Mobile Camera Parameter Recovery in an Unknown Environment without Point Features

A. Eppendahl
Institute of Cybernetics
Akadeemia 21, Tallinn, Estonia

R. Maigre
Tallinn University of Technology
Ehitajate 5, Tallinn, Estonia

***Abstract*— We show that unknown camera and vehicle parameters may be recovered from images of an unknown environment with no point features. Our method complements established geometric methods which require point features. Given an image sequence, we construct a series of light field models over a range of parameter values. The 'sharpest' model is seen to correspond to the correct value of the parameter, and our method may be viewed as an abstract form of focusing. We describe experiments with a one-dimensional camera mounted on a robot moving about in a two-dimensional environment that does not contain point features. The results demonstrate that accurate parameter values can be recovered from real images using methods that do not require point features.**

***Index Terms*— Mobile robot, robot vision, self-calibration, light field model, space carving.**

I. Introduction

It is generally assumed that precise vehicle and camera calibration is required to construct an occupancy model from an image sequence captured by a mobile camera. In [1] it is observed that accurate camera parameters are required to obtain a good model. We show that this can be turned around: a good model implies accurate parameters. By measuring the quality of models, we can calibrate our vehicle and camera by finding the parameters that produce the best model. With this method of self-calibration, there is no need for feature extraction or even for the images to contain features.

In robotics, where even a low resolution 2D model can be useful, occupancy models have been common for some time [2]. Because high-resolution 3D and 5D models require much more memory, occupancy models for scene or object reconstruction [3] and light field models for rapid view synthesis [4] have only become feasible more recently. These memory intensive occupancy techniques are in some sense orthogonal to the geometric techniques commonly used in vision. Occupancy techniques assume that all camera parameters and positions are known and assume nothing about the structure of the scene [3]. Geometric techniques assume that the scene contains features with known locations in the images and assume nothing about camera parameters and positions [5]. Here we apply occupancy methods to the problem of calibration, which is often regarded as requiring geometric methods based on image features. To ensure that we are not somehow using feature information implicitly, as is done in [6] for example, we conduct our experiments in a featureless environment.

The particular model used in the experiments we report here is a form of light field [4]. A light field model is an occupancy model of light. In a 2D environment, each location cell is spilt into a 'pie' of light cells, with one light cell for each direction light may pass through that location. A very low resolution light field model is depicted in Figure 1. We extend the standard model to record contradictions by allowing each light cell to hold more than one colour of light. This allows us so to measure the quantity of contractions produced by each parameter value. The crucial observation is that this measure has a minimum at the correct value. This may be compared with the process of focusing a camera: an unfocused image is the result of an incorrect parameter, the distance to the lens, and is the average of contradictory light intensities from different areas of the lens. With this example in mind, we think of our method as 'focusing the occupancy model'.

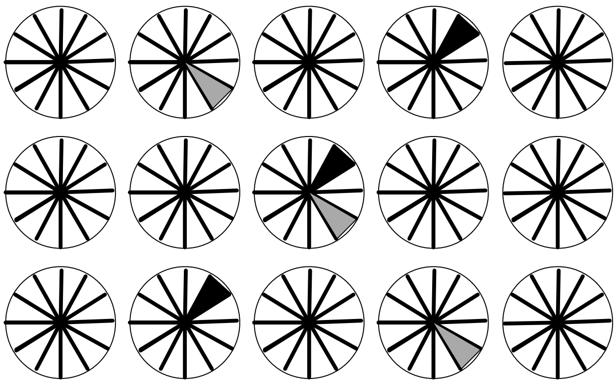

Fig. 1. A light field model of 2D space with 15 locations and 12 light cells at each location. The model shows a ray of black light passing through from the lower left to the upper right and a ray of grey light passing from upper left to lower right. The rest of the model is saturated with white light.

II. Point Features

The method we present here does not rely on the point features required for geometric methods. The basic assumption behind the geometric methods described, for example, in [5] is that points matched between images are projections of a single physical point. Let us define a *point feature* to be a physical feature with projections to image features along two distinct lines (and hence in separate images). Point features may be formed by colour variations on a surface, *painted* point features, or by the shape of an object seen against a background, *sculpted* point features.

The first two diagrams in Figure 2 show a painted point feature and a sculpted point feature: in both diagrams, the image features A and B are projections of the same physical feature C along distinct lines. In the third diagram, the object has no colour variation and hence no painted point features. Furthermore, because the object is round, X and Y are necessarily the projections of two different physical features, and so the object has no sculpted point features.

While geometric methods may be adapted to cope with an absence of point features by extracting tangents to contours in 2D images of a 3D environment [7], our method applies even in the case of 1D images of a 2D environment in which there are no contours and hence no tangents. In our experiments, we used round, uniformly coloured objects in a 2D environment. This is the worst possible situation for geometric methods.

We are not suggesting that robots typically operate in 2D environments without point features. When the environment does contains point features, locating them in images them can be difficult, expensive, inaccurate or unreliable, and so we would like to show that they are not so important as geometric methods suggest. We would also like to obtain all the information we can from simple, memory intensive, occupancy methods. Memory is only getting cheaper and, moreover, the high resolution models required for view synthesis are unnecessary for many applications in robotics, the present application to calibration being an example.

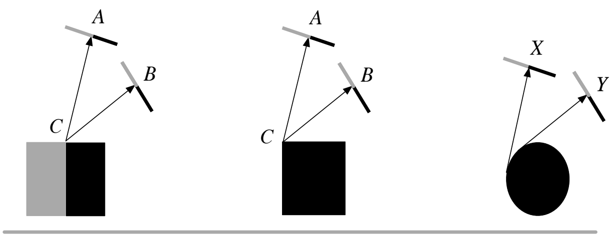

Fig. 2. A painted point feature, a sculpted point feature and an object with no point features.

III. Experiments

Experiments were conducted with a Khepera robot fitted with a K213 linear vision turret [8]. This is a small, cylindrical robot with differential drive and a fixed 1D camera that images a horizontal line of 64 grey-scale pixels over a 36 degree field of view. The auto-iris of the camera was disabled to obtain a fixed exposure and frame-rate.

The environment was a rectangular enclosure with white, banked walls lit from above with white florescent lamps. Two black cylinders were placed in the enclosure and images were thresholded so that the environment appeared to contain perfectly black cylinders silhouetted against

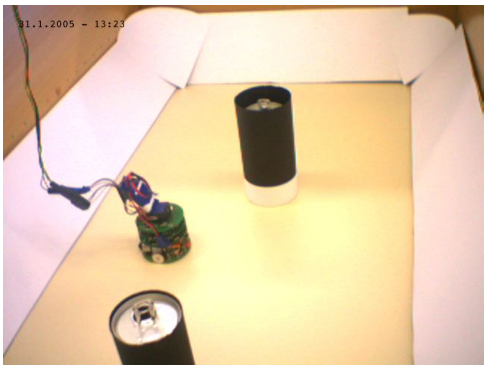

Fig. 3. The robot sits pensively in its arena. The cylinders are soft-drink cans wrapped in black paper. The camera on top of the robot images a horizontal line of 64 grey-scale pixels spanning 36 degrees.

perfectly white walls. The set-up we used is shown in Figure 3.

In the experiments, the robot was driven about by remote control without touching the cylinders. A path was chosen to ensure a number of different views of the cylinders. At regular intervals, the robot relayed lines of raw image and wheel counter data to a host computer, where the data was stored and processed off-line.

The data from a given run is processed to produce a series of models with varying camera and drive parameters. The camera parameters are focal length, in radians per pixel, together with fore/aft camera offset and left/right camera offset, in raw wheel counter units. The drive parameter is the distance between the wheels, in raw wheel counter units.

The model consists of a 2D array representing 100 by 100 physical locations in the robot's environment, spaced 150 wheel counter units apart. Each element of the location array holds a circular 1D array representing 32 directions through that physical location. Each element of each direction array has two counters, one for white light and one for black light.

For each parameter setting, the light counters of the model are set to zero and incremented as follows. For each line of image and wheel data, the wheel data is used to dead-reckon a new position for the robot and, for each pixel, either the white or the black light counters (depending on the pixel colour) along the ray corresponding to the pixel are incremented. This is actually done by first computing the camera projection for each image and then running through each location and projecting onto the images to decide which counters to increment (much as in [3]). Also, to avoid problems with motion-blurring, we only use images associated with zero wheel movement.

The drive parameter is used in the new position calculation and the camera parameters are used, together with the new position, in the ray calculation. When the parameters are correct and before dead-reckoning begins to introduce errors, this produces a partial model of the real 2D light field in the plane of the robot's 1D vision system, except the model contains light beyond the source of light rays in the real scene. Much of this non-existent light could be removed from the model using the space carving method described in [3], but this requires additional computation and we were curious to see if a naive method would be adequate for calibration.

IV. Results

We show the results for a simple run in which the robot first turns to look at each cylinder, then passes between the two cylinders, and finally turns around to look at both cylinders again from the opposite side.

Figure 4 shows maps derived from the light field models produced with five different values of the focal length parameter (with the other parameters at their correct values). These maps are produced by inspecting the light passing through each location in the light field model. We mark a location as *occupied* (solid square) if no direction has both black and white counts, at least two directions have black counts, and no direction has white counts. We mark a location to be *blurred* (dotted square) if at least one direction has both black and white counts, at least one direction has only black counts, and no direction has only white counts.

With these definitions and this data, blurring generally occurs around the edges of occupied areas. In the five maps shown, there are well defined patches in the areas occupied by the two black cylinders, one large and one small. The occupied locations are displayed to help us understand the blurred locations, but only the blurred locations are counted when measuring model blur.

The middle map was produced with the correct parameter values and, as one would hope, there is very little blurring. The other maps show the effect of decreasing or increasing the focal length value. As error in the the focal length increases, more and more locations have both black and white light in the same direction, and hence more blur. This is the result of rays of one colour being plotted over rays of another colour as the robot rotates and, due to the incorrect focal length, the image shift does not match the reckoned change in heading.

Figure 5 shows how the number of blurred locations varies as various parameters pass their correct value. Figures 5.a and 5.b show the number of blurred locations as the focal length and fore/aft parameters vary. In both cases there is a well-defined dip bottoming out at the correct value, and so these two parameters can be recovered accurately and with confidence.

Figure 5.c, shows how blurring varies with different values for the left/right offset and is very noisy. With filtering, we might find a weak minimum near the correct value, but it is clear that the left/right offset is much harder to recover than the fore/aft. Notice that the two minima occur slightly above the supposed correct value of 0. As it happens, the camera on the K213 vision module is slightly off centre, about 70 wheel count units in the positive direction. So, even with all the noise, the locations of the minima suggest correctly that the value of the left/right is nearer 100 than 0.

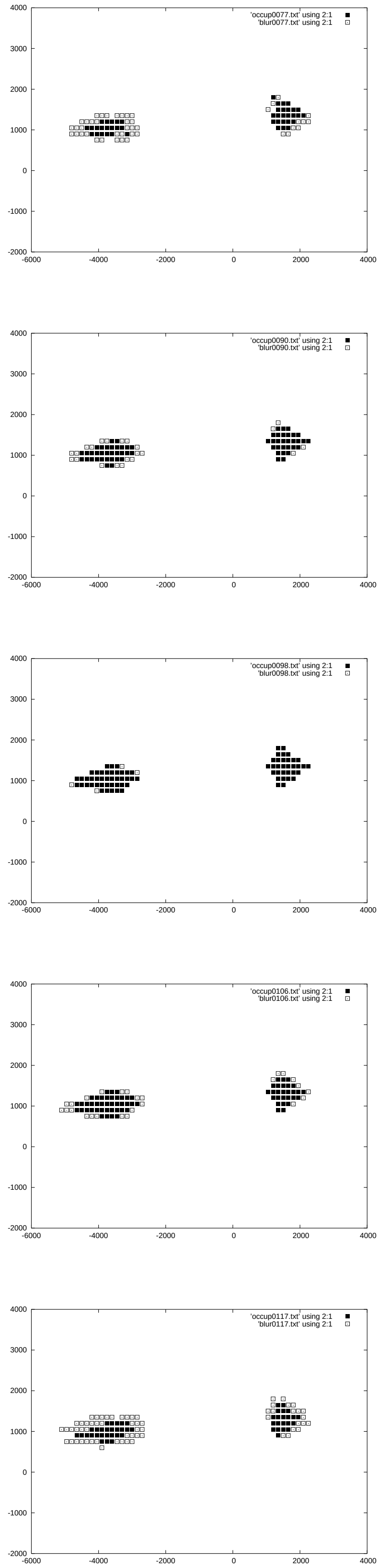

Fig. 4. Occupation and blur maps for focal length values (a) .0077, (b) .0090, (c) .0098, the correct value, (d) .0106 and (e) .0017, in radians per pixel. Occupied locations are indicated by solid squares and blurred locations, by dotted squares. Distances along the two axes, as reckoned from the initial location of the robot, are shown in raw wheel counter units.

Figure 5.d, shows the effect of the wheelbase value on model blur. there is a sharp dip at the correct value and second dip a little below. The global minimum gives the correct parameter value here, but a double dip pattern also appears in data from other runs and sometimes the correct value is between the two dips. With a little filtering, it should be possible to recover a good estimate of the wheelbase parameter.

We were surprised to see such a large qualitative difference between the left/right and fore/aft offsets. At first glance, one might think that camera offsets would generally have the same effect in any direction, simply shifting the model in that direction, but of course the offsets are with respect to the vehicle which is constantly changing position with respect to the model. Still, errors in the fore/aft offset make objects appear at the wrong size, while errors in the left/right offset make objects appear at the wrong position, and one might think the resulting blur would vary in roughly the same way.

To explain the difference, we observe that a large part of the variation in the left/right function appears to be an oscillation with period about 75. We believe this is a beat caused by aliasing against the discrete model (although we have not tested this hypothesis). As the camera on the K216 has 36 degree field of view, the fore/aft offset displaces the line of each pixel ray much less than the left/right offset, and may thus be less sensitive to aliasing effects. We would then expect less of a difference using a wide-angle camera.

The wheelbase graph also has lot of local minima, but this is less surprising given the odd, non-linear effect of the wheelbase parameter on the model construction. When the parameter is decreased, the path of the robot becomes curly, in the extreme case curling up around the initial location. When the parameter is increased, the path straightens out, in the extreme case straightening into a line along the initial heading.

We have shown the results for a short run. As the robot continues to move about, dead-reckoning errors begin to introduce additional blurring. Eventually, the parameter signal is lost. To see how far the robot can travel before dead-reckoning errors begin to obscure the parameter information, the robot was driven repeatedly between the two cylinders, pausing occasionally to turn and look at each cylinder. Figure 6 shows how the graph of blurring with respect to focal length evolves as more and more views are included in the model. The collection of graphs is indexed by total wheel movement.

At first there are not enough views to produce any blurring at all. Then, starting with the second time the robot stops to look at the cylinders, at about total distance 10000, a graph similar to Figure 5.a appears. From there to a total distance of about 30000, the graph has a clear minimum at the correct parameter value. Finally, by the fifth or sixth time the robot stops to look around, dead-reckoning errors begin to flatten the graph. This means that only the recent past, going back about 25000 wheel count units in the case of the Khepera, should be used to calibrate the robot.

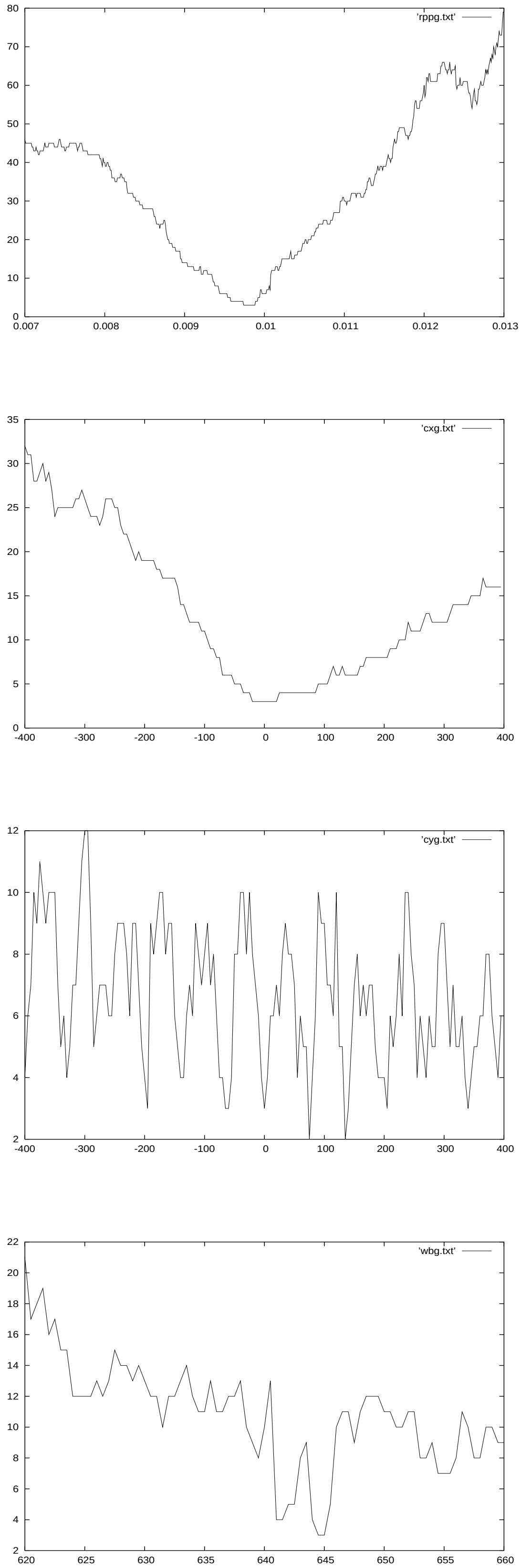

Fig. 5. Number of blurred locations plotted against (a) focal length, correct at .0098, in radians per pixel and against (b) fore/aft offset, correct at 0, (c) left/right offset, correct at 0 (but see the discussion) and (d) wheelbase, correct at 645, in raw wheel counter units.

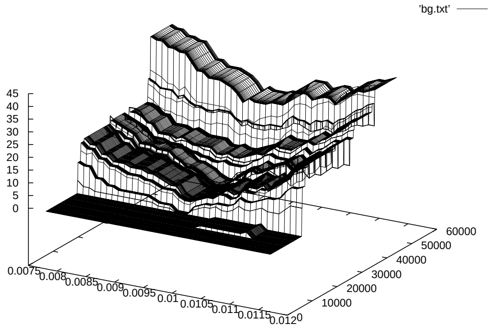

Fig. 6. Number of blurred locations plotted against focal length, in radians per pixel, and total wheel distance, in raw wheel counter units.

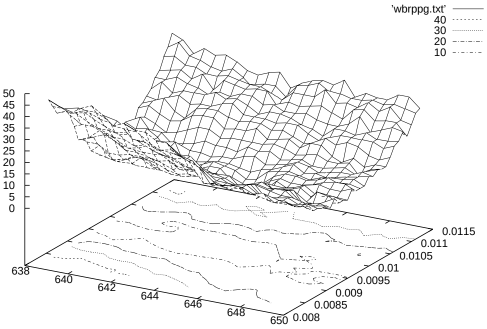

Fig. 7. Number of blurred locations plotted against wheelbase and focal length.

V. Future Work

In our experiments we have used a very naive model construction that ignores occlusions. As we were only interested in parameter recovery and not in view synthesis, we wanted to know if the very simplest construction would be adequate. Now that this appears to be the case, we might consider the more sophisticated method of ‘space carving’ [3] which handles occlusions correctly. In particular, we conjecture that it is possible to ’focus’ the models produced by the probabilistic version of space carving [9] by minimising model uncertainty. Again, because we are not interested in view synthesis, these models could be sparser and hence computed more rapidly than the photo-realistic models produced in [3] and [9].

Given an unlabelled image sequence, geometric methods can produce simultaneous estimates of all camera parameters and all camera positions [5] [10]. We are currently investigating the use of our focusing method for multiple parameter recovery without point features. Figure 7 shows how the blur count varies around the correct pair of wheelbase and focal length values. From the trough-like shape of the surface, it appears that it is best to calibrate the focal length before calibrating the wheelbase. It may also be possible to use our method to recover estimates of camera positions, and hence to localise, without point features. We might use the recent past for calibration, and the distant past for localisation (which would amount to view matching).

VI. Conclusion

We have shown that a sequence of image and wheel data can be used to determine accurately an unknown focal length or fore/aft offset for a mobile camera and to estimate a good wheelbase value. Not only have we shown that an occupancy method is capable of providing information ordinarily obtained using geometric methods, by conducting our experiments in a 2-D environment with no point features, we have shown that an occupancy method can be used where geometric methods cannot.

Acknowledgements

Preliminary versions of the results presented here were obtained by Hakim Miah as part of an undergraduate project [11]. The authors would also like to thank Lourdes Agapito for stimulating discussions and Maarja Kruusmaa for commenting on a draft of this note. Conference participation is supported by EU project IST-2001-37592 (eVikings II).

References

[1] S. M. Seitz and C. R. Dyer. Photorealistic scene reconstruction by voxel coloring. International Journal of Computer Vision, vol. 35 (2), pp. 151-173, 1999

[2] H. P. Moravec and A. Elfes. High Resolution Maps from Wide Angle Sonar. In Proceedings of the IEEE Conference on Robotics and Automation, Washington D.C., pp. 116-121, 1985.

[3] K.N. Kutulakos and S.M. Seitz. A theory of shape by space carving. In Proceedings of the 7th International Conference on Computer Vision, 1999.

[4] M. Levoy and P. Hanrahan. Light-field rendering. Computer Graphics, Annual Conference Series, 1996.

[5] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision,* second edition. Cambridge University Press, 2000.

[6] Luc Robert. Camera calibration without feature extraction. Computer Vision, Graphics, and Image Processing, vol. 63 (2), pp. 314-325, March 1995. Also INRIA Technical Report 2204.

[7] K. Y. K. Wong and R. Cipolla. Structure and motion from silhouettes. In Proceedings of ICCV01, June 2001.

[8] R. M. Harlan, D. B. Levine and S. McClarigan. The Khepera Robot and the kRobot Class: A Platform for Introducing Robotics in the Undergraduate Curriculum. In Proceedings of the 32nd SIGCSE Technical Symposium on Computer Science Education, pp. 105109, 2001.

[9] R. Bhotika, D. J. Fleet and K. N. Kutulakos. A Probabilistic Theory of Occupancy and Emptiness. In Proceedings of the European Conference on Computer Vision (ECCV), pp. 112-132, 2002.

[10] A. J. Davison. Real-time simultaneous localisation and mapping with a single camera. In Proceedings of the 9th International Conference on Computer Vision, Nice, 2003.

[11] H. Miah. Remote Exploration with Parameter Recovery Using an Occupancy Model. BSc project report, Department of Computer Science, Queen Mary, University of London, 2002.