Reviews of Optic Flow, Motion Segmentation, Edge finding and Corner Finding

Technical Report TR97SMS1

(Derived originally from [99])

1997

Stephen M. Smith
Oxford Centre for Functional Magnetic Resonance Imaging of the Brain (FMRIB),
Department of Clinical Neurology, Oxford University, Oxford, UK
steve@fmrib.ox.ac.uk
www.fmrib.ox.ac.uk/~steve

1 Finding Optic Flow

Almost all work in image sequence processing begins by attempting to find the vector field which describes how the image is changing with time. Ideally, the projection into the two dimensional image plane of the three dimensional velocity field seen by the camera should be computed. However, this is not just difficult to achieve in practice, it is usually impossible to achieve (perfectly) even in theory. The reasons for this are well understood; Horn and Schunk in [52] give the example of a rotating sphere with no surface markings which, under constant illumination, causes no changes in the image intensity over time, even though there *is* motion in the world. Related to this is the well known aperture effect – for example, when a pole moves behind a window, only the component of motion perpendicular to the pole can be found. These problems have led to careful definitions of optic flow being made; it is frequently necessary to make the distinction between the motion perceived in the image and the theoretical projection of the world velocities into the image. A discussion on this subject can be found in [116].

With 2D feature-based optic flow, the need for making explicit which sort of optic flow is being discussed does not arise. Neither does the question of whether the full flow is being found, or just a component of it. This is for two reasons. Firstly, optic flow is found by matching two dimensional features. One of the strongest advantages of this approach is the fact that it does not suffer from the problems of the aperture effect, for obvious reasons. (In [71] Nagel discusses this in detail.) Secondly, the work is largely aimed at producing good results from images of real events, mostly taken outside. Thus there is rarely a difference in practice between the image flow and the projected world motion. Thus the term optic flow will be used freely and generally.

The different methods used for finding optic flow reported here do not always fit neatly into the three categories described below, but an attempt has been made to give a clear discussion of the subject by looking at each approach in turn.

1.1 The Feature-Based Approach

The method of finding image displacements which is easiest to understand is the feature-based approach. This finds features (for example, image edges, corners, and other structures well localized in two dimensions) and tracks these as they move from frame to frame. This involves two stages. Firstly, the features are found in two or more consecutive images. The act of feature extraction, if done well, will both reduce the amount of information to be processed (and so reduce the workload), and also go some way towards obtaining a higher level of understanding of the scene, by its very nature of eliminating the unimportant parts. Secondly, these features are matched between the frames. In the simplest and commonest case, two frames are used and two sets of features are matched to give a single set of motion vectors. Alternatively, the features in one frame can be used as seed points at which to use other methods (for example, gradient-based methods – see the following section) to find the flow.

The two stages of feature-based flow estimation each have their own problems. The feature detection stage requires features to be located accurately and reliably. This has proved to be a non-trivial task, and much work has been carried out on feature detectors; see the reviews in Sections 3 and 4. If a human is shown, instead of the original image sequence, a sequence of the detected features (drawn onto an *empty* image), then a smoothly moving set of features should be observable, with little feature flicker. The feature matching stage has the well known correspondence problem of ambiguous potential matches occurring; unless image displacement is known to be smaller than the distance between features, some method must be found to choose between different potential matches.

Finding optic flow using edges has the advantage (over using two dimensional features) that edge detection theory is well advanced, compared with that of two dimensional feature detection. It has the advantage over approaches which attempt to find flow everywhere in the image, such as the method developed by Horn and Schunk in [52] (see Section 1.2), that these other methods are poorly conditioned except at image edges; thus it is sensible to only find flow at the edges in the first place.

Most edge detection methods which have been used in the past for finding optic flow are related to either finding maxima in the first image derivative, see [21], or finding zero-crossings in the Laplacian of a Gaussian of the image, see [59]. The major problem with using edges is that unless assumptions about the nature of the flow are made, and further processing (after the matching) is performed, only the component of flow perpendicular to each edge element can be found. Unfortunately, the assumptions made will invariably lead to inaccuracies in the estimated flow, particularly at motion boundaries, etc.

The first major work taking this approach was that of Hildreth; see [51]. The edges are found using a Laplacian (∇^2) of a Gaussian edge detector, and the image motion is found at these edges by using the brightness change constraint equation (see the following section). Various additional constraints necessary to recover the motion parallel to the edge direction are discussed. The constraints all make the basic assumption that the image flow is smoothly varying across the image. Thus each minimizes some measure of flow variation. The measure chosen is the variation in velocity along the edge contours. If this is completely minimized, the data is largely ignored, so a balancing term is added to the minimization equation which gives a measure of the fit of the estimated flow to the data. Thus the following expression is minimized;

$$\int_{\text{along contour}} \left(\left(\frac{\partial u}{\partial s} \right)^2 + \left(\frac{\partial v}{\partial s} \right)^2 \right) ds + \beta \int_{\text{along contour}} (\vec{u}.\vec{n} - u^{\perp})^2 ds, \tag{1}$$

where $\vec{u} = (u, v)^T$ is the estimated optic flow, $\vec{n}$ is the unit vector perpendicular to the edge, $u^{\perp}$ is the measured component of flow perpendicular to the edge, s is the "arclength" parameter and β is the weighting factor which determines the relative importance of the two terms. The first term expresses the variation in estimated flow, and the second term expresses the quality of the fit of the estimated flow to the measured edge displacements. The expression is minimized using the conjugate gradient approach, also described in [51]. The overall algorithm is intrinsically sequential, and is fairly computationally expensive.

The use of edges not just in space but in space-time was introduced by Buxton and Buxton; the theory is developed in [18], and testing is reported in [17]. Here they use the d'Alembertian ($\nabla^2 - \frac{1}{u^2}\frac{\partial^2}{\partial t^2}$) of a spatiotemporal Gaussian, instead of the Laplacian. The velocity parameter u and the scale of the Gaussian filter need to be applied at different channels to search for different motion possibilities. Although noise suppression was claimed to be greater than that achieved using Hildreth's approach due to the extended dimensionality of the filter, a corresponding increase in computational expense also resulted. In [19] this work is extended; the negative metric is replaced with the positive one (i.e. $\nabla^2 + \frac{1}{u^2}\frac{\partial^2}{\partial t^2}$) to reduce noise sensitivity and removing "curious" cross-over effects (mentioned in [18], but never utilized). In [19], instead of using Hildreth's velocity smoothness constraint, an object's motion and position parameters are estimated directly from the measurements of the normal components of flow. However, this "3D solution to the aperture problem", solved with a least squares fit to the normal flow, assumes that the objects in question are planar, and that the image is already segmented into differently moving objects.

In [22] Castelow et. al. use Canny edgels (edge elements found using the Canny edge detector) to find optic flow at edges. In this method, developed originally by Scott (see [90]), the flow parallel to the edges is recovered by iterative relaxation of the edgel matching surface; the edgels near the high curvature parts of the contours can have both components of flow found from the data, so these cues to the full flow are propagated along the contour into less highly curved sections by relaxation. Thus local support of optic flow is used to improve the motion estimate iteratively. The results presented did not appear to be as good as those shown in [51] and [19], although the method has been successfully used by Murray et. al. to reconstruct three dimensional objects in structure from motion work; see [70] and [68]. This method of finding optic flow has also been used to track cataracts, in [45].

In [44], Gong and Brady found the full flow at points of high curvature on zero-crossing contours using their "curve motion constraint equation". The flow is then propagated along the contours by using a wave/diffusion process. This method was developed further by Wang et. al. in [119]. Edges are found using Canny's edge detector, and are then segmented and grouped. "Corners" in the edges are found by fitting local B-splines, and the edges at these points are matched from one frame to the next (using a cross-correlation method to achieve correct matches) to give the full flow. This flow is then propagated along the edges using the wave/diffusion process. The algorithm was implemented in parallel, and so achieved an impressive speed compared with the original algorithm, and compared with Hildreth's approach. The results obtained were good (for an edge-based method), although the full flow was not always correctly estimated along straight portions of edges. The use of high curvature parts of edges as two dimensional features can lead to the rejection of features which are well localized in two dimensions (and therefore useful for recovery of optic flow) but are not placed on well defined edges. ASSET [103, 102, 100] finds two dimensional features directly from the image, and not via an intermediate edge finding stage, and therefore avoids this problem.

Some of the ways in which edges have been used to find optic flow for different purposes are now given.

In [30] Deriche and Faugeras track *straight* edges in two dimensions using Kalman filters for each edge. The edges are found using the Deriche filter (see [29]) and are represented by their length, orientation and position. The resulting matched and tracked line segments are fed into a structure from motion algorithm; however, this approach is limited to fairly structured environments, as it only uses straight lines.

In [12] Bolles et. al. use spatiotemporal edge filters as developed by Buxton and Buxton (see earlier) to track features in single raster lines over time. The tracks of features within these raster lines create "epipolar-plane images"; these contain information about the three dimensional structure of the world. The assumption is made that the images are taken in quick succession compared with the motion of the features, so that there is no correspondence problem. In [112] Tomasi and Kanade track edges found using the Laplacian of a Gaussian in an approach very similar to that of Bolles. From the tracked edges the shape of an object is computed "without first computing depth". Very accurate results are obtained, using highly constrained camera motion.

In [89] Sandini and Tistarelli use Laplacian of a Gaussian edge contours, tracked over time by a camera moving with constrained motion. The tracked contours are used to construct the three dimensional structure of objects in view. In [8] Blake and Cipolla (again, given a deliberate, constrained and known camera motion) use the deformation of edge contours to recover surface curvature in the world. In [36] Faugeras discusses the problems of tracking image edges associated with non-rigid bodies. In [24] Chen and Huang match straight line segments in stereo sequences for object motion estimation.

Optic flow methods based on the detection of two dimensional features (which shall be known as corners for the purpose of this discussion) have the advantage that the full optic flow is known at every measurement position. The (related) disadvantage is that only a sparse set of measurements is available, compared with methods which give a measurement at every image position. However, if the features are found appropriately then the maximum possible number of different optic flow measurements in the image should be recoverable, without the degradation in flow estimate quality introduced by the various assumptions and constraints required to recover the full flow using other methods. Note that arguments have often been made that gradient-based (and related) methods have their flow estimations achieving the most well conditioned results near edges. In the same way, flow estimates found at edges are most accurate near corners. In the ASSET [103, 102, 100] system, corners are used for finding optic flow, and image segmentation is based on this; the segmentation results are then further refined by using "intra-frame" image edges. This appears to be a sensible use of the image data, since the flow is found at those places where this can be most accurately achieved, and remaining (instantaneous) image information about possible object boundaries can be used to make adjustments to the interpretation made using the flow. Another advantage of using corners is that the early data reduction achieved should allow computationally efficient algorithms to be developed.

Some of the published discussions on optic flow found from moving corners and some of the ways in which this information has been recovered and used are now described.

Much work on structure from motion has assumed that the flow field is only recovered at relatively sparse image positions. The work has largely consisted of theory relating how much scene information can be given from different numbers of views of varying numbers of "world points". For example, in [38] Faugeras and Maybank show that in general the least number of point matches (found using two viewpoints) which allows only a finite number of interpretations of the camera positions (relative to the world) is five. With this number of matches, there are a maximum of ten possible interpretations of the camera's motion.

With regard to the problem of matching corners from one frame to the next, early work was undertaken by Barnard and Thompson; see [4]. Here the Moravec interest operator is used to find points which are well localized in two dimensions in consecutive images. Next, the (computationally expensive) method of iterative relaxation of the matching surface is used to find the optimal set of point matches. Each point in the first frame is given a probability of matching to each point in the second; the initial probability is inversely proportional to the sum of the squares of the difference between the brightness of each point within a small patch centred on the "corner" in the first frame and a small patch centred on the "corner" in the second. Then relaxation is applied to force the flow field to vary smoothly. (This matching validation approach could be overly restrictive in certain situations, where flow discontinuities are important sources of information.) In [111] this method is used to successfully detect moving objects as seen by a moving camera; this work is discussed later.

In [23] Charnley and Blissett describe the vision system DROID. DROID uses two dimensional features found with the Plessey feature detector (see Section 4), and tracks these over many frames in three dimensions using a Kalman filter for each feature. Time-symmetric matching is performed on the basis of feature attributes, using (where available) predicted feature positions from the filters. The tracked features are used to construct a three dimensional surface representing the world. The constraint is applied that the world must contain no moving objects.

In [93] L.S. Shapiro et. al. describe a corner tracker using the Wang corner detector (see [118]). Small inter-frame motion is assumed. Corners are matched using correlation of small patches. Where no matches are found, motion is predicted, using again the correlation of small patches, in a manner very similar to the method of Lawton (see the following paragraph). Tracking is achieved over many frames using simple two dimensional filters which model image velocity and (if required) image acceleration.

In [56] Lawton uses optic flow to recover world structure assuming a static environment and translational camera motion. Image flow is found by taking corners from one frame only, and using correlation of small patches centred at the features with patches placed at all possible positions in the second frame. Corners are found by using an approach similar to that of Moravec, (see Section 4) but are only looked for along Laplacian of Gaussian edges. The flow is found to sub-pixel accuracy by using inter-pixel interpolation in both frames, once the approximate displacement has been found. Again, cross-correlation is used for this. In [1] Adiv uses Lawton's method to find the structure and motion of independently moving planar objects; the planar spatial regions are later fused into multi-facet objects.

In [95] Sinclair et. al. use the motion of corners to find three dimensional planes in the world (via projective invariants) and also the direction of the camera's motion. Burger and Bhanu, in [14] use the motion of "endpoints and corners of lines as well as centers of small closed regions" to find the direction of translation of the camera. Work on image segmentation using the motion of two dimensional features includes [10], [11], [120] and [28].

Some work has been carried out which attempts to integrate the motion of corners and edges. In [37] Faugeras et. al. compare recovered three dimensional structure from the motion of points and edges theoretically and experimentally. In [107] Stephens and Harris used edges and corners to recover the three dimensional structure of a static environment (in a project related to the DROID system). They integrated the tracked features with edges in an attempt to provide wire-frame models of static objects in the world. This integration proved difficult, and was dropped in later versions of DROID, where edges were not used.

The details of how ASSET recovers a useful image flow field from sets of two dimensional features are given in [103, 102, 100], where some justification of the approach taken is also given.

1.2 The Gradient-Based Approach

Gradient-based methods use spatial and temporal partial derivatives (or related functions; see the approach of Heeger below) to estimate image flow at every position in the image. If the image motion is not known in advance to be restricted to a small range of possible values then a multi-scale analysis must be applied (coarse to fine) so that the scale of the smoothing prior to derivative estimation is appropriate to the scale of the motion. This can result in gradient-based methods being computationally expensive. This problem is the equivalent of the correspondence problem in feature-based methods.

In [35] Enkelmann tests a gradient-based "analytical" method against an edge-contour-based method and a 2D-feature-point-based method, in an experiment to detect a static obstacle in front of a moving vehicle. Although all three methods appear to detect the obstacle, Enkelmann states a preference for the gradient-based method, apparently because of its reliability. However, the DROID system mentioned earlier, based on feature points, is designed for exactly this kind of situation, and has reliably found static obstacles in the path of a vehicle. Also, the gradient-based method has the weakness, which is mentioned in [35] (and which is shown in the results) that the accuracy of the outline of the obstacle is poor, due to smoothing effects. This problem would be accentuated in the case of moving obstacles.

Probably the most well known work on optic flow is [52]. Here Horn and Schunk use spatiotemporal derivatives of the evolving image brightness function to give a single equation which partially determines the optic flow; the assumption is made that the brightness of any part of the imaged world changes very slowly, so that the total derivative of the brightness is zero. When differentiated using the chain rule, this gives what is often known as the brightness change constraint equation (BCCE);

$$\frac{\partial I}{\partial x}\frac{dx}{dt}+\frac{\partial I}{\partial y}\frac{dy}{dt}+\frac{\partial I}{\partial t}=0, \tag{2}$$

where $I(x, y, t)$ is the image brightness function. Replacing $\frac{dx}{dt}$ and $\frac{dy}{dt}$ with the optic flow $\vec{u}=(\frac{dx}{dt},\frac{dy}{dt})^T$, gives the short formulation of the BCCE;

$$\vec{u}.\vec{n}=-\frac{I_t}{|\nabla I|}, \tag{3}$$

where $\nabla I=|\nabla I|\,\vec{n}$. This equation has formed the basis of a very large proportion of research into optic flow. Horn and Schunk note that on its own the BCCE cannot fully determine the flow; it only gives the component of flow in the direction of the brightness gradient. This is, of course, directly related to the aperture effect mentioned with regard to moving edges. Thus a constraint is imposed *isotropically* which forces a smooth variation in the flow across the image. This is done in a manner very similar to that of Hildreth described earlier, by minimizing a weighted sum of the smoothness term (left) and the brightness change constraint expression (right);

$$\iint\limits_{\text{over image}}\left(\left(\tfrac{\partial u}{\partial x}\right)^2+\left(\tfrac{\partial u}{\partial y}\right)^2+\left(\tfrac{\partial v}{\partial x}\right)^2+\left(\tfrac{\partial v}{\partial y}\right)^2\right)dx\,dy+\beta\iint\limits_{\text{over image}}(\nabla I.\vec{u}+I_t)dx\,dy. \tag{4}$$

Again the weighting factor β determines the relative importance of the smoothness to the fit to the data. The optic flow is found from this using an iterative approach; Horn and Schunk show that making one iteration per image in a sequence of $n+1$ frames (carrying over the results of the calculations to the next frame as an initial estimate) gives better results than n iterations applied to the measurements from just two frames.

There are two major problems with the approach of Horn and Schunk. One is that the BCCE is only well conditioned at parts of the image which have high gradient, as can be seen in Equation 3. Thus results must be "spread" into those regions with low gradient if the aim of achieving flow estimates at every image position is to be achieved. The other problem is that the optic flow will be smoothed across flow discontinuities, resulting in inaccurate flow estimates. This problem was addressed by Nagel; see [72]. Here the optic flow is only smoothed in the direction perpendicular to the image brightness gradient, so that discontinuity boundaries are much better preserved. In [74], Nagel extends this "oriented smoothness constraint" into the temporal domain, using spatiotemporal partial derivatives of the image brightness function – not just the spatial ones – to determine the best "directions" in which to smooth.

In [72], Nagel also shows that Hildreth's work can be viewed as a special case of the "oriented smoothness" gradient-based approach. Similarly, in [71], Nagel discusses the relation between the flow found from using gradient-based methods and that found from using two dimensional features. He shows how the estimate of the error in the flow should vary at and around corners.

In an approach *similar* to gradient-based methods Heeger (see [50]) used spatiotemporal filters to find optic flow at every point in the image. The filters used here are different from those used by Buxton and Buxton (described above); they are spatiotemporal Gabor filters, created by multiplying spatiotemporal Gaussian functions with trigonometric functions to achieve band limiting and selection in the spatial and the frequency domains. Orthogonal Gabor filters are used to find the Gabor energy at 12 different orientations and at several different spatial frequencies. The strongest local velocity orientation is found, to give an estimate of image flow. The results are good, but again, there is smoothing of the flow at flow boundaries.

1.3 The Correlation-Based Approach

The idea of the correlation of parts of one image with parts of the next (in order to find image flow) has been used more often as *part* of flow estimating algorithms than as a single method in its own right. The method does not perform the data reduction of the feature-based techniques, and is computationally expensive. Also, as with gradient-based methods, correlation-based flow will be the most well conditioned near image features. However, in [16] and [6], Burt et. al. use correlation at multiple scales to find optic flow in cases where multiple motion may be occurring. In [122] correlation and image warping is used whilst moving downwards through scale space to find image flow. See also the references in [56].

More often, such as in [56], [4] and [93], correlation is used to aid the matching of image features or to find image motion once features have been determined.

2 Image Segmentation and Object Tracking

The segmentation of image sequences into differently moving objects (or, more correctly, their projections in the image) is one of the aims of this thesis. In this section previous work on segmentation is reviewed. The different approaches taken in the past could be presented in an order that depended either on the method used to find segmentation or on the assumptions used about the world, the objects in it, the motion of the camera, and the motions of the objects. The former approach is taken, to give greater continuity to the discussion.

The segmentation methods lie in (or between) two groups; those detecting flow discontinuities (local operations) and those detecting patches of self-consistent motion according to set criteria (global measurements). It is usually pointed out by proponents of the latter approach that the former is very sensitive to noisy flow measurements. The methods of statistical regularization and image transformation fall somewhere in between these groups, attempting to achieve global minimization of locally defined cost functions, or finding best fits to image transformations respectively.

Much of the segmentation research reviewed here relies on the assumption that object motion is restricted to certain movement types, most typically, translational motion. For obvious reasons a more general approach was preferred in the work reported in this thesis, so that a wider range of world events could be correctly handled.

Note should be taken of how little work has been undertaken which involves the temporal integration, or tracking, of segmentation results. The research described in this thesis covers this area as well as the segmentation, that is, a temporally coherent list of segmented objects is maintained as time proceeds, and objects move about in the image.

2.1 Segmentation Based on Local Measurements

In [106] Spoerri and Ullman claim to segment the images before flow is found, using various flow discontinuity tests. (It seems unlikely that this is achieved, as edge-based motion is used to drive the local discontinuity tests.) The sort of tests which are used are bimodality in local normal flow histograms and appearance/disappearance of thin image segments, requiring fine image texture. Only very simple images are used for testing. In [81] Overington utilizes the normal components of flow computed at edges (using gradient-based techniques) to find discontinuities in the normal flow component. The discontinuities are used to detect moving objects in a scene, taken from a static camera.

In [120] Waxman and Duncan discuss the use of stereo and motion for recovering information about the world. In their experiments they tracked dots (painted onto objects) by assuming that image motion was less than dot separation. They attempt to segment spatially discontinuous scenes by finding boundaries of "regions of analyticity" in the flow. Even with the artificially aided (and therefore hopefully accurate) way of recovering optic flow used, the segmentation boundaries are poor.

2.2 Segmentation Based on Simple Clustering or Inconsistency with Background Flow

In [64] Meygret and Thonnat match edge contours at multiple scales to give image flow, using stereo sequences. They then link contours into groups with locally consistent flow, assuming translational movement. This produces patches of "separate motion". The results presented suggest that the large areas where no linked edges are present (and therefore where no motion or segmentation information is available) could be better represented if the motion of two dimensional features were used. For example, road markings appear as separate objects (from each other and the background) because they are unlinked in the image.

In [110] Thompson attempts to find moving objects in situations where only translational *image* motion is taking place. Thus no divergence, rotation, or deformation is allowed. The BCCE is used to find the normal component of flow, and the flow measurements are combined with image contrast information to aid the determination of object outlines. The outlines are found using region merging on the basis of similarity in brightness (and consistency in flow or lack of flow information) or similarity in flow.

In [111] Thompson and Pong note that discontinuities (and more general variation) in flow can be due to independent motion *or* depth discontinuities. To help make the distinction they discuss a series of additional constraints, such as ensuring knowledge of the camera's motion, or assuming that the world is planar. They then use these constraints to find moving objects by virtue of local discontinuities in flow. The flow is found using the method of Barnard and Thompson described in [4], that is, the motion of corners.

In [60] McLauchlan et. al. assume zero camera translation and known rotation (a valid assumption in their case, as they are using a pan-tilt-verge stereo head fixed in space). In this case the background flow can be completely subtracted out, leaving only moving objects with non-zero flow. Normal flow is found using the BCCE. Because of the simplicity of the algorithm and also the use of dedicated and parallel hardware, this algorithm runs in real time.

Nelson and Aloimonos, in [77], use very coarse measurements of image flow divergence from spatiotemporal filters to give "time-to-impact" estimates. In [76] Nelson describes a system similar to that of McLauchlan et. al. which finds the normal flow using the BCCE, and determines those parts of the image which are not consistent with the rigid background motion. This is done by comparing the normal flow field with simple precomputed possible motions, allowing the background flow to be cancelled. No attempt to provide image flow segmentation into differently moving objects is made, or to model or track shapes over time. Also in this paper a method is proposed which detects motion by virtue of image acceleration, which is assumed to be high only for independently moving objects. This will clearly work only for objects which have high acceleration in the world. Both of these systems are implemented to run at approximately half real time speed.

2.3 Segmentation Based on Analytic Image Transformations

Segmentation using image transformations finds fits within the flow field of analytic functions with a number of parameters. The number of parameters used varies widely across the research field. The assumption may be made of constant flow, or linearly varying flow etc., and the image is divided up into regions conforming internally to their own fit to the model. In top-down segmentation, the primary fit to the whole image is found first, and then recomputed once outliers to this fit have been thrown away. Then each region of outliers has its own fit made, and so on. In bottom-up segmentation, regions are grown outwards as the model allows. The fit used may be based on a three dimensional world model, usually giving a large number of parameters in the projected two dimensional fit.

In [1] Adiv uses flow fields found either from spatiotemporal derivatives or by using Lawton's method described in [56]. Here the flow field is segmented into regions corresponding to planar surfaces in the environment. These surfaces are then merged into rigid objects. Object outlines are found, but not very accurately.

Blostein and Huang, in [10] and [11], look at the detection of very small objects (typically one pixel in size) moving with constant image velocity. They track these objects over many frames using a tree-search algorithm. Given this restrictive set of assumptions about the environment their algorithms successfully found the tracks of points moving in noisy images which humans could not see. In [28] Debrunner and Ahuja use this method of tracking feature points to segment image sequences into rigid objects of constant rotation per frame. This is done by taking point trajectories and splitting regions into separately moving sub-regions when the motion estimate (which finds the translation and the fixed rotation) fits the measured motion poorly, i.e. when a region clearly contains two independent regions. The algorithm allows regions to merge as well as split, if their motion parameters are similar. A problem (shown in the results) with using this world model is that static points may be linked with rotating regions if they lie on the projection in the image of the rotation axis.

In [32] Diehl models two dimensional transformations of parabolic world surfaces. (There is no merging of adjacent surfaces with similar three dimensional motion into single objects. For example, two faces of a cube will be treated as being two separate objects by the model of parabolic object surfaces.) He finds image motion by warping and differencing, estimating the image transformation of the largest part of the image to do the warping, and then calculating those parts of the image which do not fit this transformation. The algorithm is then applied to the smaller parts to find their image motion. This type of approach has problems if any two (or more) adjacent image sections at the same hierarchical level have similar area, as the transformation estimation will not be well conditioned. This could easily occur in outdoor scenes.

The work of Burt, Bergen et. al. uses a similar segmentation approach. In [16] and [15] the flow field is hierarchically segmented into background and moving objects using coarse to fine flow estimation and coarse to fine segmentation. Again, this works best when moving objects take up a small fraction of the image. The flow is found using correlation, spatiotemporal derivatives or Fourier domain phase shifts (similar in practice to spatiotemporal derivatives). Roughly planar surfaces are assumed, giving rise to an affine flow model being used. In [6] two motions in an image are found "simultaneously"; the method works even when the motions are superimposed across the image (for example semi-transparent motion). In practice the two motions are computed in alternating iterations of the algorithm estimating the affine transformations. The "second" motion at any stage is found by taking the current estimate of the first motion and using this to perform image registration (warping one image to match the second according to the motion transformation) and differencing. In [54] Irani et. al. use these segmentation methods with temporal integration of the segmentation results to improve performance, in cases where different regions at the same hierarchical levels have very different sizes. This "temporal integration" does *not* involve any sort of shape tracking or modelling, so that information about scene events is not readily available.

In [113] Torr and Murray assume that the background motion corresponds to an affine image transformation, and segment out parts of the image which have flow inconsistent with this transformation. The transformation is applied directly to the spatiotemporal image brightness derivatives. This method relies on linear variations in the background flow, and therefore becomes better conditioned as the world becomes flatter. Good segmentation results are presented. It is not clear how well the method would work in the problem cases mentioned above.

2.4 Segmentation Based on Regularization

Methods based on statistical regularization[1] typically set up a definition of ideal image segmentation, normally using Bayesian maximum *a posteriori* (MAP) criteria, and an energy cost functional made up of two or more components. The image is then modelled as a spatial

[1]The term "statistical regularization" is often used interchangeably with the following: "simulated annealing", "stochastic relaxation" and "global optimization". Simulated annealing, as its name suggests, finds a global minimum of a multi-dimensional function by allowing random perturbations in the current state estimate; the probability of changing state (analogous with temperature) decreases with time. The change to the state is accepted if the total cost function is decreased by it. As the "temperature" reaches zero, the system *should* settle into the global minimum.

or spatiotemporal Markov random field (MRF), and iterative minimization is used to find the globally optimal fit to the definition. This approach is usually computationally expensive.

In [69] Murray and Buxton developed the use of statistical regularization for motion segmentation. The assumption is made that the objects are roughly planar and that normal components of flow are available. The cost functional used has the following components: a term measuring the spatial variation in the segmentation labelling, a term measuring the error between the current motion estimate and the normal flow field, a term representing "line processes" (i.e. motion boundaries) and a term measuring the temporal variation in the segmentation labelling. The functional is minimized iteratively, using simulated annealing, starting with a "random" guess at the solution.

In [13] Bouthemy and Lalande use this method on real images taken with a static camera, in order to find moving vehicles. A similar cost functional to that of Murray and Buxton is used (without line processes). Very simple labelling is applied; each image point is either static or moving. In [41] François and Bouthemy extend the method to segment images taken from a moving camera; the BCCE is used to find normal optic flow. Regions are assumed to have linearly varying flow, and are labelled according to their motion *type* (i.e. qualitatively estimated as having divergence, rotation or shear) in a manner similar to the approach of Verri discussed below. In [63] Meyer and Bouthemy use this approach to achieve segmentation, and then track the moving objects over time. The objects are assumed to be convex, and are matched over time using a polygonal representation. The number of vertices used to track each object is fixed at the instantiation of that object and is not allowed to vary; if the number of vertices becomes unsuitable for accurate representation of the object's image projection then the model must be restarted. Also the scale of the polygon is fixed, so that objects are not allowed to change size in the image. Results are only presented for a sequence taken from a static camera.

In [7] Black uses a much more complicated cost functional than the methods already discussed. He combines measures similar to those of Murray and Buxton with intra-frame intensity information. The intra-frame measures are similar to those used in the weak-membrane approach of [9]. Thus image boundaries are combined with motion information. The problem of the speed of simulated annealing algorithms is reduced by taking the solution (after a relatively low number of iterations) at one frame as the starting state estimate of the next frame in the sequence, by warping the solution using estimated image motion. The results show which image features are likely to be "physically significant". In a simple example, the results were good, but in a more complicated sequence, some "insignificant" edges were not suppressed, whilst some flow discontinuities were lost.

2.5 Segmentation Based on Globally Organized Clustering

In [117] Verri et. al. look for regions of uniform flow or uniform expansion, rotation or shear. Optic flow is found using a method derived from the BCCE. This is used to segment the image into different objects. The work is extended in [86] where global segmentation is used. Here each small patch in the image has its centre of expansion/contraction, centre of rotation and direction of constant flow calculated. (Zero shear is assumed.) These three positions are drawn onto three global maps, and strong clusters are looked for in each map. Next each image patch is assigned to a cluster found in one of the three graphs. The results are smoothed using deterministic relaxation, providing as output the segmented image. The results presented do not give very accurate motion boundaries.

3 Edges

There has been an abundance of work on different approaches to the detection of one dimensional features in images. The wide interest is due to the large number of vision applications which use edges[2] and lines as primitives, to achieve higher level goals. This section describes some of the approaches taken in the past. For other reviews of edge finding, see [3], [20], [78] and [48]. An attempt has been made to group the edge detectors into several different types.

3.1 Early Uses of the First Image Brightness Derivative

Some of the earliest methods of finding edges in images used small convolution masks to approximate the first derivative of the image brightness function, thus enhancing edges. Roberts (see [85]) used masks of size two pixels by two pixels to find orthogonal derivatives; $\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$ and $\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$. Prewitt (see [84]) used masks of size three pixels by three pixels in a similar way; $\begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$ and $\begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$. This is preferred to Roberts' approach because of the slightly greater amount of noise suppression, the fact that the gradient image is not shifted by half a pixel in both directions, and because extension to higher image dimensions is not possible

[2]Much research in this field has assumed that the only one dimensional features of interest are step edges. However, there exist many other types of feature. These include lines (ridges in the image surface), ramp ends and roof edges. There are three main reasons for the concentration on step edges. The first is that they are the most common type of one dimensional change. The second is that edges containing a step component are the most well localized one dimensional features, that is, they are formed by a "first order" change. The third reason for working only with step edges is that some proposed edge finders (such as Canny's) are easily extended to finding other types of change once the theory for step edges has been completed. Thus many detectors have been developed using rigorous derivations of optimal algorithms using various criteria based on the model of the ideal step edge.

with the Roberts operator. Sobel (see [105]) used slightly different masks to Prewitt, in that "Gaussian averaging" is performed in one direction, and differentiation is performed in the other; $\begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$ and $\begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$. In [26] the Sobel filter is analyzed in terms of separable two pixel by one pixel components, and the extension to larger filters and more dimensions is made. The image gradient magnitude for all three of these methods is found by taking the square root of the sum of the squares of the outputs from the two orthogonal masks. Likewise, the gradient direction can be found from the ratio of the outputs of the masks.

Other small mask operators have been suggested, such as "compass" operators, which have more than two masks, oriented in several directions. In [48] the responses to gradient magnitude of several small gradient masks are analyzed. It is shown that the choice of mask should depend on the type of image noise present. It is also shown that the isotropy of response does not depend much on the choice of mask.

3.2 Zero Crossings of the Laplacian of a Gaussian of the Image Brightness Function

In [59] Marr and Hildreth proposed the use of zero crossings of the Laplacian of a Gaussian (LoG), that is, the use of the rotationally symmetric convolution filter,

$$\begin{aligned} f(x,y) &= A\nabla^2 e^{-\frac{x^2+y^2}{2\sigma^2}} \\ &= A\left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}\right) e^{-\frac{x^2+y^2}{2\sigma^2}} \\ &= A'\left(2-\frac{x^2+y^2}{\sigma^2}\right) e^{-\frac{x^2+y^2}{2\sigma^2}}, \end{aligned} \tag{5}$$

where $f(x,y)$ is the convolution filter, A and A' are constants and σ determines the spatial scale of the Gaussian. The LoG convolution can be approximated by convolution with the difference of two Gaussians (DoG, sometimes referred to as the "Mexican hat") of appropriate scale and magnitude. The zero crossings can be tested for spurious responses by thresholding the magnitude of the gradient of the zero crossing. The synthesis of results found at different scales is suggested, to enable the detection of a larger range of edge "widths". (See [121] for a description of a successful approach to the synthesis of multiple scale zero crossings, and [78] for a detailed discussion of this subject.)

Contours produced using the LoG filter have the property, convenient for some purposes, of producing closed contours. However, connectivity at junctions is poor, and corners are rounded. Also, the use of non-directional derivatives means that the edge response parallel to an edge is always measured (as well as the expected response perpendicular to it), reducing the signal to noise ratio. The use of directional first and second derivatives improves on this. Finally, the LoG filter gives no indication of edge direction, which may be needed by higher level processes.

3.3 More Advanced Uses of the First Image Brightness Derivative

In [20] Canny described what has since become one of the most widely used edge finding algorithms. The first step taken is the definition of criteria which an edge detector must satisfy, namely, reliability of detection, accuracy of localization and the requirement of one response only per edge. These criteria are then developed quantitatively into a total error cost function. Variational calculus is applied to this cost function to find an "optimal" linear operator for convolution with the image. The optimal filter is shown to be a very close approximation to the first derivative of a Gaussian,[3] i.e., in one dimension,

$$f(x) = Axe^{-\frac{x^2}{2\sigma^2}}. \tag{6}$$

Non-maximum suppression in a direction perpendicular to the edge is applied, to retain maxima in the image gradient. Finally, weak edges are removed using thresholding. The thresholding is applied with hysteresis. Edge contours are processed as complete units; two thresholds are defined, and if a contour being tracked has gradient magnitude above the higher threshold then it is still "allowed" to be marked as an edge at those parts where the strength falls below this threshold, as long as it does not go below the lower value. This reduces streaking in the output edges.

The Gaussian convolution can be performed quickly because it is separable and can be implemented recursively. However, the hysteresis stage slows the overall algorithm down considerably. While the Canny edge finder gives stable results, edge connectivity at junctions is poor,[4] and corners are rounded, as with the LoG filter.

The scale of the Gaussian determines the amount of noise reduction; the larger the Gaussian the larger the smoothing effect. However, as expected, the larger the scale of the Gaussian, the less accurate is the localization of the edge.

Canny also investigated the synthesis of results found at different scales; in some cases the synthesis improved the final output, and in some cases it was no better than direct superposition of the results from different scales.

Finally, Canny investigated the use of "directional operators". Here several masks of different orientation are used with the Gaussian scale larger along the direction parallel to the edge than the scale perpendicular to it. This improves both the localization and the reliability of detection of straight edges; the idea does not work well on edges of high curvature.

---

[3] The problem with using image derivatives is that differentiation enhances noise as well as edge structure, so most edge detectors include a noise reduction stage. Thus the use of the derivative of a Gaussian enables differentiation to take place at the same time as the smoothing; this is allowable, as the two processes commute. The problem of noise enhancement is even worse when differentiation is performed twice.

[4] Some higher level algorithms use Canny's method *because* of this characteristic, as they work better with simple unconnected edges. However, achieving full connectivity at junctions is clearly a worthwhile goal as it correctly represents the scene. In [57] Li et. al. suggest heuristic extensions to the Canny algorithm to enable the joining of open contour ends with nearby contours. This however produces some false edge extensions.

In [29] Deriche uses Canny's criteria to derive a different "optimal operator"; the difference is that the filter is assumed to have infinite extent. The resulting convolution filter is sharper than the derivative of the Gaussian;

$$f(x) = Axe^{-\frac{|x|}{\sigma}}. \tag{7}$$

This is also implementable as a recursive filter for speed.

In [94] Shen and Castan describe another related linear operator; the form is,

$$f(x) = A\frac{x}{|x|}e^{-\frac{|x|}{\sigma}}. \tag{8}$$

Like the Deriche filter, this is implemented recursively and has an infinite support region. The filter is even sharper than that of Deriche; the argument presented is that the larger the scale of the Gaussian, the more planar the central region, giving rise to an "unnecessary" reduction in edge localization. Hence the filter contains a discontinuity at $x = 0$, and information very close to the centre of the filter is given *more* weighting (and not *less*, as is usual) than that from slightly further out. However, it is suggested in [65] that the discontinuity can induce multiple edges. The Shen filter is not separable (in two dimensions), so an approximation to the optimal function must be made.

In practice, the first derivative of the Gaussian, and the Deriche and Shen operators all give very similar results when applied to real images.

In [65] Monga et. al. extend two dimensional linear filters (in particular, the Shen and Deriche filters) to find edges in three dimensional data such as nuclear magnetic resonance scans.

3.4 The Second Directional Derivative of Image Brightness

In [46], Haralick proposes the use of zero crossings of the second directional derivative of the image brightness function. This is theoretically the same as using maxima in the first directional derivatives, and in one dimension is the same as the LoG filter. The zero crossings are found by fitting two dimensional functions to the image brightness surface and then analyzing the resulting fit (see the following section). The functions used are "discrete orthogonal polynomials of up to degree three". There is a problem with "phantom edges" created by the second directional derivative at "staircase structures". Connectivity at junctions is poor.

In [39] Fleck describes the use of the second directional derivative for edge finding, with various extensions to the basic use of zero crossings. The problem of phantom edges is reduced with a test using the first and third derivatives. Image noise is reduced using topological sums; smoothing is achieved at potential edge points by measuring the local support they have for their "edge point type" classification. The overall algorithm (and in particular the noise reduction part) is computationally expensive.

3.5 Fitting Functions to the Image Brightness Surface

The approach of surface fitting defines a two dimensional function which should be "flexible" enough to provide a good approximation to the image brightness function. The best fit of the function to the image is then found at each image position, and the parameters of the fit are used either to estimate image derivatives directly or to find edges using alternative definitions. The problem with the mathematical model approach is that deviations from the surface model cannot (by definition) be accommodated by varying the model's parameters.

In [53] Hueckel used circular windows within which he fitted a single step edge. Clearly this approach does not perform well at junctions, corners or non-step edges.

The approach of Haralick which was described in the previous section also falls into this category, using a polynomial fit to find zero crossings in the second directional derivative.

In [75] Nalwa and Binford use the hyberbolic tangent as the function to fit to what are assumed to be short straight sections of edge. Thus no edge curvature is allowed if an ideal fit is to be achieved. It is proposed that other types of edge (than the step edge) can be detected using combinations of the tanh function. The computation necessary in order to achieve the tanh fit is large; first a planar fit and a cubic fit are made, so that the local patch may be classified into "non-edge" or "potential edge", and so that the edge direction may be found. Next the tanh fit is made, as is a quadratic fit, to allow a further test of the presence of an edge to be made. Finally, the strength and direction of the edge are found from the parameters of the tanh fit, and the strength is thresholded to remove weak edges. As might be expected, edges of high curvature are broken up. A similar fitting method is used in [97] by Sinha and Schunk. They perform two stage fitting; a planar fit is followed by a bicubic spline fit.

3.6 Using Mathematical Morphology

In [78] Noble uses mathematical morphology to find image structure. Several different morphological operations are described; these are used to enhance edges and find two dimensional features. The "erode-dilate" operator is similar to a first order derivative, and the "open-close" operator is similar to a second order derivative. Good quality edges are found by tracking both sides of each edge and then stitching these "half boundaries" together. Connectivity at junctions is good, although spurious short "tails" sometimes appear at structures such as "T" junctions. The algorithm, including edge tracking, is fairly computationally expensive.

3.7 Energy-Based Methods

In [115] and [114] Venkatesh, Owens et. al. describe the approach of using "local energy" (in the frequency domain) to find features. The local energy is found from quadrature pairs of image functions, such as the image function and its Hilbert transform. Fourier transforms of the image were initially used to find the required functions, at large computational cost. In a more "realistic" implementation also described, processing is performed using a function pair very similar to a first and a second derivative; this is shown to be equivalent to the method proposed originally.

This approach has the advantage of being able to find a wider variety of edge types than that typically found using antisymmetric linear filters, however, this is at the expense of optimizing the signal to noise ratio. (In [21] Canny makes a similar point, when discussing the symmetric and antisymmetric components of a filter.)

In [114] the concept of filter idempotency is also discussed, that is, the property of making no changes to the initially processed image when further iterations are applied.

A similar approach to that of Venkatesh, Owens et. al. is described in [83], where Perona and Malik use pairs of filters to detect "composite edges". The emphasis here is on edges which are formed from combinations of different simple types, rather than on a variety of separate simple edge types. The different filters which are combined to give the "composite edge filter" can work at different spatial scales, to match the expected scales of the various edge components. A set of filters is used to find the responses at different orientations. Typically, the second derivative of the Gaussian is used for the symmetric filter, and the Hilbert transform of the image function is used for the antisymmetric filter. The method appears to give edges which are quite fragmented.

In [88] Rosenthaler et. al. use a similar method; a set of oriented Gabor-based filters is used to find local energy. The oriented energy map is used to find edges and two dimensional features (see later). The results presented are fairly good, but junction connectivity is poor.

3.8 Using Regularization to Find Edges

The method of regularization was described in Section 2.4 with regard to image flow segmentation. This type of global image processing can also be used to find edges, via more general surface reconstruction. (In fact, regularization was used with individual images before it was used with image sequences.) These methods tend to be computationally expensive.

In [9] Blake and Zisserman used a weak membrane model which has many similarities to the work described earlier. The elements of the cost functional are the smoothness of the estimated surface, the quality of the fit of the estimated surface to the original data and the "line processes", i.e., the discontinuities in the membrane. This method does not pick up fine complicated detail very well.

In [43] Geiger and Girosi extend this approach to cope with sparse data. In [79] Nordström uses a similar approach from the viewpoint of anisotropic diffusion. Thus rather than solving one global equation, local diffusion is performed iteratively (using similar cost function elements to those mentioned above to control the anisotropy) to achieve image reconstruction. Image edges can be extracted at some stage in the iterative procedure, before the image is homogenized.

In [109] Tan et. al. use a slightly different approach from that of surface reconstruction followed by edge extraction. Firstly, the image is processed with dissimilarity enhancement to enhance edges, and non-maximum suppression is performed. The enhancement is typically region dissimilarity when standard edge finding is to be performed, but texture preprocessing can be used instead to find texture boundaries. Then, minimization is performed to find the best global edge configuration, using five elements for the cost functional; a cost for edge curvature, a cost for region dissimilarity (with edge pixels contributing zero cost), a cost for the number of edge points, a cost for edge fragmentation (i.e., related to the number of edge ends and isolated points), and a cost for edge thickness. As with other minimization methods, the relative weighting of the different terms in the cost functional is difficult to reliably optimize.

3.9 Local Edge Context

The regularization-based approach discussed in the previous section is not the only way of using local edge context to find reliable image edges. Two other methods of using this information are described in [80] and [47].

In [80] Otte and Nagel find the edge gradient response in a Gaussian smoothed image. Then, after non-maximum suppression, a rule-based scheme is used to validate potential edges; the local gradient direction is analyzed, and edge-like continuity in orientation is required for the edge to be reported. This context-based edge thresholding process is used instead of the more usual thresholding based on gradient magnitude. Results suggest that this method can give cleaner output than simple one level thresholding, but connectivity at junctions is not improved.

In [47] Haralick and Lee find the probability of a pixel being an edge point by analyzing possible edge routes through the pixel's locality. An evaluation function for the edge quality of each possible route is defined, and the optimal route is calculated. The method is tested on very simple synthetic images; it is not apparent how well this method would work on complicated (i.e. real) images.

3.10 Other Methods

Finally, some edge finding methods which do not fit any of the above categories are briefly described, for the sake of completeness of the review.

In [91] Shann and Oakley implement Canny's edge finder and the LoG filter in a new way. Instead of convolving the discrete image with a discrete approximation of an edge enhancing filter, they use a (virtual) continuous convolution filter to give a filtered virtual image which is continuous; this is analyzed using numerical methods to find exact edge positions. No results are presented.

In [34] Eichel and Delp use a small scale local operator to measure image variance to enhance edges. Results presented are poor.

In [108] Strickland et. al. assume that the edge profile and the lighting conditions are fixed, and use a correlation-based template matching scheme to find edges. Within the limitations of the assumptions, the results presented are very good. However, the method clearly does not have general applicability.

In [27] Dattatreya and Kanal use one dimensional Kalman techniques to find smooth edge contours. Potential edge points are tracked in the x and y directions independently using Kalman filters. The two sets of results are combined to give an edge map. As with the context dependent approach of Haralick and Lee, it is not clear how junctions would be handled, and no results are presented of complicated images.

4 Two Dimensional Features

Mathematical definitions of image structures that are well localized in two dimensions have been even more diverse in the past than definitions of edges. Much of the work on two dimensional features has concentrated solely on "corners", in name and in practice, that is, features formed at boundaries between only two image brightness regions, where the boundary curvature is "sufficiently high". Much theory has been presented which is only strictly accurate for such simple image structures. However, many other types of localized structure arise, for example, "T", "Y" and "X" junctions, as well as less rigidly defined structures. These will not usually fit into the model[5] of corner finding theory, and so will not be correctly dealt with.

A few methods which find *points of localized image structure* directly (i.e., take the italicized phrase as the "brief" for developing an algorithm) have been used, for example the Moravec

[5] Here *model* means the type of image structure assumed to be present when discussing any particular aspect of corner finding theory. Note that almost all approaches to corner finding assume a simple corner *model*, whether or not they are "model-based".

interest operator and the Plessey algorithm. These are described below. This relaxation of the corner "definition" is to be commended; however, they each have their own shortcomings. In the case of the Plessey feature point detector, the increase in reliability of point detection and in the variety of points which may be detected (when compared with "corner"-based methods, such as Kitchen-Rosenfeld and Dreschler-Nagel) is balanced by a loss in localization accuracy (for two reasons explained below).

The SUSAN detector [101, 104] makes no assumptions about the form of the local image structure around a well localized point. Neither does it look for "points of interest". Instead, it analyzes different regions separately, using direct local measurements, and finds places where individual region boundaries have high curvature, i.e., finds corners formed by single regions. Thus at junctions involving more than two regions, for example, "T" junctions, more than one region may contribute to the detection of a "corner", and all junctions will be correctly processed, no matter how complicated they are. Also, no assumptions about the internal structures of the regions around a "corner" are made; it is common to assume that the regions are constant in value, but in the case of the SUSAN detector, sloping regions are allowed.

In this section a short review of methods for finding corners and feature points is given. No attempt has been make to divide the approaches into categories, as the dividing lines are extremely blurred. Other reviews can be found in [48], where some *corner* detectors are described, and in [78], where a more detailed analysis and discussion is given.

A few methods of using binary edge maps to find corners have been suggested. The edges are detected and then edge curvature is calculated in order to find corner locations. See, for example, [42], [2] and [61]. Clearly this sort of approach has problems with junctions.

In [66] and [67], Moravec developed the idea of using "points of interest". These are defined as occurring when there are large intensity variations in every direction. This definition is realized by computing an un-normalized local autocorrelation in four directions and taking the lowest result as the measure of interest. This response is thresholded and then local non-maxima are suppressed. This method has been used successfully in vision applications; some of these have been mentioned above. However, some problems with the Moravec operator have been identified (principally by the proponents of the Plessey detector, which builds on the same feature definition): the response is anisotropic, due to the use of only four directions for finding the local autocorrelation, the response is noisy, as the profile of the "window" used for finding the autocorrelation is square, and the operator is sensitive to strong edges, as the response is given by the minimum in the autocorrelation measurements, and not by the variation in them.

In [5] Beaudet enhanced high curvature edges (i.e., looked for saddle points in the image brightness surface) by calculating image Gaussian curvature (the product of the two principle curvatures). In [55] Kitchen and Rosenfeld used a local quadratic surface fit to find corners. The parameters of the surface were used to find the gradient magnitude and the rate of change of gradient direction; the product of these quantities was used to define "cornerness", and local maxima were reported as corners. In [33] Dreschler and Nagel defined corners to be points lying between extrema of Gaussian curvature. Later, (for example, see [73]) Nagel used the definition (of corner position) of "the point of maximum planar curvature for the line of steepest slope". In [123] Zuniga and Haralick found corners at significant changes in curvature along edges.

In [78] Noble shows that these four approaches are all making the same underlying measurement, that is, the product of the gradient magnitude and the edge contour curvature. They are all sensitive to noise, and have relatively low reliability, as shown in various tests in [48].

In [92] Shann and Oakley use a method very similar to their edge finding approach described above, i.e., they use virtual continuous filters to find virtual continuous responses. They find positions of zero second derivative measured parallel to the edge (which can be shown to be equivalent to the above methods). It is suggested that this will improve the localization accuracy of the corners. Results are presented only for very simple images, and it is to be expected that the sensitivity to noise and reliability will not be any better than the image curvature corner detectors described above.

In [31] Deriche and Giraudon use Beaudet's "cornerness" measure in conjunction with edge detection to obtain accurate corner localization. Corners are found at two different scales, and lines are drawn between a corner at one scale and the same corner at the other. Then the intersection of this line with the nearest zero crossing of the Laplacian edge is defined as the correct position of the corner. Again, it is not clear whether sensitivity to noise and reliability of detection is improved with this method.

In [118] Wang and Brady develop the above curvature-based methods. As well as requiring that curvature be a maximum and above a threshold, they require that gradient perpendicular to the edge be a maximum and above a threshold. Also false corner response suppression is performed to prevent corners being wrongly reported on strong edges. However, in the presence of large amounts of noise, this is not sufficient to prevent false responses on strong diagonal edges. The corners are found at different smoothing levels allowing an estimation of the corner positions at zero smoothing.

In [49] Harris and Stephens described what has become known as the Plessey feature point detector. This is built on similar ideas to the Moravec interest operator, but the measurement of local autocorrelation is estimated from first order image derivatives. The variation of the autocorrelation over different orientations is found by calculating functions related to the principle curvatures of the local autocorrelation. This well conditioned algorithm gives robust detection, that is, feature points are reliably detected. However, localization accuracy is poor, particularly at certain junction types. This occurs for two reasons. The first is that the method can only work accurately at most junctions if the masks used to find the derivatives and to perform smoothing have zero size. This can be easily understood in the case of "T" junctions, where the two "smaller" regions at the junction have similar brightness, and the "larger" region has a very different brightness from the other two. Clearly, for finite mask sizes, one curvature (that measured perpendicular to the strong edge) will have a maximum value all along that edge. However, the other curvature (that parallel to the strong edge) will have a value which is constant at all points along the weak edge and which starts to drop as the strong edge is approached, the distance at which this begins to be noticeable being approximately half the size of the masks used. Thus the position at which both curvatures are largest is not well defined, and when some compromise position is chosen, it will be displaced away from the real junction position. The other reason for localization error is explained fully in [98], where it is shown that the actual measure of interest depends to a large extent on the relative values of the two curvatures, and is greatest when they are equal. In the case described above, the two curvatures will be equal at some point along the weak edge, away from the strong edge, and therefore away from the correct position of the junction.

In [78] Noble relates the Plessey operator to the four "greylevel corner detectors" mentioned above, and shows that it can be viewed as measuring image curvature also.

In [40] Förstner and Gülch describe a method which appears to use exactly the same measure of "cornerness" as the Plessey operator. However, a more complicated implementation is used; the detection and localization stages are separated, into the selection of windows, in which features are known to reside, and feature location within selected windows. This results in improved localization accuracy with some corner types, but the algorithm is, as one would expect, even slower than the Plessey detector.

In [25] Cooper et. al. use image patch similarity tests in two edge-based corner finding algorithms. In the first method the similarity function (an un-normalized local autocorrelation function) is found at positions in both directions along an edge; if the function gives a high result in either direction parallel to the edge then a corner is reported. This is clearly similar to the principle behind the Moravec and Plessey detectors. In the second method the results of the similarity tests performed along the edge are used to estimate image curvature directly. In both cases the localization accuracy at junctions is poor.

In [78] Noble uses mathematical morphology (as mentioned earlier), applied to the approach of facet-based structure modelling, to propose a two dimensional feature detector. Gaussian curvature is used to determine whether local patches are elliptic, hyperbolic, parabolic or flat; this information is extracted from the image using morphological techniques. It is not clear how well junctions will be dealt with, as no detector is actually developed, and results of testing the morphological approach on two dimensional features are not presented.

In [114] Venkatesh uses the local energy approach discussed earlier to find two dimensional features. Maxima in local energy are searched for along the edge direction. Only tests performed on very simple (and noise free) images are presented.

In [88] Rosenthaler et. al. use the integrated edge and "keypoint" detector (mentioned earlier) to find two dimensional features. Oriented directional energy filters are used to find first and second derivatives of energy along the direction of each filter; these are combined to give a final response which is a maximum at "keypoints". The results presented are good, but performance in the presence of noise is not investigated.

In [58] Liu and W.-H. Tsai assume that only two regions form a corner. The image is divided up into patches, and each patch is segmented into two regions using grey level moment preserving thresholding. The two regions are then analyzed to determine whether a corner exists anywhere within the patch. If it does, the corner's position and angle are found. The method clearly cannot cope with junctions. It has apparently only been tested on very simple images, and does not function very well in the presence of noise.

In [62] Mehrotra et. al. use half-edge detection to find corners. Directional masks which are zero in one half are used to find either gradient maxima or zero crossings of the second derivative. (The line splitting the mask into "zero" and "non-zero" parts is perpendicular to the measured edge direction. Large masks are used at eight orientations.) Then for each edge point the difference in orientation between the neighbouring edge points is used to find corners. Results are presented for very simple images only.

In [87] Rohr uses strict models to detect and localize corners and junctions. The assumptions are made that regions making up junctions have constant underlying values and that junctions are well isolated in the image. Large masks are used to fit local structures to models with several parameters, including the feature position and orientation, the intensities of regions touching at the junction, the amount of blurring, and the relative positions of the edges forming the junction. Thus feature detection from a large scale viewpoint is performed. Results are only presented for very simple low noise images.

In [96] Singh and Shneier attempt to overcome the problem of the trade-off between reliable detection and accurate localization by fusing various corner finding methods. Initially a template-based corner detector is described. Three different methods for performing the matching between the image and the template are given, and it is shown that they have differing degrees of reliability and accuracy. (These two quantities are related to each other for any one method by a qualitative "uncertainty principle".) Fusing the three methods' outputs should give both reliable and accurate results. In the second part of the paper, a gradient-based approach is taken, similar to that of Kitchen and Rosenfeld. Different methods are described, each using a slightly different definition of "cornerness". Again, the different methods are fused to provide a final list of detected corners. Results presented appear poor.

In [82] Paler et. al. note that at a corner the median of local brightness values taken over a small mask is significantly different from the centre value. Thus the difference between the median and the centre value is used to produce a corner response. This approach is limited to situations where the edge widths and the contrast between "the object and the background" can be accurately estimated. Junctions cannot be dealt with at all.

References

[1] G. Adiv. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 7(4):384–401, 1985.

[2] H. Asada and M. Brady. The curvature primal sketch. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(1):2–14, 1986.

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

[4] S.T. Barnard and W.B. Thompson. Disparity analysis of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2(4):333–340, 1980.

[5] P.R. Beaudet. Rotational invariant image operators. In *Proc. of the Int. Conf. on Pattern Recognition*, pages 579–583, 1978.

[6] J.R. Bergen, P.J. Burt, R. Hingorani, and S. Peleg. Computing two motions from three frames. In *Proc. 3rd Int. Conf. on Computer Vision*, pages 27–32, 1990.

[7] M.J. Black. Combining intensity and motion for incremental segmentation and tracking over long image sequences. In *Proc. 2nd European Conf. on Computer Vision*, pages 485–493. Springer-Verlag, 1992.

[8] A. Blake and R. Cipolla. Robust estimation of surface curvature from deformation of apparent contours. *Image and Vision Computing*, 9(2):107–112, 1991.

[9] A. Blake and A. Zisserman. *Visual Reconstruction.* MIT Press, Cambridge, USA, August 1987.

[10] S.D. Blostein and T.S. Huang. A tree-search algorithm for target detection in image sequences. Technical Report ISP-450, University of Illinois, 1987.

[11] S.D. Blostein and T.S. Huang. Detection of small moving objects in image sequences using multistage hypothesis testing. In *Int. Conf. on Acoustics, Speech and Signal Processing*, 1988.

[12] R.C. Bolles, H.H. Baker, and D.H. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *Int. Journal of Computer Vision*, 1:7–55, 1987.

[13] P. Bouthemy and P. Lalande. Detection and tracking of moving objects based on a statistical regularization method in space and time. In O. Faugeras, editor, *Proc. 1st European Conf. on Computer Vision*, pages 307–311. Springer-Verlag, 1990.

[14] W. Burger and B. Bhanu. Estimating 3-D egomotion from perspective image sequences. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(11):1040–1058, 1990.

[15] P.J. Burt. Image motion analysis made simple and fast, one component at a time. In *Proc. 2nd British Machine Vision Conference*, 1991.

[16] P.J. Burt, J.R. Bergen, R. Hingorani, R.J. Kolczynski, W.A. Lee, A. Leung, J. Lubin, and J. Shvaytser. Object tracking with a moving camera; an application of dynamic motion analysis. In *IEEE Workshop on Visual Motion*, Irvine, CA., 1989.

[17] B.F. Buxton and H. Buxton. Low-level edge detection for the extraction of optical flow data and passive ranging. Technical Report 3, Long Range Research Laboratory, GEC, Wembley, 1983.

[18] B.F. Buxton and H. Buxton. Monocular depth perception from optical flow by space time signal processing. *Proc. Royal Society of London*, B 218:27–47, 1983.

[19] B.F. Buxton, H. Buxton, D.W. Murray, and N.S. Williams. Machine perception of visual motion. Technical Report 3, Long Range Research Laboratory, GEC, Wembley, 1985.

[20] J.F. Canny. Finding edges and lines in images. Master's thesis, MIT, Cambridge, USA, 1983.

[21] J.F. Canny. A computational approach to edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(6):679–698, November 1986.

[22] D.A. Castelow, D.W. Murray, G.L. Scott, and B.F. Buxton. Matching Canny edgels to compute the principal components of optic flow. In *3rd Alvey Vision Conference*, pages 193–200, September 1987.

[23] D. Charnley and R.J. Blissett. Surface reconstruction from outdoor image sequences. *Image and Vision Computing*, 7(1):10–16, 1989.

[24] H.H. Chen and T.S. Huang. Matching 3-D line segments with applications to multiple-object motion estimation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(10):1002–1008, October 1990.

[25] J. Cooper, S. Venkatesh, and L. Kitchen. Early jump-out corner detectors. Technical Report 90/14, University of Western Australia, Department of Computer Science, 1990.

[26] P.-E. Danielsson. Generalized and separable Sobel operators. In H. Freeman, editor, *Machine Vision for Three-Dimensional Scenes*. Academic Press, 1990.

[27] G.R. Dattatreya and L.N. Kanal. Detection and smoothing of edge contours in images by one-dimensional Kalman techniques. *IEEE Trans. on Systems, Man and Cybernetics*, SMC-20(1):159–165, 1990.

[28] C. Debrunner and N. Ahuja. Motion and structure factorization and segmentation of long multiple motion image sequences. In *Proc. 2nd European Conf. on Computer Vision*, pages 217–221. Springer-Verlag, 1992.

[29] R. Deriche. Using Canny's criteria to derive a recursively implemented optimal edge detector. *Int. Journal of Computer Vision*, 1(2):167–187, 1987.

[30] R. Deriche and O. Faugeras. Tracking line segments. In O. Faugeras, editor, *Proc. 1st European Conf. on Computer Vision*, pages 259–268. Springer-Verlag, 1990.

[31] R. Deriche and G. Giraudon. Accurate corner detection: An analytical study. In *Proc. 3rd Int. Conf. on Computer Vision*, pages 66–70, 1990.

[32] N. Diehl. Segmentation and motion estimation in image sequences. *SPIE 1260; Sensing and Reconstruction of Three-Dimensional Objects and Scenes*, pages 250–261, 1990.

[33] L. Dreschler and H.-H. Nagel. Volumetric Model and 3D Trajectory of a Moving Car Derived from Monocular TV-frame Sequence of a Street Scene. *Computer Vision, Graphics and Image Processing*, 20(3):199–228, 1981.

[34] P.H. Eichel and E.J. Delp. Quantitative analysis of a moment-based edge operator. *IEEE Trans. on Systems, Man and Cybernetics*, SMC-20(1):59–66, 1990.

[35] W. Enkelmann. Obstacle detection by evaluation of optical flow fields from image sequences. In *Proc. 1st European Conf. on Computer Vision*, pages 134–138, 1990.

[36] O.D. Faugeras. On the motion of 3D curves and its relationship to optical flow. In *Proc. 1st European Conf. on Computer Vision*, pages 107–117, 1990.

[37] O.D. Faugeras, F. Lustman, and G. Toscani. Motion and structure from motion from point and line matches. In *Proc. 1st Int. Conf. on Computer Vision*, pages 25–34, 1987.

[38] O.D. Faugeras and S. Maybank. Motion from point matches: Multiplicity of solutions. *Int. Journal of Computer Vision*, 4:225–246, 1990.

[39] M.M. Fleck. Spectre: an improved phantom edge finder. In *Proc. 5th Alvey Vision Conference*, pages 127–132, 1989.

[40] W. Förstner and E. Gülch. A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *ISPRS Intercommission Workshop, Interlaken*, pages 149–155, June 1987.

[41] E. François and P. Bouthemy. Multiframe-based identification of mobile components of a scene with a moving camera. In *Proc. Conf. Computer Vision and Pattern Recognition*, 1991.

[42] H. Freeman and L.S. Davis. A corner finding algorithm for chain code curves. *IEEE Trans. on Computers*, 26:297–303, 1977.

[43] D. Geiger and F. Girosi. Parallel and deterministic algorithms from MRFs: Surface reconstruction and integration. In *Proc. 1st European Conf. on Computer Vision*, pages 89–98, 1990.

[44] S. Gong and M. Brady. Parallel computation of optic flow. In O. Faugeras, editor, *Proc. 1st European Conf. on Computer Vision*, pages 124–133. Springer-Verlag, 1990.

[45] K.J. Hanna and L. Tarassenko. Tracking cataract by the four-line method. In *4th Alvey Vision Conference*, pages 253–258, 1988.

[46] R.M. Haralick. Digital step edges from zero crossing of second directional derivatives. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6(1):58–68, January 1984.

[47] R.M. Haralick and J.S.J. Lee. Context dependent edge detection and evaluation. *Pattern Recognition*, 23:1–19, 1990.

[48] R.M. Haralick and L.G. Shapiro. *Computer and Robot Vision*, volume 1. Addison-Wesley, 1992.

[49] C.G. Harris and M. Stephens. A combined corner and edge detector. In *4th Alvey Vision Conference*, pages 147–151, 1988.

[50] D.J. Heeger. Optical flow from spatiotemporal filters. *Int. Journal of Computer Vision*, pages 279–302, 1988.

[51] E.C. Hildreth. Computations underlying the measurement of visual motion. *Artificial Intelligence*, 23:309–354, 1984.

[52] B.K.P. Horn and B.G. Schunk. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[53] M.H. Hueckel. An operator which locates edges in digitized pictures. *Journal of the Association for Computing Machinery*, 18(1):113–125, 1971.

[54] M. Irani, B. Rousso, and S. Peleg. Detecting and tracking multiple moving objects using temporal integration. In *Proc. 2nd European Conf. on Computer Vision*, pages 282–287. Springer-Verlag, 1992.

[55] L. Kitchen and A. Rosenfeld. Gray-level corner detection. *Pattern Recognition Letters*, 1:95–102, 1982.

[56] D.T. Lawton. Processing translational motion sequences. *Computer Vision, Graphics and Image Processing*, 22:116–144, 1983.

[57] D. Li, G.D. Sullivan, and K.D. Baker. Edge detection at junctions. In *Proc. 5th Alvey Vision Conference*, pages 121–125, 1989.

[58] S.-T. Liu and W.-H. Tsai. Moment-preserving corner detection. *Pattern Recognition*, 23:441–460, 1990.

[59] D. Marr and E.C. Hildreth. Theory of edge detection. *Proc. Roy. Soc. London.*, B-207:187–217, 1980.

[60] P.F. McLauchlan, I. Reid, and D.W. Murray. Coarse image motion for saccade control. In *Proc. 3rd British Machine Vision Conference*, 1992.

[61] G. Medioni and Y. Yasumoto. Corner detection and curve representation using cubic b-splines. *Computer Vision, Graphics and Image Processing*, 39:267–278, 1987.

[62] R. Mehrotra, S. Nichani, and N. Ranganathan. Corner detection. *Pattern Recognition*, 23:1223–1233, 1990.

[63] F. Meyer and P. Bouthemy. Region-based tracking in an image sequence. In *Proc. 2nd European Conf. on Computer Vision*, pages 476–484. Springer-Verlag, 1992.

[64] A. Meygret and M. Thonnat. Segmentation of optical flow and 3D data for the interpretation of mobile objects. In *Proc. 3rd Int. Conf. on Computer Vision*, pages 238–245, 1990.

[65] O. Monga, R. Deriche, G. Malandain, and J.P. Cocquerez. Recursive filtering and edge tracking: Two primary tools for 3D edge detection. *Image and Vision Computing*, 9(4):203–214, 1991.

[66] H.P. Moravec. Towards automatic visual obstacle avoidance. In *Proc. of the International Joint Conference on Artificial Intelligence*, page 584, 1977.

[67] H.P. Moravec. Visual mapping by a robot rover. In *Proc. of the 6th International Joint Conference on Artificial Intelligence*, pages 598–600, 1979.

[68] D.W. Murray. Algebraic polyhedral constraints and 3D structure from motion. In *Proc. 5th Alvey Vision Conference*, pages 215–220, 1989.

[69] D.W. Murray and B.F. Buxton. Scene segmentation from visual motion using global optimization. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9(2):220–228, March 1987.

[70] D.W. Murray, D.A. Castelow, and B.F. Buxton. From an image sequence to a recognized polyhedral object. In *Proc. 3rd Alvey Vision Conference*, pages 201–210, 1987.

[71] H.-H. Nagel. Displacement vectors derived from second order intensity variations in image sequences. *Computer Vision, Graphics and Image Processing*, 21:85–117, 1983.

[72] H.-H. Nagel. On the estimation of optic flow: Relations between different approaches and some new results. *Artificial Intelligence*, 33:299–324, 1987.

[73] H.-H. Nagel. Principles of (low level) computer vision. In J.P. Haton, editor, *Fundamentals in computer understanding: speech and vision*, pages 113–139. Cambridge University Press, 1987.

[74] H.-H. Nagel. Extending the 'oriented smoothness constraint' into the temporal domain and the estimation of derivatives of optical flow. In *Proc. 1st European Conf. on Computer Vision*, pages 139–148, 1990.

[75] V.S. Nalwa and T.O. Binford. On detecting edges. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(6):699–714, November 1986.

[76] R.C. Nelson. Qualitative detection of motion by a moving observer. In *Proc. Conf. Computer Vision and Pattern Recognition*, 1991.

[77] R.C. Nelson and J. Aloimonos. Using flow field divergence for obstacle avoidance: towards qualitative vision. In *Proc. 2nd Int. Conf. on Computer Vision*, pages 188–196, 1988.

[78] J.A. Noble. *Descriptions of Image Surfaces*. D.Phil. thesis, Robotics Research Group, Department of Engineering Science, Oxford University, 1989.

[79] N. Nordström. Biased anisotropic diffusion – a unified regularization and diffusion approach to edge detection. In *Proc. 1st European Conf. on Computer Vision*, pages 18–27, 1990.

[80] M. Otte and H.-H. Nagel. Extraction of line drawings from gray value images by non-local analysis of edge element structures. In *Proc. 2nd European Conf. on Computer Vision*, pages 687–695. Springer-Verlag, 1992.

[81] I. Overington. Gradient-based flow segmentation and location of the focus of expansion. In *Proc. 3rd Alvey Vision Conference*, pages 169–177, 1987.

[82] K. Paler, J. Föglein, J. Illingworth, and J. Kittler. Local ordered grey levels as an aid to corner detection. *Pattern Recognition*, 17(5):535–543, 1984.

[83] P. Perona and J. Malik. Detecting and localizing edges composed of steps, peaks and roofs. In *Proc. 3rd Int. Conf. on Computer Vision*, pages 52–57, 1990.

[84] J.M.S. Prewitt. Object enhancement and extraction. In B.S. Lipkin and A. Rosenfeld, editors, *Picture Processing and Psychopictorics*. Academic Press, 1970.

[85] L.G. Roberts. Machine perception of three dimensional solids. In J.T. Tippet, editor, *Optical and Electro-optical Information Processing*, pages 159–197. MIT Press, 1965.

[86] A. Rognone, M. Campani, and A. Verri. Identifying multiple motions from optical flow. In *Proc. 2nd European Conf. on Computer Vision*, pages 258–266. Springer-Verlag, 1992.

[87] K. Rohr. Modelling and identification of characteristic intensity variations. *Image and Vision Computing*, 10(2):66–76, March 1992.

[88] L. Rosenthaler, F. Heitger, O. Kübler, and R. von der Heydt. Detection of general edges and keypoints. In *Proc. 2nd European Conf. on Computer Vision*, pages 78–86. Springer-Verlag, 1992.

[89] G. Sandini and M. Tistarelli. Active tracking strategy for monocular depth inference over multiple frames. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(1), January 1990.

[90] G.L. Scott. 'Four-line' method of locally estimating optic flow. *Image and Vision Computing*, 5(2):67–72, 1987.

[91] R.T. Shann and J.P. Oakley. Novel approach to boundary finding. *Image and Vision Computing*, 8(1):32–36, February 1990.

[92] R.T. Shann and J.P. Oakley. Local method for curved edges and corners. In *Proc. 2nd British Machine Vision Conference*, pages 319–322, 1991.

[93] L.S. Shapiro, H. Wang, and J.M. Brady. A matching and tracking strategy for independently-moving, non-rigid objects. In *Proc. 3rd British Machine Vision Conference*, 1992.

[94] J. Shen and S. Castan. An optimal linear operator for step edge detection. *Computer Vision, Graphics and Image Processing*, 54(2):112–133, March 1992.

[95] D. Sinclair, A. Blake, S. Smith, and C. Rothwell. Planar region detection and motion recovery. In *Proc. 3rd British Machine Vision Conference*, 1992.

[96] A. Singh and M. Shneier. Grey level corner detection: A generalization and a robust real time implementation. *Computer Vision, Graphics and Image Processing*, 51:54–69, 1990.

[97] S.S. Sinha and B.G. Schunk. A two-stage algorithm for discontinuity-preserving surface reconstruction. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(1):36–55, 1992.

[98] S.M. Smith. Extracting information from images. First year D.Phil. report, Robotics Research Group, Department of Engineering Science, Oxford University, June 1990.

[99] S.M. Smith. *Feature Based Image Sequence Understanding*. D.Phil. thesis, Robotics Research Group, Department of Engineering Science, Oxford University, 1992.

[100] S.M. Smith. ASSET-2: Real-time motion segmentation and object tracking. Internal Technical Report TR95SMS2, Defence Research Agency, Chobham Lane, Chertsey, Surrey, UK, 1995. Available at www.fmrib.ox.ac.uk/~steve for downloading.

[101] S.M. Smith. SUSAN – a new approach to low level image processing. Internal Technical Report TR95SMS1, Defence Research Agency, Chobham Lane, Chertsey, Surrey, UK, 1995. Available at www.fmrib.ox.ac.uk/~steve for downloading.

[102] S.M. Smith. ASSET-2: Real-time motion segmentation and object tracking. *Journal of Real Time Imaging*, 1997. Special issue on Real-Time Motion Analysis, In publication.

[103] S.M. Smith and J.M. Brady. ASSET-2: Real-time motion segmentation and shape tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(8):814–820, 1995.

[104] S.M. Smith and J.M. Brady. SUSAN - a new approach to low level image processing. *Int. Journal of Computer Vision*, 23(1):45–78, May 1997.

[105] I. Sobel. An isotropic 3×3 image gradient operator. In H. Freeman, editor, *Machine Vision for Three-Dimensional Scenes*, pages 376–379. Academic Press, 1990.

[106] A. Spoerri and S. Ullman. The early detection of motion boundaries. In *Proc. 1st Int. Conf. on Computer Vision*, pages 209–218, 1987.

[107] M. Stephens and C. Harris. 3D wire-frame integration from image sequences. In *4th Alvey Vision Conference*, pages 159–165, 1988.

[108] R.N. Strickland, T. Draelos, and Z. Mao. Edge detection in machine vision using a simple L_1 norm template matching algorithm. *Pattern Recognition*, 23:411–421, 1990.

[109] H.L. Tan, S.B. Gelfand, and E.J. Delp. A cost minimization approach to edge detection using simulated annealing. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(1):3–18, January 1980.

[110] W.B. Thompson. Combining motion and contrast for segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2(6):543–549, 1980.

[111] W.B. Thompson and T.-C. Pong. Detecting moving objects. In *Proc. 1st Int. Conf. on Computer Vision*, pages 201–208, 1987.

[112] C. Tomasi and T. Kanade. Shape and motion without depth. In *Proc. Image Understanding Workshop*, 1990.

[113] P.H.S. Torr and D.W. Murray. Statistical detection of independent movement from a moving camera. In *Proc. 3rd British Machine Vision Conference*, 1992.

[114] S. Venkatesh. *A Study of Energy Based Models for the Detection and Classification of Image Features*. PhD thesis, The University of Western Australia, Department of Computer Science, 1990.

[115] S. Venkatesh and R. Owens. An energy feature detection scheme. In *Proceedings, IEEE Int. Conf. on Image Processing, Singapore*, pages 553–557, September 1989.

[116] A. Verri and T. Poggio. Against quantitative optical flow. In *Proc. 1st Int. Conf. on Computer Vision*, pages 171–180, 1987.

[117] A. Verri, S. Uras, and E. De Micheli. Motion segmentation from optical flow. In *Proc. 5th Alvey Vision Conference*, pages 209–214, 1989.

[118] H. Wang and J.M. Brady. Corner detection with subpixel accuracy. Technical Report OUEL 1925/92, Dept. Engineering Science, University of Oxford, 1992.

[119] H. Wang, M. Brady, and I. Page. A fast algorithm for computing optic flow and its implementation on a transputer array. In *Proc. 1st British Machine Vision Conference*, pages 175–180, 1990.

[120] A.M. Waxman and J.H. Duncan. Binocular image flows : Steps toward stereo-motion fusion. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(6):715–729, 1986.

[121] A.P. Witkin. Scale-space filtering. In *Proc. IJCAI 1983*, pages 1019–1021, 1983.

[122] A.P. Witkin, D. Terzopoulos, and M. Kass. Signal matching through scale space. In *5th National Conference on AI*, 1986.

[123] O.A. Zuniga and R.M. Haralick. Corner Detection Using the Facet Model. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 30–37, 1983.