IEEE/RSJ International Workshop on Intelligent
Robots and Systems IROS '91, Nov. 3-5, 1991,
Osaka, Japan. IEEE Cat. No. 91TH0375-6

On the usefulness of optical flow for robotic part manipulation

K. Dawson†, M. Tistarelli†, D. Vernon†

†Trinity College, Dublin, Ireland and †University of Genova, Italy.

Abstract

This paper describes experience gained in using optical flow fields, arising from constrained camera motion, to estimate range information, for robotic part manipulation. Several key issues in robot vision are addressed. These include the computation of optical flow, computation of depth, interpolation of depth values, model construction, object recognition, and pose estimation. The paper provides an example of the approach and evaluates the system's performance in the context of its applicability to part manipulation.

1 The robot vision paradigm for adaptive robot manipulation

It is worth outlining at the outset the type of system envisaged in this paper. Visual understanding of a 3-D environment is essentially an active process, requiring the acquisition of several images of the local environment prior to analysis. This is popularly known as 'active vision' but it must be distinguished from the common industrial techniques which exploit 'active sensing'. The word active in the former refers to the *investigative* nature in passive vision while the latter refers to the use of contrived illumination (e.g. light stripers, lasers) or other intentionally radiated signals (e.g. ultra-sonics) to effect the computation of 3-D information. In this paper, we will use the word 'passive' to refer to the former approach and 'active' to refer to the latter. This robot vision paradigm, then, comprises image acquisition from multiple viewpoints, computation of 3-D structure, object matching or recognition, pose estimation, formulation of a grasping strategy, and subsequent manipulation. We are concerned in this paper with all of these issues except the last two, although it is readily acknowledged that they are indeed key issues.

2 Structure from motion.

The essential idea is to analyse the apparent motion of objects arising from the changing vantage point of a moving camera in order to compute, first, the distance from the camera of key features on the object (e.g. edge contours), and second, the structure of the object by interpolation and model construction. From an intuitive point of view, camera motion is identical to the stereo process in that we are identifying points in the image (in this case, edge contours) and then tracking them as they appear to move due to the changing position and, perhaps, attitude of the camera system. At the end of the sequence of images, we then have two sets of corresponding points, connected by optic flow vectors, in the first and last images of the sequence. The depth, or distance, of the point in the world can then be computed by triangulation of the two ends of the optic flow

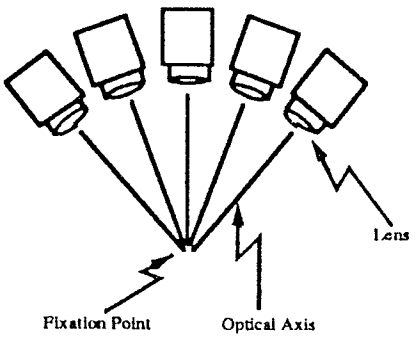

Figure 1: Rotational motion of a camera about a fixation point

vectors. In the research described in this paper, we have confined our attention to one simple type of camera motion in which the camera is rotated about a fixation point (see figure 1). The exact details of this technique can be found in [7] and [9]; we will content ourselves here with just a summary of the approach.

There are eight steps in the computation of the optical flow field. These are:

1. Acquisition of a number of images (nine in this instance).
2. Laplacian of Gaussian filtering.
3. Zero-crossing extraction.
4. Adaptive selection of zero-crossings.
5. Computation of (temporal) first differences.
6. Computation of the orthogonal component of velocity.
7. Computation of the true velocity vector.
8. Tracking of feature by isolation of adjoining (contiguous) flow vectors.

There are two components of optical flow: the rotational v_r and the translational v_t, i.e., $v = v_r + v_t$. The rotational component can be determined directly from the known camera trajectory while the direction of the translational component is also constrained by the camera motion. The main difficulty when attempting to compute the true optical flow of a point on an edge contour in the image is that we cannot say with any certainty in which direction the point has moved, based purely on local information. The only thing we can compute with certainty is the *orthogonal component of the velocity vector*, i.e. the component which is normal to the local contour orientation. This vector component is referred to as $v^{\perp}$, and the second component is referred to as the tangential component, $v^{\top}$. Thus, the true velocity vector v is also given by:

$$v = v^{\perp} + v^{\top}$$

If the luminance intensity does not change with time (i.e. there are no moving light sources in the environment) the component of the orthogonal velocity vector for each image point is given by:

$$v^{\perp} = -\frac{\partial I/\partial t}{\mid \nabla I \mid}$$

where ∂ indicates the partial derivative operator and $\mid \nabla I \mid$ is the local intensity gradient.

In the technique described here, we compute the time derivative of a $\nabla^2 G$ filtered image instead of the raw intensity image and compute the optical flow at zero-crossing contours. This means that the amount of data to be processed is limited and, futhermore, the effects of noise are less pronounced.

For the constrained camera motion shown in figure 1, the image velocity components can then be written as (see [9] for details):

$$v_t = \left(\frac{x(D_1 - D_2\cos\theta) - FD_2\sin\theta}{Z\Delta t}, \frac{y(D_1 - D_2\cos\theta)}{Z\Delta t} \right)$$

$$v_r = \left(\frac{-(x^2 + F^2)\theta}{F\Delta t}, \frac{-xy\theta}{F\Delta t} \right)$$

Where, F is the focal length of the lens, x and y are the coordinates of the point in the image plane at time T, θ is the rotational angle of the camera, D_1 an D_2 are the distances of the camera from the fixation point at time T and $T+\Delta t$, respectively, and Z is the distance of the camera to the world point corresponding to the image point (x, y).

We now have two expressions for v:

$$v = v^{\perp} + v^{\top}$$

$$v = v_r + v_t$$

The computation of the true velocity vector v is effected by combining these expressions. This can be accomplished directly by solving the attendant system of equations or indirectly by a geometrical construction (see [7]).

Computing v in this manner and, in particular, computing $v^{\perp}$ using image differences, errors can still recorded in the final flow. A significant improvement can be achieved by performing a contour-to-contour matching between successive frames, along the direction of the flow vectors, tuning the length of the flow vectors to the correct size. Although a small difference between successive frames is required to guarantee the accuracy in the computation of the orthogonal component $v^{\perp}$, a long baseline is required for the depth measurement. For this reason, many images are normally considered and the flow field obtained for a sequence of images is used for range computation: the flow vector from the first image to the last image being employed in the computation of depth.

The depth, for each contour point, is computed by applying the inverse perspective transformation, derived from camera models corresponding to the initial and final camera positions, to the two points given by the origin of the optical flow vector and the end of the optical flow vector.

To illustrate this approach to computing the depth of objects, figure 2 shows a motion sequences of comprising nine images. Each of the constituent images in these image sequences

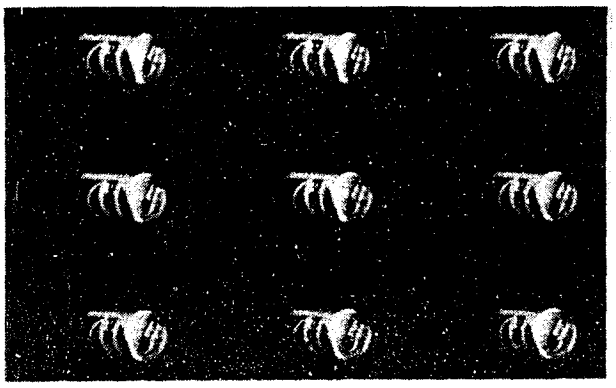

Figure 2: A sequence of intensity images

were then convolved with a Laplacian of Gaussian mask (standard deviation of the Gaussian function = 4.0 pixels) and the zero-crossings contours were extracted. An adaptive thresholding technique was employed to identify the most relevant contours (see figure 3). The associated optical flow field is also shown in figure 3, together with a range image representing the range of all visible points on the surface. This was generated by weighted linear interpolation between depth values on zero-crossing contours. Spurious values in this range image have been suppressed by applying an 11x11 pixel modal filter, whereby the value of each point in the range image is replaced with the value associated with the mode of a histogram of the values in a local region around that point.

We proceed now to the issue of construction of 3-D models from the range data and to the recognition of these models and the computation of pose for subsequent robotic manipulation.

3 Model matching and pose estimation

The 3-D representation developed and used in this system is a simple planar-surface based representation, not dissimilar to that of Roberts [5]. This representation most naturally describes polyhedral objects, but by using multiple surfaces can be employed to represent surfaces which are

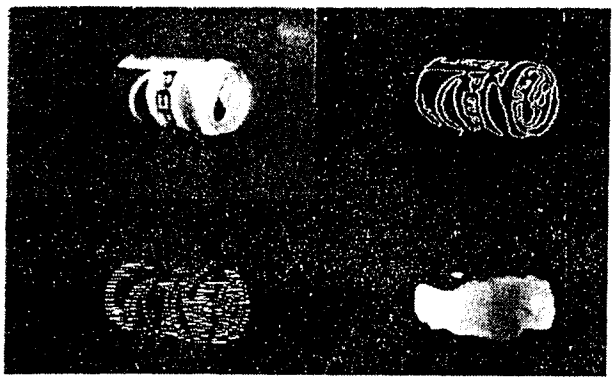

Figure 3: The third image in a sequence of nine images (top left), its associated 'significant' zero-crossings (top right), the optical flow determined between the third and eight images (bottom left), and the resultant interpolated and filtered range map (bottom right)

curved. Obviously, though, this representation of objects is not stable (i.e. different instances of the same object can have different representations), but along with the recognition technique to be detailed next, it is more than adequate for the task of representing objects to be recognised. Faugeras [2,3] and Henderson [4] describe a method of interpolating surface representations from well-sampled point data (e.g. range data). '3-point seed' planar surfaces are generated using any combination of three points which are within the sampling distance of each other. These seed surfaces are subsequently merged into larger surfaces. This type of technique is employed here, although the 3-point seed surfaces are not merged into larger surfaces. Rather, they are processed and subsequently used to generate other 3-D representations: the $2\frac{1}{2}$-D sketch (a representation of depth and local surface orientation), EGIs (Extended Gaussian Images — a 2-D histogram of the local surface normal vectors of an object), and directional histograms (1-D histograms of certain components of the local surface normal vectors of an object).

The object recognition strategy proceeds by invoking approximate views of *known* models (i.e. models in the database of objects), fine tuning these views with respect to the *viewed* object (i.e. the object view generated from the optical flow), and then by evaluating a measure of similarity between *known* models and *viewed* objects. The following summarizes the essentials of the algorithm; for details see [1].

Model view invocation is performed by determining possible orientations from which each *known* model could be viewed (in order to generate a view similar to the *viewed* object). The focal axis of the viewing device/camera with respect to the *known* model's frame of reference is first determined, and subsequently possible values for the roll of the camera with respect to its own frame of reference (i.e. around the focal axis) are calculated. In order to determine potential orientations of the camera with respect to a *known* model's frame of reference, a sample of all possible orientations is used. This sampling of orientation space is defined by the EGI of the *known* model. Using each of EGI elements (surface normals directions) as possible orientations of the camera focal axis, directional histograms of the tilts[1] visible from the *known* model are determined. These directional histograms are compared using a correlation technique with a directional histogram of tilt determined from the *viewed* object, resulting in a degree-of-fit for each possible orientation.

Having determined potential orientations for the camera focal axis, potential values for roll, around that focal axis, must be calculated. This is accomplished by correlating the directional histogram of roll defined by the *viewed* object with that derived from the *known* model in an arbitrary roll, as viewed using the previously determined focal axis.

At this stage, we have potential orientation frames of reference for *known* models. These, however, are only approximations as the orienta-

[1]The tilt is defined as the angle subtended by the principle ray of the camera and the local surface normal vector.

tion space is quite coarsely sampled. One task of model matching is, then, to fine tune these orientation frames and to determine the position of the viewing camera relative to the *known* model.

The object recognition strategy continues, then, by

1. Determining of approximate object position by computing the distance to the centroid of the viewed object and by invoking a *known* view for a camera at this distance.

2. Fine tuning of object orientation in a manner similar to that of computing the camera roll above.

3. Fine tuning of viewed object position by translating it in a direction normal to the principle ray of the camera, choosing the position (translation) which yields the highest correlation between maps of the surface normal vectors of both *viewed* and *known* models.

4. Fine tuning of object depth by translating along the principle ray of the camera, choosing the position which yields the highest correlation between range images of both *viewed* and *known* models.

5. Fine tuning of viewed object position to subpixel accuracy by fitting a quadratic function to the variation in correlation between surface normal maps and choosing the position corresponding to the maximum of this correlation curve.

The final object recognition is accomplished by choosing the *known* model (view) which yields the maximum correlation in this fine tuning; clearly, the object pose has also been computed at this stage also. Figure 4 shows examples of an object data-base while figure 5 shows the result of this object recognition strategy.

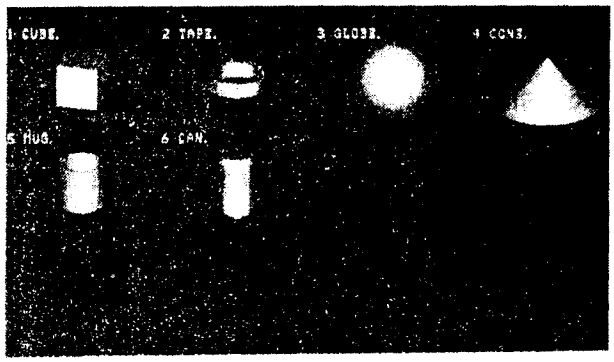

Figure 4: Database of object models

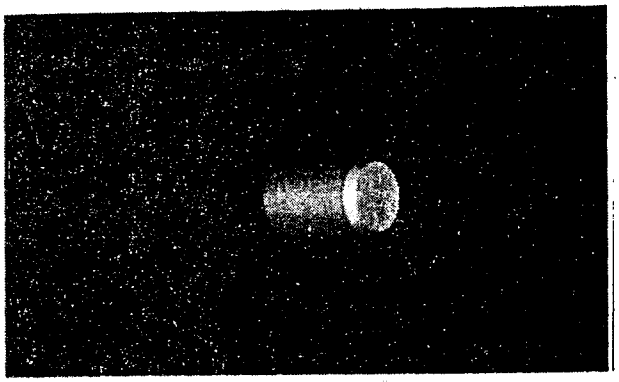

Figure 5: Pose of model which achieved maximum score in recognition strategy

4 The *VIS a VIS* environment

A brief word in about the computer vision environment used in this research is in order. Unfortunately, space constraints prohibit no more than a cursory summary. All of the results described in this paper were obtained using a parallel computer vision system, *VIS a VIS*, which effects coarse granularity parallelism on a array of transputers. The acronym *VIS* stands for Virtual Image System: the environment facilitating dynamic creation of wide variety of image types. Parallelism is achieved by Remote Procedure Calls to other instantiations of *VIS* running on other transputer nodes and robot manipulation is effected using an in-built language which controls a Scara-type robot in a Cartesian frame of reference. See [6,8] and a forthcoming book [10] for further details.

5 Summary and discussion

The technique described in this paper has been extensively tested using the passively-acquired range data generated using the structure from motion approach described above and it has achieved 100% success in object recognition for unambiguous objects. While it would be foolish to contend that this approach represents a 'final' solution to the problems associated with object recognition using passive computer vision, it has proved to be quite robust — given the noisy nature of the models which are generated using passive vision — and demonstrates the usefulness of optical flow for object recognition and pose estimation in the context of robotic part manipulation.

There are several outstanding issues in the use of passive vision for robotic part manipulation. These include segmentation, grasping and manipulation strategies, and the computational complexity of the task. These issues are the next to be addressed in this research programme. However, the computational complexity of the technique outlined places it, at present, far outside the bounds of industrial feasibility; on a small parallel system (twelve nodes), the total time required for object recognition and pose estimation is of the order of 45 minutes. It can be argued that the technique will scale well but it remains to be demonstrated that this is the case. As an industrial tool, its robustness and repeatability must also be established. Obviously there remains much to do, but the prognosis is good!

6 References

[1] Dawson K.M. and Vernon D. 1991, 'Model-Based 3-D Object Recognition Using Scalar Transform Descriptors,' Proc. SPIE, Vol. 1609 (accepted for publication).

[2] Faugeras, O. D. 1983. 'Conversion algorithms between 3D Shape Representations' in"Fundamentals in Computer Vision" - edited by O. D. Faugeras. pp. 305-314. Cambridge University Press.

[3] Faugeras, O.D. and Herbert, M. 1986. 'The representation, recognition and location of 3-D objects', International Journal of Robotics Research, Vol. 5, No. 3. pp.27-52.

[4] Henderson, T. 1983. 'Efficient 3-D Object Representations for Industrial Vision Systems', IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-5, No. 6, pp. 609-618

[5] Roberts, L.G. 1965. Machine Perception of Three-Dimensional Solids.', Chapter 9 in Optical and Electro-optical information processing (edited by J.T. Teppett *et al.* , MIT Press, Cambridge, MA. pp.159-197.

[6] Sandini, G., Tistarelli, M., and Vernon, D. 1988 'A Pyramid Based Environment for the Development of Computer Vision Applications', IEEE International Workshop on Intelligent Robots and Systems, Tokyo.

[7] Sandini, G. and Tistarelli, M. 1990. 'Active Tracking Strategy for Monocular Depth Inference from Multiple Frames', IEEE Transanctions on Pattern Analysis and Machine Intelligence, Vol. 12, No. 1, pp.13–27.

[8] Vernon, D. and Sandini, G. 1988. 'VIS: A Virtual Image System for Image Understanding', Software — Practice and Experience, Vol. 18, No. 5, pp. 395–414.

[9] Vernon, D. and Tistarelli, M. 1991. 'Using Camera Motion to Estimate Range for Robotic Parts Manipulation', IEEE Transaction on Robotics and Automation, Vol. 6, No. 5, pp. 509–521.

[10] Vernon, D. and Sandini, G. (eds.) 1992. 'Parallel Computer Vision — The *VIS a VIS* System', forthcoming publication by Ellis Horwood, London.