Visual Cues for a Fixating Active Agent

Mårten Björkman and Jan-Olof Eklundh

Computational Vision and Active Perception Laboratory (CVAP)
Department of Numerical Analysis and Computing Science
Royal Institute of Technology (KTH), S-100 44 Stockholm, Sweden
celle@nada.kth.se,joe@nada.kth.se

Abstract. In order for an active visual agent to act in a dynamic environment, it needs the ability to fixate onto objects that might be of interest. In this article we will discuss issues concerning the design of a binocular system with such capabilities. The problems range from gaze shifting and saccading, to epipolar geometry and ego-motion estimation. In the end of the paper it will be shown how scene parts of independent motion, that will be used to trigger saccades, can be efficiently detected.

1 Introduction

A person that moves around in the world, while looking at various locations and things in his way, will experience that objects will enter and leave his field of view, due to his ego-motion or the motion of the objects. He will sometimes shift his gaze to such objects, to find out e.g. if their trajectories cross his path or to determine what kind of things they are. In fact, he may be looking for specific types or instances of objects, for example they may be obstacles to avoid or something he needs.

These activities form part of what an intelligent agent, acting and existing in the world, use visual perception for. In our research we are engaged in a long-term effort to study principles and methods for developing such agents and implement them in terms of mobile robots, capable of a set of behaviours, including also grasping and manipulating things. These systems will be engaged in various foreground and background tasks, defined by their interests and drives. In this paper we will discuss work on the early visual mechanisms of such an active and purposive agent, and present how it can derive a spatial understanding of its environment that can serve higher level processes, such as recognition, and also motor behaviours.

2 Gaze shifting and saccading

Depending on the task at hand, an active visual agent moves its gaze differently from occasion to occasion. An agent that has found an object of interest, fixates onto the object and tracks it as it moves, in order to gather as much information about the object as possible. If the agent is looking for something special, it keeps moving its cameras between different locations, extracting enough information to judge whether an object is of interest has been found. It could also be the case that the agent is not at all interested in what is going on in the neighbourhood and moves its cameras only to stabilize the image data. To detect objects that might enter the scene, it is necessary to have a background process, that reacts on radical changes in the images. If the visual data is not stabilized, such changes might not be properly detected.

There can be a number of reasons why saccades are triggered in a system like ours. If the agent loses interest in an object that is being tracked, the cameras may be moved towards a more interesting part of the scene, which means that saccades are triggered by the robot itself in a top-down fashion. A triggering might also originate from low-level processes that pass information about possible interesting objects entering the field of view bottom-up, so that the visual system can react accordingly. Consequently, it is necessary for an artificial system to include low-level vision processes that are capable of detecting areas, that might be of interest, over the whole visual field.

A difficult question is what is supposed to be considered as interesting. Objects that directly affect the performance of the agent, will always be necessary to identify. There are a number of visual cues that might be important for the agent to know where to look next and since they all might be useful, they have to be considered in parallel. In this study we have been concentrating on the ability to identify areas of independent motion. The work is much in the spirit of [12], but rather than using image motion, figure-ground segmentation is performed using motion in 3D. In cases of translations, it might otherwise be hard to separate moving objects from the background.

3 Epipolar geometry estimation

In order to identify image regions as objects in 3D space, an agent would benefit from the use of binocular disparities. Disparities have been used extensively in active vision and robotics, but the applications have often been limited to behaviours such as navigation and obstacle avoidance [16, 9] or simply binocular tracking [4]. In robotics, cameras have typically been mounted in parallel, so as to simplify the procedure of calculating the actual disparities. However, if the binocular camera system is to be used in for example manipulation, parallel cameras might lead to objects only being visible in one of the two cameras. In order to maximize the applicability and flexibility of the system, it is necessary for the system to be able to verge dynamically.

The reason why verged cameras are seldom used in practise, is because it is hard to keep track of the epipolar geometry, that is the relative position and orientation of the cameras. Knowing the epipolar geometry is essential if image features are to be matched between the two cameras and distances measured in metric space. Most stereo head systems have counters on their motors, which can be used to estimate the epipolar geometry. However, all systems include delays and if the cameras are in continuous motion it is often hard to know the true relation between two newly grabbed images. The problem gets even harder when the computational load of the system is varying as the agent goes from one task to another. In our study, we have concluded that motor counters are preferably used to get a rough estimate of the epipolar geometry, but if you like to calculate a dense disparity map, the epipolar geometry better be estimated using information available in the images themselves.

3.1 The essential matrix

Typically the essential matrix $\mathbf{E}$ is used to describe the relationship between the projections of 3D points onto the left and right image planes [11]. For a system such as the one used in this study, an image point $\mathbf{x_l}$ in the left image is constrained to a line, defined by the corresponding projection in the right image $\mathbf{x_r}$, according to the equation:

$$\mathbf{x_l^T E x_r} = \begin{pmatrix} x_l \\ y_l \\ 1 \end{pmatrix}^T \begin{pmatrix} 0 & -\sin(\alpha_l) & 0 \\ -\sin(\alpha_r) & 0 & \cos(\alpha_r) \\ 0 & -\cos(\alpha_l) & 0 \end{pmatrix} \begin{pmatrix} x_r \\ y_r \\ 1 \end{pmatrix} = 0. \quad (1)$$

In the equation the projections, $\mathbf{x_l}$ and $\mathbf{x_r}$, are in homogeneous coordinates. The pan angles, α_l and α_r, are the unknown parameters to be searched by the epipolar estimation process. The stereo head is constrained such that it only involves two degrees of freedom, even if a typical binocular stereo system might have as many as six [3]. However, rotations around the optical axes do not change the visual data, only its orientation, and will not be needed here. Since the system always will be in fixation, there is no relative tilt between the cameras. A joint tilt of both cameras does not change the nature of the problem.

In our work, the angles α_l and α_r are iteratively searched in a least square framework. In order to minimize the influence of outliers, we use random sampling. Random sets of six points each are generated and for each set the angles are estimated. Unlike RANSAC [6, 15], where a winner is selected among the resulting estimates, we simply calculate the mean of all values that can be considered as feasible. It turns out that sets including outliers result in slower convergence and can easily be eliminated from the final result. Further details and evaluations can be found in [2].

3.2 An optical flow model

The essential matrix has got one major disadvantage. If the cameras are located almost parallel, the least square problem described in Section 3.1 will be illconditioned and the results can not be trusted. The problem originates from the fact the flow induced by a small rotation can not be separated from a translation along the baseline, since the depths have been eliminated in the essential matrix and the magnitude of the translational flow can not be used.

In the presented system, we use an alternative model based on optical flow [10], that is typically used in structure from motion algorithms. The disparities, that is the difference in image position between the two cameras, can be described as follows:

$$\begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} 1+x^2 \\ xy \end{pmatrix} \beta + \frac{1}{Z} \begin{pmatrix} \cos(\alpha_l) - x\sin(\alpha_l) \\ -y\sin(\alpha_l) \end{pmatrix}. \tag{2}$$

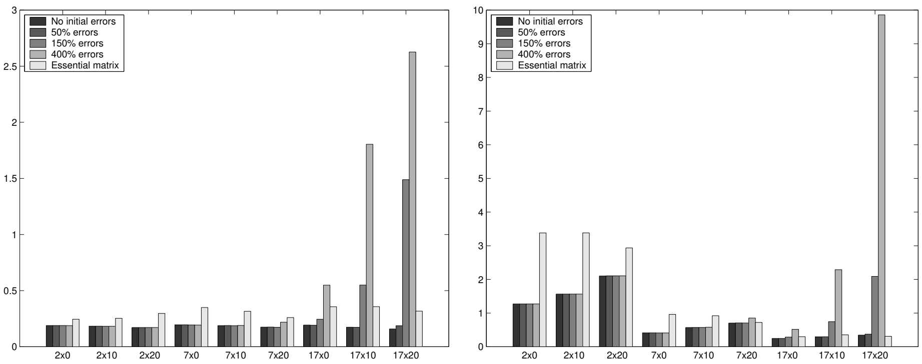
The vergence angle β is the sum of the two pan angles, α_l and α_r, but unlike the essential matrix, the depth Z has not been eliminated from the equation and has to be estimated as well. The reason why this approximate model is rarely used in stereo vision, is because it collapses if the difference in position and orientation between the cameras is too large. However, our study show that the model is in fact appropriate for a stereo head system under typical working conditions.

The unknown parameters are solved iteratively, with one pass estimating the angles and another pass determining the depths. Considering the fact that horizontal disparities never change ordering as the cameras are verging, it is possible to get a rough estimate of the vergence angle, that can be used to initialize the procedure. We match a guessed median depth of points in 3D space to the median disparity of extracted feature points and calculate the corresponding vergence angle. Simulations show that the procedure will converge, even if the initial depth is relatively far away from the truth. If the guess is within a factor of two from the true value, the procedure rarely diverges, but if the error is as large as 400%, the iterations often diverge for large vergence angles.

3.3 Simulations

Both methods were tested through a number of simulations based on randomly generated 3D points, spread around an area 10 to 30 baselines in front of the cameras. Gaussian noise, with a standard deviation of one pixel, was added to the points that were projected onto two 360×288 pixel image planes. Each simulation included about 500 feature correspondences, out of which 20% were outliers. These outliers, that were modeled with an additional noise source of 30 pixels, represent features that have been wrongly matched, in the sense that the two projections do not originate from the same point in 3D space.

The results can be divided into two components, a rotational component and a translational one, that is the vergence angle and the position of right camera in the coordinate frame of the left. Figure 1 shows the translational and rotational errors for a number of cases. The rotational error is about $0.2°$ for the optical flow method and slightly larger for the essential matrix. For near parallel systems, the translation is considerably harder to estimate and for vergence angles of $2°$, this error can be as much as ten times larger. Asymmetric systems are slightly harder, then symmetric ones. The major difference between the methods is in the convergence. As the essential matrix method rarely converges for parallel systems, the second method have problems with vergence angles larger than $15°$, especially if the initial median depth is far from the truth. The optical flow method is about twice as fast as the first method and requires about 27 ms on a 195 MHz MIPS R10K processor.



(a) (b)

Fig. 1. The standard deviation in degrees of the rotational (a) and translational (b) errors for various combinations of true rotations and translations. The results of the optical flow based method are shown in the first four bars of each group, for different errors in the initial median depth. The last bar in each group represents the results based on the essential matrix.

4 Ego-motion estimation

Knowing the motion of the agent itself, the ego-motion, is essential in order to stabilize the image data and find image regions of independent motion. Just like the case of epipolar geometry estimation using motor feedback, odometry can be used to get an idea of the ego-motion, but the problem is somewhat harder than that. If any conclusions are to be drawn from the images based on ego-motion, the estimated ego-motion has to be relative to the cameras, not to the agent itself. As the agent is moving, the neck is rotating and the cameras are constantly changing orientation, its terribly hard to know the exact direction and position of the cameras at every instance of time, especially since all systems involve delays that may be hard to predict. It becomes a necessity for the system to estimate the ego-motion based on extracted image data, even if an initial prediction based on odometry might still be of good use.

In a monocular system estimating ego-motion is a very difficult problem, if the cameras are in translation. There are a number of alternative methods that can be used [8, 14], but so far none of these reach the requirements on robustness and speed needed for a system like ours. Fortunately, the problem is greatly simplified in the binocular case. In our system we use reconstructed 3D features, that are easily found once the epipolar geometry is known, and try to minimize the equation

$$\sum_{i=1}^{N} \|\mathbf{y_i} - (\mathbf{R}\mathbf{x_i} + \mathbf{t})\|^2, \tag{3}$$

where $\mathbf{R}$ is a rotation matrix, $\mathbf{t}$ a translation and, $\mathbf{x_i}$ and $\mathbf{y_i}$ are the reconstructed feature points of two consecutive time frames. Since it can be expected that the clouds of 3D points have the same shape in both instances of time, the translation can be found as the difference in position between the centers of the two clouds, that is $\mathbf{t} = \bar{\mathbf{y}} - \bar{\mathbf{x}}$. Subtracting the centers from the 3D points, we get two new sets of points $\acute{\mathbf{x}}_\mathbf{i} = \mathbf{x_i} - \bar{\mathbf{x}}$ and $\acute{\mathbf{y}}_\mathbf{i} = \mathbf{y_i} - \bar{\mathbf{y}}$.

We use an approach based on singular value decomposition (SVD), as proposed by Arun et al [1]. It can be shown that minimizing Equation 3 is equivalent to maximizing

$$f_2(\mathbf{R}) = \sum_{i=1}^{N} \acute{\mathbf{y}}_\mathbf{i}^\mathbf{T} \mathbf{R} \acute{\mathbf{x}}_\mathbf{i} = \mathrm{Trace}(\mathbf{RH}), \tag{4}$$

where $\mathbf{H} = \sum_{i=1}^{N} \acute{\mathbf{x}}_\mathbf{i} \acute{\mathbf{y}}_\mathbf{i}^\mathbf{T}$. The rotational component $\hat{\mathbf{R}}$ that maximizes $f_2(\mathbf{R})$ can be found by first performing a SVD of $\mathbf{H}$. The decomposition $\mathbf{H} = \mathbf{UDV^T}$ consists of two orthogonal matrices $\mathbf{U}$ and $\mathbf{V}$, and a diagonal matrix $\mathbf{D}$ of non-negative elements. The minimizer of Equation 4 is then given by $\hat{\mathbf{R}} = \mathbf{VU^T}$, which will be used as our estimated rotation. In practise we perform these operations twice and use weights, to minimize the influence of reconstructed points with large errors.

4.1 Simulations

Simulations show that the rotational errors are rarely more than about $0.1°$, which only happens in cases of large rotations and small translations parallel to the image plane. In such cases the translational error might be as large as $6°$ in direction and 0.03 baselines in speed, whereas typical errors are $0.02°$ in rotation, $2°$ in translational direction and 0.01 baselines in speed. The feature points involved in the simulations were generated such as described in Section 3.3. Since features have to be visible in both cameras at two different instances of time, only 100 reconstructed points were used, with 20% being outliers representing areas of independent motion. A systematic error of $0.3°$ was added to the vergence angle before reconstruction, modeling the inaccuracy of the epipolar geometry estimation. The computational cost of the method is as low as 3 ms, excluding the matching of corner features. Since it relies on reconstructed 3D points, the features have to be matched in stereo, as well as in motion, such as shown in Figure 2. The feature extraction itself costs as much as about 15 ms per image, using Harris [7] corner detector. Further details about the matching processes and how outliers are identified, can be found in [2].

5 Independent motion

The benefit of knowing the ego-motion of the cameras is the fact that images of future time frames can be predicted. Changes in the scene can then easily be detected through simple image subtraction, even if the agent itself is in continuous motion. In the system presented here, disparities are calculated using a method based on correlations and dynamic programming [13, 5]. These calculations are done after images have been rectified, using the results from the

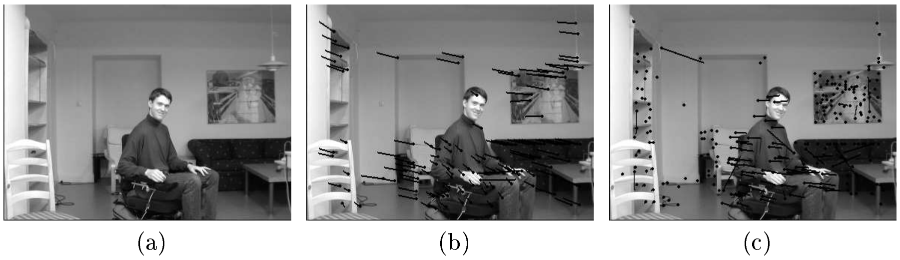


Fig. 2. The left image (a), the right image including stereo feature correspondences (b) and features matched in motion (c).

epipolar estimation. Using the disparities, such as the example in Figure 3(a), and the estimated ego-motion, a warping is then performed from the current image to the next one. This predicted image is then subtracted from the true next image. Results after thresholding can be seen in Figure 3(b). Points above the threshold are finally put into a three-dimensional histogram, the largest peak of the histogram is found and a segmentation, such as the one shown in Figure 3(c), can be done.

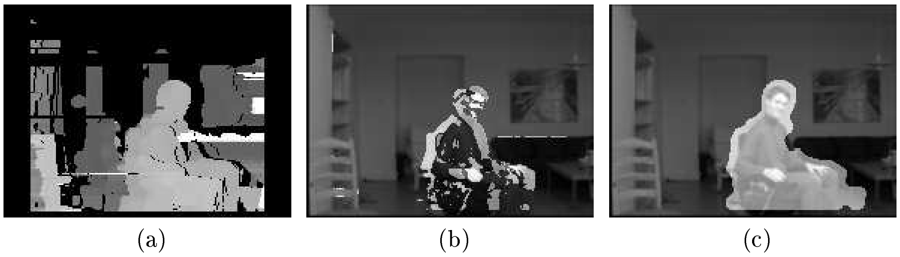


Fig. 3. The disparity map (a), the result after warping and subtraction (b) and image regions of independent motion (c).

The methods presented in previous sections were implemented on a Nomad 200 platform, powered by a 450 MHz Pentium III. The complete system runs in about 6 Hz, including feature extraction, matching, epipolar and ego-motion estimation, as well as the calculation of dense disparity maps and segmentation.

6 Discussion

In the presented paper we have been dealing with the problem of designing a system capable of dynamic fixation. The technical part of the paper included a presentation of two different methods for estimating the epipolar geometry of a binocular system. It was shown that this is in fact feasible in terms of speed, as well as robustness. Once feature points have been reconstructed in 3D space, the ego-motion was estimated at a very low computational cost. In the end of the article we show how knowledge about depth and ego-motion can be used to find image region of independent motion. An issue that has not been covered is how to exploit the available 3D information during tracking and gradually improve the speed as new information becomes available. How information about objects in 3D space is to be stored and updated is another important and difficult question.

References

1. K.S. Arun, T.S. Huang and S.D. Blostein, Least-Squares Fitting of Two 3-D Point Sets, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9(5), pp. 698-700, Sep 1987.
2. M. Björkman and J-O. Eklundh, Real-Time Epipolar Geometry Estimation and Disparity, Tech. Report ISRN KTH/NA/P-00/09-SE, NADA, Royal Institute of Technology, Mar 2000.
3. M.J. Brooks, L. de Agapito, D.Q. Huynh and L. Baumela, Direct Methods for Self-Calibration of a Moving Stereo Head, In *Proc. of the 4th European Conf. on Computer Vision*, pp. 415-426, 1996.
4. D. Coombs and C.M. Brown, Real-Time Binocular Smooth-Pursuit, *Int. Journal of Computer Vision*, 11(2), pp. 147-165, Oct 1993.
5. I.J. Cox, S.L. Hingorani, S.B. Rao and B.M. Maggs, A Maximum Likelihood Stereo Algorithm, *Computer Vision and Image Understanding*, 63(3), pp. 542-567, 1996.
6. M.A. Fischler and R.C. Bolles, Random Sampling Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography, *DARPA Image Understanding Workshop*, pp. 71-88, 1980.
7. C. Harris and M. Stephens, A Combined Corner and Edge Detector, In *Proc. of the 4th Alvey Conf.*, pp. 189-192, 1988.
8. D.J. Heeger and A.D. Jepson, Subspace Methods for Recovering Rigid Motion I: Algorithm and Implementation, *Int. Journal of Computer Vision*, 7(2), pp. 95-117, 1992.
9. K. Konolige, Small Vision Systems: Hardware and Implementation, *8th Int. Symposium on Robotics Research*, Hayama, Japan, 1997.
10. H.C. Longuet-Higgins and K. Prazdny, The Interpretation of a Moving Retinal Image, *Proc. of Royal Society of London*, B-208, pp. 385-397, 1980.
11. H. Longuet-Higgins, A Computer Algorithm for Reconstructing a Scene From Two Projections, *Nature*, 293, pp. 133-135, 1981.
12. D.W. Murray, K.J. Bradshaw, P.F. McLauchlan, I.D. Reid and P.M. Sharkey, Driving saccade to pursuit using image motion, *Int. Journal of Computer Vision*, 16(3), pp 205-228, 1995.
13. Y. Ohta and T. Kanade, Stereo by Intra- and Inter-Scanline Search Using Dynamic Programming, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 7(2), pp. 139-154, 1985.
14. J. Oliensis, Computing the Camera Heading from Multiple Frames In *Proc. of IEEE Computer Vision and Pattern Recognition*, pp. 203-210, 1998.
15. P.H.S. Torr and D.W. Murray, The Development and Comparison of Robust Methods for Estimating the Fundamental Matrix, *Int. Journal of Computer Vision*, 24(3), pp. 271-300, 1997.
16. V. Tucakov, M. Sahota, D. Murray, A. Mackworth, J. Little, S. Kingdon, C. Jennings and R. Barman, Spinoza: A Stereoscopic Visually Guided Mobile Robot, In *Proc. the 13th Annual Hawaii Int. Conf. on System Sciences*, pp. 188-197, Jan 1997.