A Stochastic Framework For Object Localisation

Y. Shao, J. E. W. Mayhew
AIVRU, The Univeristy of Sheffield, Sheffield S10 2TP, UK
email: {yuan,mayhew}@aivru.shef.ac.uk

Abstract

We describe a Bayesian architecture to estimate the position and pose of a 3D object. The system starts with knowledge of the 3D structure of the object and the prior probability distribution of its position and orientation in the workspace. This information is used to guide the search for focus features in the image, and the information recovered from the image processing is used to refine the estimates of the x-y position and pose of the object. The results of intermediate stages of processing is propagated using a Bayesian methodology. After iteration around the network, the peaks of the final probability distributions are used to estimate the position and pose, and the widths of the distributions provide a measure of confidence.

The results of the study suggest that grey-level image processing algorithms and a simple 3D model, embedded in a Bayesian statistical reasoning architecture can provide a highly effective, albeit specialised object localisation system.

keywords: model-driven vision, Bayes net, deformable template, object localisation

1 Introduction

This paper is to establish a stochastic framework for object localisation, as a part of an ongoing research project "model-driven stereo vision under camera geometry" [12].

The starting point for the research reported here is the work of The Rochester group [10] who exploited a Bayesian reasoning paradigm to build an architecture for the control and deployment of a suite of vision processing algorithms. They chose as their original task domain a table setting reasoning problem. The system answered questions concerning whether the number of places, whether the meal was breakfast or dinner etc. Later work [1] extended the system to reason about the scenes containing moving objects, model trajectories and plan appropriate monitoring strategies. The system embodied knowledge of the task domain, the vision algorithms, and their situation dependent appropriateness and costs of deployment. A foveating strategy controlled by the emerging interpretation of the scene determined the size and position of the image region processed and the algorithms utilised therein.

BMVC 1996 doi:10.5244/C.10.49

Our research is to exploit and later, extend the general principles demonstrated by Brown and colleagues in a different task domain. The application domain we have chosen is the 3D pose identification and object verification of an industrial part under conditions of variable camera geometry using a foveating 4 dof stereo camera head. Previous work used model driven expectations to detect obstacles on the ground plane under variable camera geometry [11].

A long term aim of the project is to develop stereo algorithms in which geometry obtained from a previous fixations [5], head positions and model driven expectations is taken into a different viewing sitations and used as contraints in the solution of the stereo fusion correspondence problem. As a step towards this aim we have begun to develop a Baseyian based task control architecture to deploy low level vision algorithms containing as much as possible (or needed) information specific to the 'to be recognised object', the workspace, and its illumination.

2 The Task

The task is to find the x and y position and rotational pose of a Toyota shaft assembly, as shown in Figure 1. Three focus features corresponding to bosses of the object labeled by crosses in the figure are chosen. We have available a precise 3D geometrical model of the object. The stereo camera rig has been calibrated, and we have an approximate estimate of the object's position and pose which we can vary as part of the experimental manipulation. The shaft assembly is constrained to have only three degrees of freedom. Its position can vary on a horizontal plane, and the position of this plane is known in the camera coodinates. The object can be rotated around the axis normal to this plane, The task is to estimate the translation along the x and y axes and the rotation around the z axis.

3 Image Grey Level Blob Detection

We refer to the process of localising the boss focus features using the Förstner operator [4] as a blob detection. The operator is designed to identify potential circular symmetrical features withinin an region of interest. The operator treats pixels within the region individually. It uses the "slope element", i.e., the straight line going through the pixel p_i and the direction of the gradient $\bigtriangledown g_i$. The idea is that the circular symmetrical center b, if exists, minimizes the weighted sum of the distance n_i from the slope emements. A Förstner blob candidate b is given by

$$(\sum_i W_i) \cdot b = (\sum_i W_i \cdot p_i), \tag{1}$$

where the weight matrix

$$W_i = \| \bigtriangledown g_i \|^2 \cdot \begin{pmatrix} \sin^2 \theta_i & -\cos\theta_i \sin\theta_i \\ -\cos\theta_i \sin\theta_i & \cos^2 \theta_i \end{pmatrix}. \tag{2}$$

Obviously, the blob centre estimate b is the weighted centre of gravity of all points p_i. This operator can also be interpreted as a straight-line fit in Hough space. Dots in Figure 1 shows detected blobs for the observed object.

This operator also provides the confidence contours (usually an ellipse) of the position of the blob. We assume that the Förstner blob operator is an unbiased estimator, and we take $r^2(v)$ as the variance. Then, assuming a Gaussian distribution, we can build the probability distribution $p(v|b)$ of the blob centre coordinates v as follows:

$$p(v|b) = \frac{1}{\sqrt{2\pi}r(v)} \exp(-\frac{dist^2(b,v)}{2r^2(v)})/N, \quad (3)$$

where *dist* is the distance function and N is the normalisation factor.

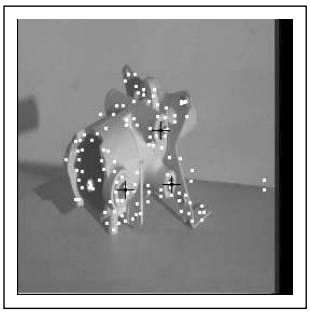

Figure 1: detected blobs

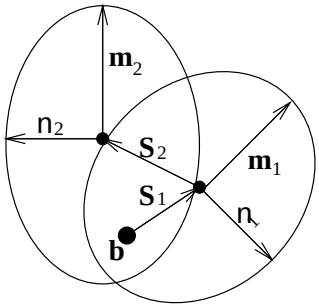

Figure 2: Deformable template for a tracked feature

4 Deformable Template

The performance of focus feature localisation using the blob operator will deteriorate quickly if the uncertainty of the initial estimation becomes large. For an instance, given as the initial estimate of the object pose ($N(50mm, 50mm)$, $N(-50mm, 50mm)$, $N(-30°, 30°))^T$ when the correct pose is $(0mm, 0mm, 0°)$, the localisation proved unsuccessful. This is due to the increase in number of the plausible but incorrect 'feature' matches.

To address this problem we combine the edge information with the blob detection using a deformable template approach [16]. This can be regarded as convolving an image with a flexible mask corresponding to the feature to be localised. The template consists of a blob and two conics, corresponding to the edges of the boss feature. The low level image processing now involves both the Förstner blob operator and the canny edge operator [2]. Note that the both operators share the same gradient detection stage. A further stage of ellipse fitting [8] are undertaken to give elliptical features.

The template geometry in the image plane is shown in Figure 2. The template is represented by twelve parameters, i.e., $(\mathbf{b}, \mathbf{s_1}, \mathbf{m_1}, n_1, \mathbf{s_2}, \mathbf{m_2}, n_2)$. $(\mathbf{m_1}, n_1)$ models (the size, shape, orientation of) an elliptical edge around a circular feature, where $\mathbf{m_1}$ is the major axis and n_1 is the minor radius. Similarly $(\mathbf{m_2}, n_2)$ gives another ellipse. $\mathbf{s_2}$ defines the shift between the centres of those two edges. $\mathbf{b}$ represents the position of the blob as determined by the Förstner operator, and $\mathbf{s_1}$ is the

difference in the position estimated by the ellipse fitting and the blob operator. The template is deformable since it is defined in image space and varies with the position and pose of the object.

The measure of the goodness of the match of an candidate elliptical edge $(\hat{\mathbf{m}}, \hat{n})$ to the model $(\mathbf{m}, n)$ is defined as follows:

$$f_e = \| \mathbf{m} - \hat{\mathbf{m}} \|^2 + k_1 \cdot (\frac{n}{\| \mathbf{m} \|} - \frac{\hat{n}}{\| \hat{\mathbf{m}} \|})^2 + k_2 \cdot \| \frac{\mathbf{m}}{\| \mathbf{m} \|} - \frac{\hat{\mathbf{m}}}{\| \hat{\mathbf{m}} \|} \|^2 . \quad (4)$$

Here k_1 and k_2 are weights. The first item of above equation relates to the size difference, the second to the shape and the third to the orientation.

Now we are able to define the measure as an energy function as follows

$$E = a_1 \cdot f_{e1} + a_2 \cdot f_{e2} + a_3 \cdot \| \mathbf{s_1} - \mathbf{s_2} \|^2 + a_4 \cdot \| \mathbf{s_1} - \hat{\mathbf{s}}_\mathbf{2} \|^2 + a_5 \cdot \| \mathbf{b} - \hat{\mathbf{b}} \|^2, \quad (5)$$

where f_{e1} is with respect to the first elliptical edge $(\mathbf{m_1}, n_1)$, and f_{e2} to the second one. The blob and edge features within the search area that minimize E are chosen as candidate features.

Using the Gibbs distribution [6], we can have the probability of the 2D template t of a focus feature f:

$$p(t) = \exp(-\frac{E}{T})/N, \quad (6)$$

where T is the temperature, which can simply set to 1 if proper factors $a_i, i = 1, 2, \ldots, 5$ have been chosen, and N is the normalization factor. We simply use this $p(f)$ to replace $p(f_j|s)$ in equation (10) to propagate the information over the Bayes net.

5 A Bayesian Net

We use a causal Bayes net [7] to represent the knowledge about the model. Besides Bayes net, probalistic knowledge representation and Dempster-Shafer can be two alternatives. Figure 3 shows the causal Bayes net for this task. In this net, the node S represents the probability distribution of the position and pose of the object in 3D space, while $F_1, F_2, \ldots, F_n$ respresent the distributions of the features in image coordinates. The object position and pose node S carries the prior distribution $N(\hat{s}_0, \hat{\sigma}_0)$.

Denote $\{f_i | i = 1, 2, \ldots, n\}$ to the object's feature set to be localised. At first we treat each feature individually. With the initial estimation $N(\hat{s}_0, \hat{\sigma}_0)$ for the object and its 3D geometrical features, the probability distribution of each feature f_i over the 3D world space, $p_w(f_i), w \subset R^3$, can be easily computed. This probability distribution is then projected into the 2D image space using known camera geometry. The distribution map $p(f_i)$ of feature f_i over the image is obtained.

Using a given confidence level (0.05 say), we calculate the search bound for the feature f_i. The search bound defines an area such that the probability of the feature lying outside this area is no more that the given confidence level. Within this area, we apply the Förstner blob operator to find candidate blobs. Suppose

blobs $\{b_{ij}|j=1,2,\ldots,m\}$ are found, i.e., we have the distributions $p(v|b_{ij})$. Under the mutual independence assumption, the joint distributions are given by

$$p(v|b_{ij}) = p(f_i) \cdot p(v|b_{ij}). \tag{7}$$

By incorporating the 2D template probability (6) into the above equation. We have

$$p(v) = p(f_i) \cdot p(v|b_{ij}) \cdot p(t_i). \tag{8}$$

We make the assumption that for feature f_i, its corresponding template is that which maximises (8). After we have located the template of the feature f_i, we then be able to update its probability map in the image. so that

$$p(f_i|v) \longleftarrow p(|v). \tag{9}$$

We now use this information (the image location of f_i) to update the knowledge about the model. Let $p(s)$ (where $s = (t_x, t_y, r_z)^T$) denote the distribution of the position and pose of the object. Using Bayes rule, we have

$$p(s|f_i) = \frac{p(f_i|s) \cdot p(s)}{p(f_i)}. \tag{10}$$

$p(s)$ is the prior distribution $N(\hat{s}_0, \hat{\sigma}_0)$, and $p(f_i)$ is the normalisation factor, with $p(f_i) = \int_{s \subset R^3} p(f_i|s)p(s)ds$. Using the estimated transformation $\hat{T}_{wo}$ and the known T_{iw}, for each position and pose s, we are able to establish the projection geometry from object reference to image coordinate:

$$v = T_{iw} \cdot \hat{T}_{wo} \cdot s. \tag{11}$$

Then,

$$p(f_i|s) = p(f_i|v). \tag{12}$$

Having updated the knowledge about the object position and pose, we propagate the evidence to the other nodes of the net. Again, using Bayes rule,

$$p(f_j|v) = \frac{p(s|f_j) \cdot p(f_j)}{p(v)}. \tag{13}$$

Here $p(f_j)$, the prior distribution, can be computed from equations (8) and (9). $p(v)$ is a normalisation factor, with $p(v) = \int_{v \subset R_I} p(s|f_j) \cdot p(f_j)dv$, where R_I refers to the image space.

To compute the probability $p(s|f_j)$, we enforce the constraints on possible object motions. eg. object position varies only on the x-y plane. We then are able to define a 3D (2 dimensions for translations along x and y axes, and one for rotation around z axis) trajectory t repesenting the refinement of the estimates of the object's position and pose s.

This trajectory is represented in a form of parameter space, as illustrated in Figure 4. Here, again subscript w is refered to the world reference and o to the object reference. Feature f_j in the object reference, i.e., $\| O_o f_j \|$ and ϕ, is given by

the 3D geometrical model of the object. For a given feature f_j in world reference, i.e., $\| O_w f_j \|$ and α, we have

$$\begin{cases} r = \alpha + \theta - \phi \\ x_w = \| O_w f_j \| \cdot \cos\alpha - \| O_o f_j \| \cdot \cos(\alpha + \theta) \\ y_w = \| O_w f_j \| \cdot \sin\alpha - \| O_o f_j \| \cdot \sin(\alpha + \theta). \end{cases} \quad (14)$$

This gives the trajectory t, with θ as the trajectory parameter changing from 0

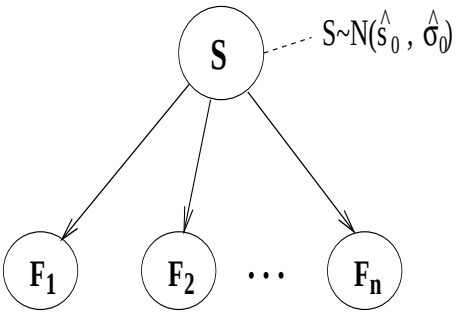

Figure 3: A causal Bayes net

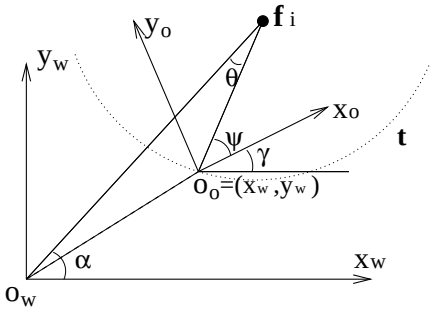

Figure 4: Trajectory of the object pose in the world reference

to 2π. Therefore,

$$p(s|f_j) = \int_t p(s)ds. \quad (15)$$

It should be noted that these constraints on the dof of the object are specific to our task domain, though whatever constraints are used, the equation (15) holds.

6 Implementation

The preceding algorithm was implemented using TINA (AIVRU's own vision system) [9]. A stereo camera rig mounted on the autonomous vehicle COMODE [15] was used to grab images of the object.

To calculate the probability distribution of the object position and pose, we digitalize the 3D space (2D for x and y translations, and 1D for z rotation) with sampling interval of $\sigma_i/20$ and centred at the initial estimate, where σ_i is the standard deviation of the prior estimate. We also set the limits of sampling at $\pm 3\sigma_i$. So for each dimension there are $[3\sigma_i - (-3\sigma_i)]/(\sigma_i/20) = 120$ sampling points. Thus the probability distribution of the object position and pose estimate is represented as a $120 \times 120 \times 120$ volume.

The probability distribution of the location of a feature is computed at every pixel over the image plane. This means 512×512 probability distribution "image" for each feature of the total 3 focus features.

Once a probability distribution of a 2D template is updated, the search bound, or the confindence contour at a given level, is to be computed. Starting from the peak of the distribution, the algorithm iteratively expands the region until the

integration of the probability distribution reaches the given confidence level. The localisation of a feature will then only be performed within its search bound.

7 Experiment

We 'tracked' the positions of the three focus features of the object, using the distributions not exceeding $(N(50mm, 50mm), N(-50mm, 50mm), N(-30°, 30°))^T$ as an initial estimate of the object position and pose. We choose the world reference coordinate to be the same as the object reference, thus the correct position and pose is $(0, 0, 0°)^T$.

Repeating experiments with different initial estimates and with the object at different poses, we found that after two iterations (or passes) a stable (and rather precise as well) estimate of the object position and pose was acquired. After completion of the processing the estimated position and pose of the object is within $(\pm 1.0, \pm 1.0, \pm 1.5°)^T$. This error is within a single step of the quantisation of the space used in the computation of the distributions.

Given a pose estimate of the object, we can directly draw the 2D template of the focus features under calibrated camera geometry. Figure 5 and Figure 6 shows templates before and after object localisation in two experiments. From both figures we can see the 2D templates almost perfectly fit with the image after localisation. This indicate the position and pose of the object is precisely recovered. Figure 7 gives a confidence boundary at level of 0.1 for the final estimate the object position and pose.

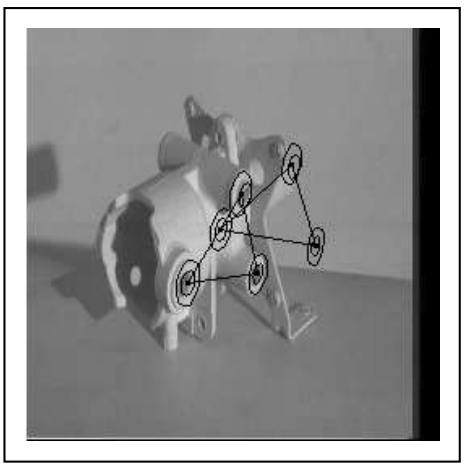

Figure 5: Templates before and after localisation. Those templates with cross at center are computed after localisation

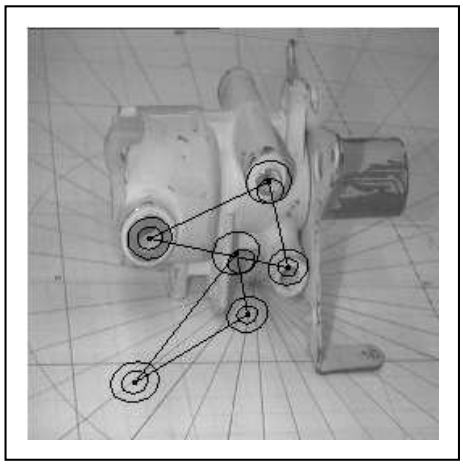

Figure 6: Templates before and after localisation with the object at another pose. Those templates with cross at center are computed after localisation

Figure 8 shows the evolution of the probability distribution $p(f_0|v)$ (so for focus featute #0) during processing. Figure 8 (a) is the prior probability distribution

of the feature. Figure 8 (b) is the updated probability distribution after locating the feature. Figure 8 (c) is the probability distribution after data from all those three features has been incorporated i.e., after the first pass. Figure 8 (d) is the probability distribution after re-localisation of this feature in the second pass.

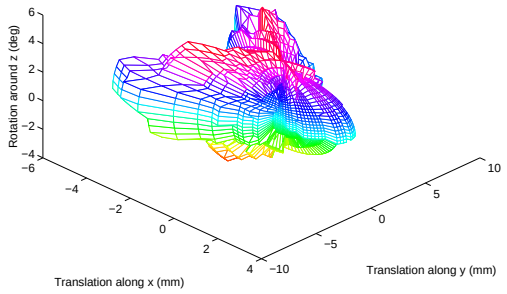

Figure 7: Confidence boundary for the final pose estimate

The shape variations and the centre shifts between Figure 8 (a), (b) and (c), and the slight difference between Figure 8 (c) and (d) indicate that

- the effect of including location information from other features is obvious;
- the major effect of the second pass is to reduce the uncertainty of, rather than the value of the position estimate.

8 Summary and Conclusions

In this paper we have presented preliminary results demonstrating the use of a stochastic framework to solve a structured and constrained task. The overall performance of the system seems most encouraging.

Starting from prior knowledge about the 3D structure of the object and its likely position in the workspace represented as probability distributions, data from preliminary observations are used to update the predicted locations of the other focus features. These search areas are then processed and the the recovered information used to update the position and size of subsequent search areas. The information is propagated using the Bayesian methodolgy.

The work is still in the early stage of development. For example it is not possible to claim any complexity of the task control structure, the current control architecture being simple and ballistic (though possible developments are obvious). For example there is no reasoning concerning the deployment of the vision algorithms; if the system is searching for focus feature #2 then it deploys the vision processing appropriate to verify 'flexible template of the model feature #2'.

In this respect the experiments may be regarded as the first steps towards the development of an architecture in which information appropriate to the particular task domain can be compiled down into the earliest stages of vision processing.

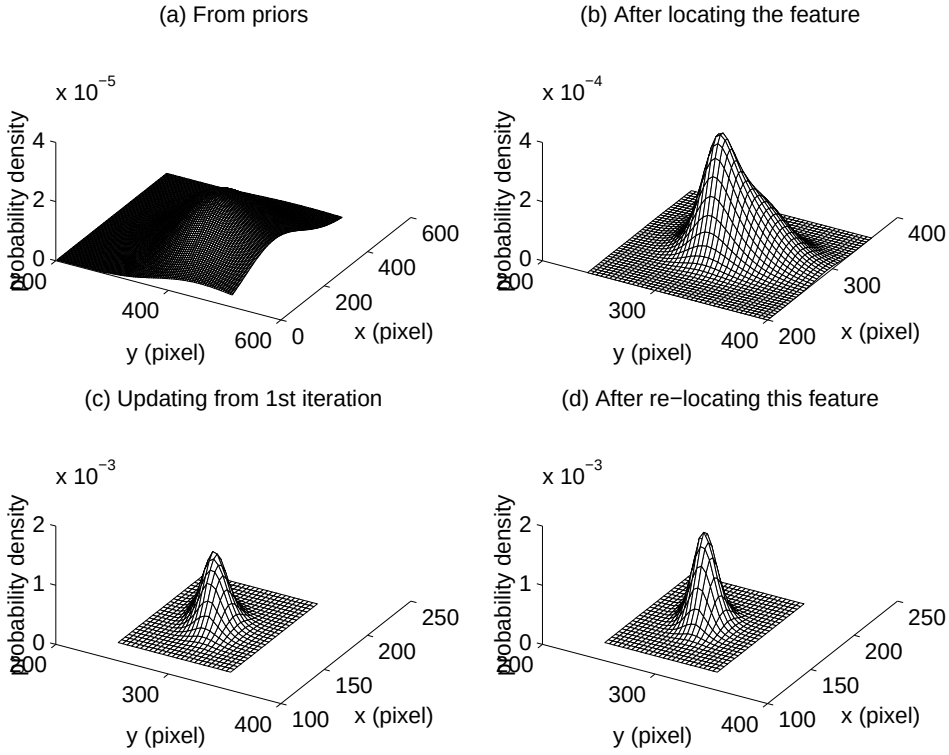

Figure 8: Evolution of probability maps of position of the focus feature #0

Thus in conclusion, if one of the holy grails of vision science is the development of the general vision system [13, 14], our particular version of the search is for a system architecture in which generality is provided by a Bayesian network (or equivalent probabilistic reasoning system); in which communication is by propagation of revised estimates of probability, and where local control is exercised by the deployment of collections of vision algorithms parametrised by the constraints of the particular component of the task for which they are uniquely specialised.

Who knows, it may work!

9 Acknowledgement

Y. Shao is sponsored by Sino-British Friendship Scholarship Scheme. The authors feel indebted to all members of AIVRU at The University of Sheffield.

References

[1] C. Brown, D. Coombs, and J. Soong, Real-tile smooth pursuit tracking, *Active*

vision, ed. A. Blake a,d A. Yuille, MIT, 1992

[2] J. Canny, A computational approach to edge detection, *IEEE Trans. PAMI*, **8**(6), 1988, pp.679-698

[3] O. D. Faugeras, Q. T. Luong and S. J. Maybank, Camera self-calibration–theory and experiments, *Lecture notes in computer science*, **588**, 1992, pp.321-334

[4] W. Förstner, Image matching, *Robot and computer vision*, vol. 2, ed. R. M. Maralick and L. G. Shapiro, Addison-Wesley, 1993

[5] A. Francisco, Active structure acquisition by continuous fixation movements, *Dissertation*, Computational Vision and Active Perception Laboratory, Royal Institute of Technology, Sweden, June 1994

[6] G. Parisi, *Statistical Field Theory*, Addison-Wesley

[7] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*, Morgan Kaufmann, 1988

[8] J. Porrill, Fitting ellipses and predicting confidence envelopes using a bias corrected Kalman filter, *Image and vision computing*, **8**(1), 1990, pp.37-41

[9] J. Porrill, S¿ B. Pollard, T. P. Pridmore, and et al, TINA: The Sheffield AIVRU vision system, *Proc. of 10th Int'l joint conf. on artificial intelligence*, **2**, 1987, pp.1138-1144

[10] R. D. Rimey, Control of selective perception using Bayes nets and decision-theory, *Tech. report* TR-468, Dept. of Computer Science, University of Rochester, Dec. 1993

[11] Y. Shao, S. D. Hippsley-Cox and J. E. W. Mayhew, Ground planeobstacle detection of stereo vision under variable camera geometry using nueral nets, *Proc. of BMVC'95*, **2**, Birmingham, Sept. 1995, pp.217-226

[12] Y. Shao, J. E. W. Mayhew, Object localisation using model-driv en vision, *AIVRU Memo-106*, University of Sheffield, March 1996

[13] M. J. Tarr and M. J. Black, A computational and evolutionary perspective on the role of representation in vision, *CVGIP - image understanding*, **60**(1), 1994, pp65-73

[14] M. J. Tarr and M. J. Black, Reconstruction and purpose–response, *CVGIP - image understanding*, **60**(1), 1994, pp.113-118

[15] N. A. Thacker and J. E. W. Mayhew, Optimal combination of stereo camera calibration from arbitrary stereo images, *Image and vision computing*, **9**(1), 1991, pp.27-32

[16] A. Yuille and P. Hallinam, Deformable templates, *Active vision*, ed. A. Blake and A. Yuille, MIT, 1992