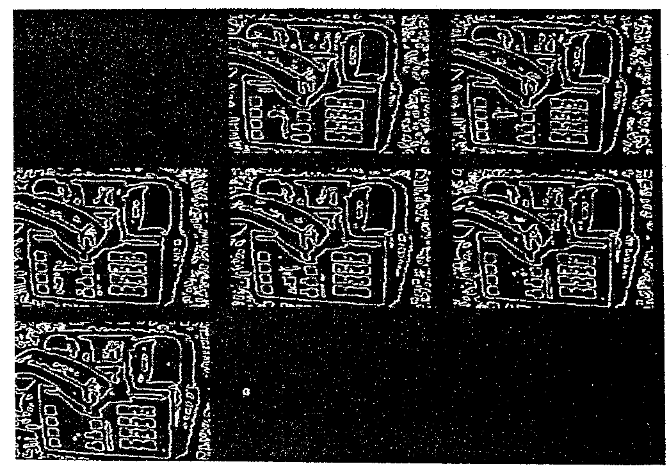

Figure 5.13: Zero-crossing images 3 through 8 of the sequence

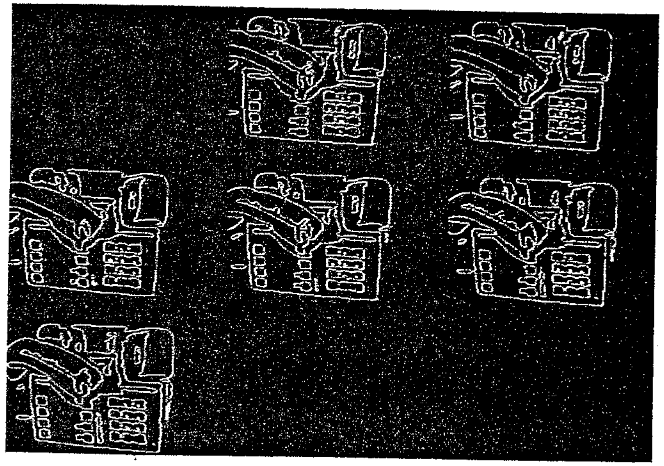

Figure 5.14: Adaptively-selected zero-crossings

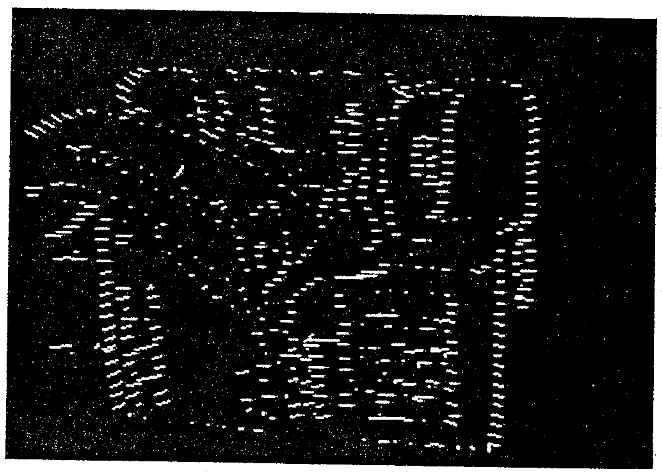

Figure 5.15: Instantaneous optic flow relative to images 3 and 4

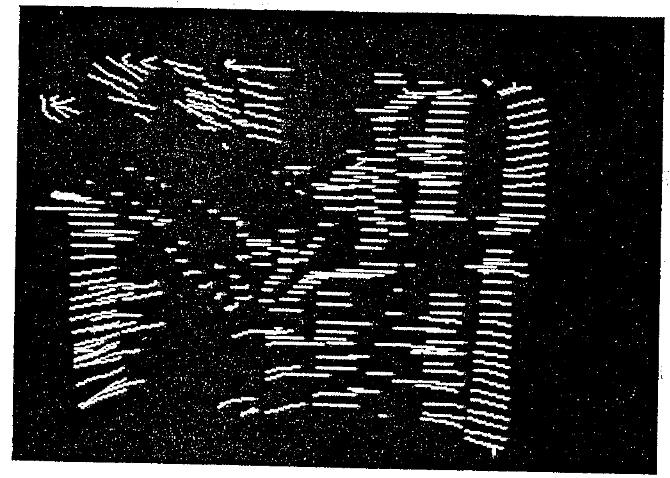

Figure 5.16: Global optic flow relative to images 3 through 8

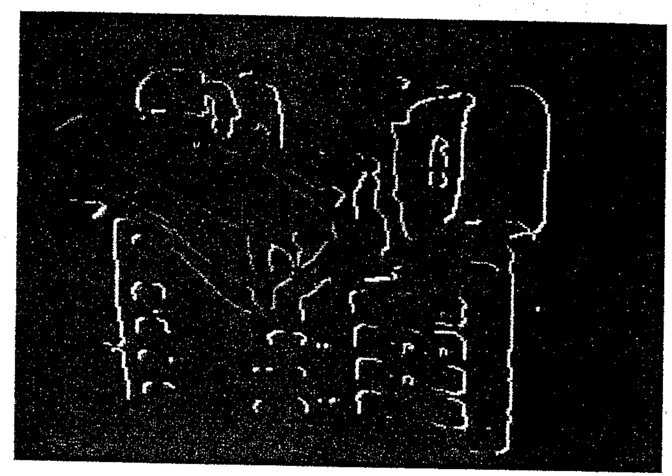

Figure 5.17: Depth image derived from the optic flow field (brightness is proportional to distance)

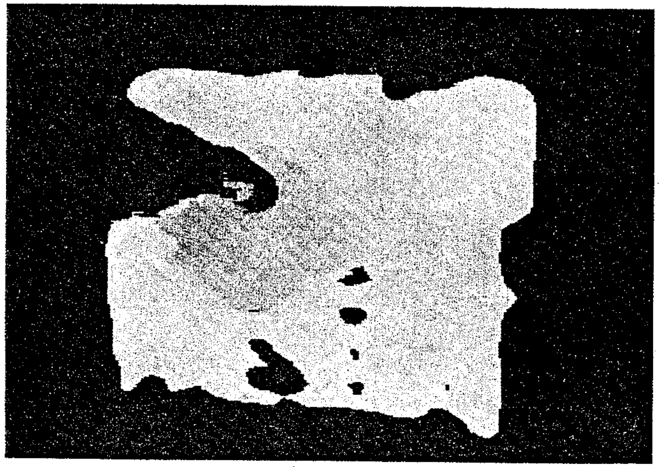

Figure 5.18: Range image derived by interpolation from the depth image (brightness is proportional to distance)

interval within the sequence, i.e., the *initium* and the *terminus* of the global optic flow vector:

$$\frac{D_f}{\left|\vec{V}_t\right|} = \left|\frac{Z}{W_Z}\right| \tag{14}$$

where Z is the distance of the environmental point from the camera, D_f is the distance of the image point from the FOE, V_t is the translational component of the flow and W_Z is the camera velocity along the optic axis. The ratio on the right gives the relative depth or the *time until contact* of the world point with respect to the camera. It is just equivalent to the real distance of the point from the camera up to a common factor that is the velocity W_Z of the observer. The information which is thus acquired is a depth map for each point on the zero-crossing contours. Clearly, this is a sparse map of depth values and a dense map (the range image) can be computed by interpolation. This is dealt with in Chapter 8. A complete analysis of the accuracy of this depth computation is contained in [30].

To illustrate this approach to computing the depth of objects, figures 5.10 and 5.18 show a range image representing the range of most visible points on the surface. This was generated by weighted linear interpolation between depth values on zero-crossing contours (see figures 5.9 and 5.17). Spurious values in this range image have been suppressed by applying an 11x11 pixel modal filter, whereby the value of

each point in the range image is replaced with the value associated with the mode of a histogram of the values in a local region around that point.

Bibliography

[1] Aggarwal, J.K. and Wang, Y.F. 1985. 'Structure and motion computation from point or line correspondence in images', Proc. International Conference on Advances in Image Processing and Pattern Recognition, Elsevier Science Publications, North-Holland, pp. 171-178.

[2] Ballard, D.H. and Brown, C.M. 1982. *Computer Vision*, Prentice-Hall, New Jersey.

[3] Bandopadhay, A., Chandra, B., and Ballard, D.H. 1986. 'Active navigation: tracking an environmental point considered beneficial', Proc. of the Workshop on Motion: Representation and Analysis, IEEE Computer Society, pp. 23-29.

[4] Bharwani, S., Riseman, E., and Hanson, A. 1986. 'Refinement of environmental depth maps over multiple frames', Proc. of the Workshop on Motion: Representation and Analysis, IEEE Computer Society, pp. 73-80.

[5] Braccini, C., Gambardella, G., Grattarola, A., Massone, L., Morasso, P., Sandini, G., and Tistarelli, M.1987. 'Object reconstruction from motion: comparison and integration of different methods', Proc. 2nd. Int. Workshop on Time-Varying Image Processing and Moving Object Recognition, Florence, Italy.

[6] Bridwell, N.J. and Huang, T.S. 1983. 'A discrete spatial representation for lateral motion stereo', Computer Graphics Vision an Image Processing, Vol. 21, pp. 33-57.

[7] DeMicheli, E., Sandini, G., Tistarelli, M., and Torre, V. 1988. 'Estimation of visual motion and 3D motion parameters from singular points', Proc. IEEE Int. Workshop on Intelligent Robots and Systems, Tokyo, Japan, 1988.

[8] Hildreth, E. 1983. *The measurement of visual motion*, MIT Press, Cambridge, USA.

[9] Horn, B.K.P. and Schunck, B.G. 1981. 'Determining Optical Flow', Artificial Intelligence, Vol. 17, No. 1-3, pp.185-204.

[10] Horn, B.K.P. 1986. *Robot Vision*, MIT Press, Cambridge, USA.

[11] Lawton, D.T. 1983. 'Processing translational motion sequences', Computer Vision Graphics and Image Processing, Vol. 22, pp. 116-144.

[12] Marr, D. and Poggio, T. 1979. 'A computational theory of human stereo vision', Proc. Royal Society of London, Vol. B204, pp. 301-328.

[13] Marr, D. and Hildreth, E. 1980. 'Theory of Edge Detection', Proc. Royal Society of London, Vol. B207, pp. 187-217.

[14] Matthies, L. and Kanade, T. 1987. 'Using undertainty models in visual motion and depth estimation', Proc. 4th International Symposium of Robotics Research, Santa Cruz, Calif., pp. 120-138.

[15] Morasso, P., Sandini, G., Tagliasco, V., and Zaccaria, R. 1977. 'Control strategies in the eye-head coordination system', IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-7, No. 9, pp. 639-651.

[16] Morasso, P., Sandini, G., and Tistarelli, M. 1986. 'Active vision: integration of fixed and mobile cameras', NATO ARW on Sensors and Sensory Systems for Advanced Robot.

[17] Nagel, H.H. 1983. 'Displacement vectors derived from second-order intensity variations in image sequences', Computer Vision Graphics and Image Processing, Vol. 21, pp. 85-117.

[18] Nagel, H.H. and Enkelmann, W. 1986. 'An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences', IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-8, No. 1, pp. 565-593.

[19] Nagel, H.H. 1987. 'On the estimation of optical flow: relations between different approaches and some new results', Artificial Intelligence, Vol. 33, pp.299-324.

[20] Nagel, H.H. 1990. 'Extending the "orientation smoothness constraint'Into the temporal domain and the estimation of derivatives of optical flow', Proc. 1st European Conference on Computer Vision, Antibes (France), Springer-Verlag, pp. 139-148.

[21] Prazdny, K. 1980. 'Egomotion and relative depth map from optical flow', Biological Cybernetics, Vol. 36, pp. 87-102.

[22] Rieger, J.T. and Lawton, D.T. 1984. 'Processing differential image motion', Technical Report, COINS Dept., University of Massachusetts, Amherst (MA).

[23] Sandini, G., Tagliasco, V., and Tistarelli, M. 1986. 'Analysis of object motion and camera motion in real scences', Proc. IEEE International Conference on Robotics and Automation, San Francisco, pp. 627-633.

[24] Sandini, G. and Tistarelli, M. 1986. 'Recovery of depth information: camera motion as an integration to stereo', Proc. of the Workshop on Motion: Representation and Analysis, IEEE Computer Society, pp. 39-43.

[25] Sandini, G., Morasso, P., and Tistaelli, M. 1987. 'Motor and spatial effects in artificial vision', Proc. of the 4th Int. Symposium of Robotics Reseach, Santa Cruz, California, MIT Press.

[26] Snyder, M.A. 1986. 'The accuracy of 3D parameters in correspondence-based techniques: startup and updating', Proc. of the Workshop on Motion: Representation and Analysis, IEEE Computer Society, pp. 53-59.

[27] Sugie, N. and Wakakuwa, M. 1970. 'Visual target tracking with active head rotation', IEEE Transactions on Systems Science and Cybernetics, Vol. SSC-6, pp.103-108.

[28] Tsai, R. Y. and Huang, T.S. 1984. 'Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces', IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-6, No. 1, pp. 13-26.

[29] Uras, S., Girosi, F., Verri, A., and Torre, V. 1988. 'Computational approach to motion perception', Biological Cybernetics, Vol. 60, pp.69-87.

[30] Vernon, D. and Tistarelli, M. 1990. 'Using Camera Motion to Estimate Range for Robotic Parts Manipulation', IEEE Transaction on Robotics and Automation, Vol. 6, No. 5, pp. 509-521.

llis HORWOOD SERIES IN
RTIFICIAL INTELLIGENCE
eries Editor: Professor JOHN CAMPBELL, Department of Computer Science, University
ollege London

ARALLEL COMPUTER VISION
he VIS a VIS System
D. VERNON, Department of Computer Science, Trinity College Dublin, Ireland, and
ofessor G. SANDINI, DIST, University of Genoa, Italy

escribing a computer vision environment which supports coarse granularity parallelism on
ansputer-based systems, this book discusses a system which treats the organization of the
age data and its representation and interdependencies equally with the visual processes
hich operate on that data. As an environment, the system—VIS a VIS—provides dynamic
eation of images and attendant data structures, each of which are linked to provide an
plicit representation of the interdependence of the visual cues. As a vision system, VIS a
S offers all the required functionality of 3-D image understanding, from image acquisition,
rough detection and analysis of intensity discontinuities, computation of depth from stereo
d camera motion, segmentation using raw and full primal sketches, through to 3-D model
nstruction and object recognition. Both geometric and functional parallelism are achieved
exploiting both the in-built vision control language—with its facility for remote execution
procedures—and the integrated system data-structures.

of these issues are dealt with in considerable detail, and the book's emphasis on system
ganization integration of information and computer architectures, rather than purely
eoretical and algorithmic issues, will make it a very welcome addition to the literature.

adership: Computer vision scientists. Industrial research and development engineers.
mputer scientists. Information technologists.

id Vernon's interests stretch from robot vision to the philosophy of science. His recent research
ivities have been concerned with the development of robot systems which display true autonomy. His
rk encompasses computational theories of perception, self-organization, and intelligence. Dr Vernon
Fellow of Trinity College Dublin, Ireland.
lio Sandini has been working on research in computer and biological vision for more than twelve
rs, his primary interests at present being the integration of many low-level visual cues in a robust
nputational framework. He has also been very active in exploiting the neurobiological principles of
ly vision to postulate and implement artificial visual sensing at the 'retinal' level of computer vision.
fessor Sandini is currently at the University of Genoa in Italy.

les of related interest

RMAL SPECIFICATION OF ADVANCED AI ARCHITECTURES
N D. CRAIG, Department of Computer Science, University of Warwick

TIFICIAL INTELLIGENCE AND INTELLIGENT SYSTEMS
e Implications
VID ANDERSON, Division of Computer Science, Teesside Polytechnic

TIFICIAL INTELLIGENCE TERMINOLOGY
eference Guide
neral Editor: COLIN BEARDON, Department of Computer Science, University of Exeter

ISBN 0-13-932716-9
9 780139 327162

LIS HORWOOD

Vernon and Sandini **PARALLEL COMPUTER VISION**
the VIS $\vec{a}$ VIS system
ELLIS HORWOOD

ELLIS HORWOOD SERIES IN ARTIFICIAL INTELLIGENCE

PARALLEL COMPUTER VISION

the VIS $\vec{a}$ VIS system

D. Vernon and G. Sandini