A MODEL-BASED VISION SYSTEM FOR MANIPULATOR POSITION SENSING

by

I. Jane Mulligan, Alan K. Mackworth,
and Peter D. Lawrence

Technical Report 89-13
June 14, 1989

Department of Computer Science
University of British Columbia
Vancouver, B.C. V6T 1W5 Canada

A Model-based Vision System for Manipulator Position Sensing

I. Jane Mulligan †
Alan K. Mackworth †
Peter D. Lawrence *

† *Department of Computer Science*
* *Department of Electrical Engineering*
University of British Columbia
Vancouver, B.C., Canada
V6T 1W5

June 14, 1989

Abstract

The task and design requirements for a vision system for manipulator position sensing in a telerobotic system are described. Model-based analysis-by-synthesis techniques offer generally applicable methods with the potential to meet the system's requirement for accurate, fast and reliable results. Edge-based chamfer matching allows efficient computation of a measure, E, of the local difference between the real image and a synthetic image generated from arm and camera models. Gradient descent techniques are used to minimise E by adjusting joint angles. The dependence of each link position on the position of the link preceding it allows the search to be broken down into lower dimensional problems. Intensive exploitation of geometric constraints on the possible position and orientation of manipulator components results in a correct and efficient solution to the problem. Experimental results demonstrate the use of the implemented prototype system to locate the boom, stick and bucket of an excavator, given a single video image.

1 Introduction

A telerobotic system integrates some autonomous robot control with high-level human supervision. Such a system should incorporate internal models of its environment and itself. As the robot moves its limbs, the perceptual system should use visual information and other senses to provide updates to its internal self-model; however, a typical current hand-eye system has to

hide its arm before looking at the scene. Surely one of the first perceptual tasks for a robot or a telerobot must be to understand images of its own arm(s). Once this has been achieved, visually-guided grasping and coordinated manipulation become possible. Using vision sensors to observe the robot's arms (and legs) further suggests using visual feedback to supplement or replace the traditional inverse kinematic and setpoint methods for path planning and path following.

In this paper we describe a model-based vision system that allows a telerobot to see its manipulator and monitor its position. The success of the system may be seen as a step toward the goal of integrating control-theoretic and model-based approaches to control. A robot manipulator is typically controlled by representing its configuration as a vector of joint angles. Individual servo loops for each joint allow precise control of the manipulator. In our model-based vision system we use an articulated, 3-D model of the limb. We envision using perceptual data to close servo loops, supplementing or replacing traditional proprioceptive joint angle sensor data and allowing for continuous control of the movement of the limb during action.

The idea of using a vision system to determine manipulator position arose in a project to teleoperate a Caterpillar 215B excavator. Joint angle sensing systems in the heavy equipment environment are fragile in that the sensor itself, the connections to the sensor and the wires linking them to controlling computers are vulnerable to mechanical damage (shock, vibration, abrasion) and electrical damage (welding repairs, short circuits due to moisture). Such sensors and their decoding electronics also introduce substantial extra cost and complexity. Furthermore, any flexing of the arm due, for example, to a heavy load, makes sensor readings inadequate to compute link positions. An inexpensive video camera system mounted in the cab could perhaps overcome these difficulties as well as providing augmented visual feedback to a remote operator.

The excavator in question has four joints formed by the rotating *cab*, the *boom*, the *stick* and the *bucket*, as illustrated in Figure 1. For the purposes of this project only the angle of the boom, θ_2, the angle between the boom and the stick, θ_3, and the angle of the bucket, θ_4, will be computed.

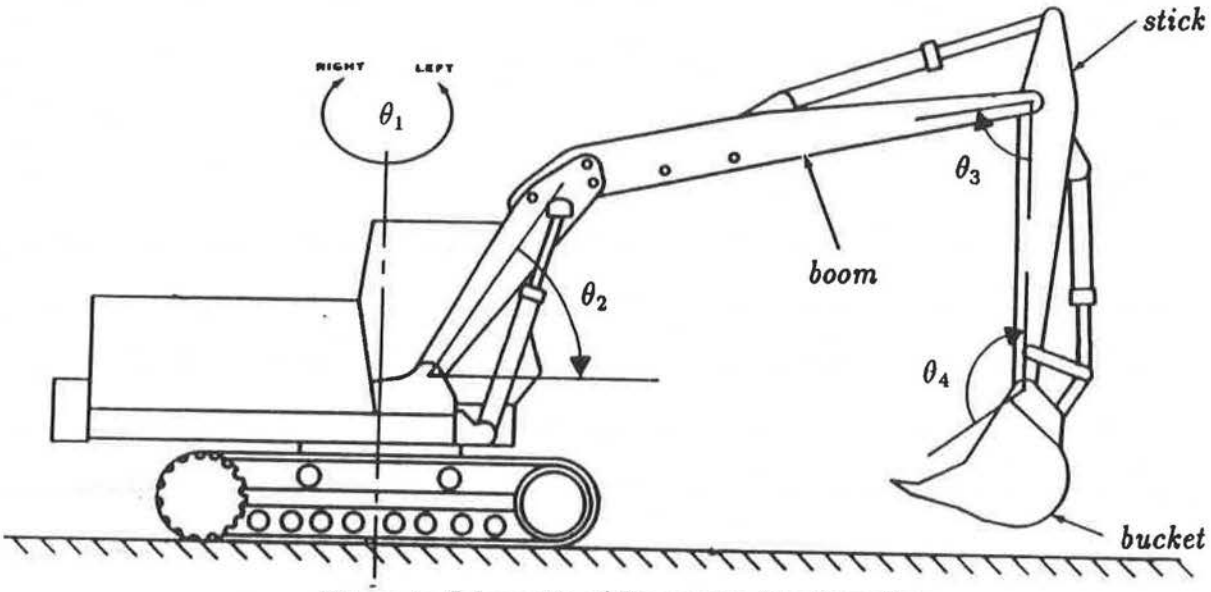

Figure 1: Schematic of Excavator Configuration

The motion of the excavator arm through the image is highly constrained by its geometry. The resulting location and orientation constraints can be efficiently used in parallel to filter irrelevant information out of the image edge data. Model-based chamfer matching can then be applied to the refined edge image to deduce the desired joint angles. The range of positions for the boom relative to the camera is limited; similarly, the position of the stick relative to the boom, and the bucket to the stick are restricted. These global constraints on the joints allow the system to break down the problem and locate the boom first, then the stick relative to it, and the bucket relative to the stick. The same process can be applied to any serial multi-link manipulator.

2 Vision and Joint Angle Sensing

There are a number of critical requirements for position sensors used to control a robot arm or excavator. The first is accuracy: accurate joint angles are essential to accurate positioning and task performance. Speed will also be crucial, particularly if the system is to compete with existing joint angle sensors. The only way to achieve this speed, given the amount of processing required is to use highly parallelizable methods and special-purpose, parallel hardware. The

degree of intrinsic parallelism of proposed algorithms is therefore an important criterion. Finally the system must be reliable over changes in lighting, operating conditions, occlusion and poor image quality. In an industrial environment such a system must be at least as robust as the system it replaces.

The environment of an excavator poses a number of challenges to the design of a computational vision system. Expected image quality is illustrated in Figure 2. First of all it is, in general, outside and subject to the shadows, reflections and variable lighting of the natural world. Further, these machines are not stationary, so variations in lighting, and landmarks in the environment can occur throughout the machine's working day. Immediately then, most of the machine vision techniques applied to robotics requiring controlled lighting, are eliminated. Special fixtures, cameras and LED's used by systems such as SELSPOT or Watsmart [8], most forms of laser range finding and all binary vision techniques are inappropriate for these operating conditions.

A second constraint imposed by the nature of the problem is that surface albedo, special markings [13], lights, or even a camera or fixture mounted on the arm itself are likely to fade, be obscured or knocked off entirely by the routine tasks of heavy equipment. Rain, holes, branches, mud and other environmental hazards will interfere with the use of these attributes. Since the goal is to backup or replace existing joint angle sensors (magnetic or optical) whose wires could be easily damaged, the vision system should avoid having similar weak points.

Recent work in model-based vision suggests that the field is now sufficiently mature to form the basis of engineering applications. Work by Grimson [7] demonstrates its effectiveness for 2-D articulated objects. Lowe uses numeric solutions in a 3-D model-based analysis-by-synthesis paradigm [10]. Model-based techniques have been demonstrated to be accurate and reliable. They offer the best alternative for meeting the requirements of our problem, provided that we can design a system that exploits parallelism to operate in near real time. Special purpose realtime vision systems have recently been built for such constrained robotics tasks as table tennis [2] and throwing and juggling a ball [1].

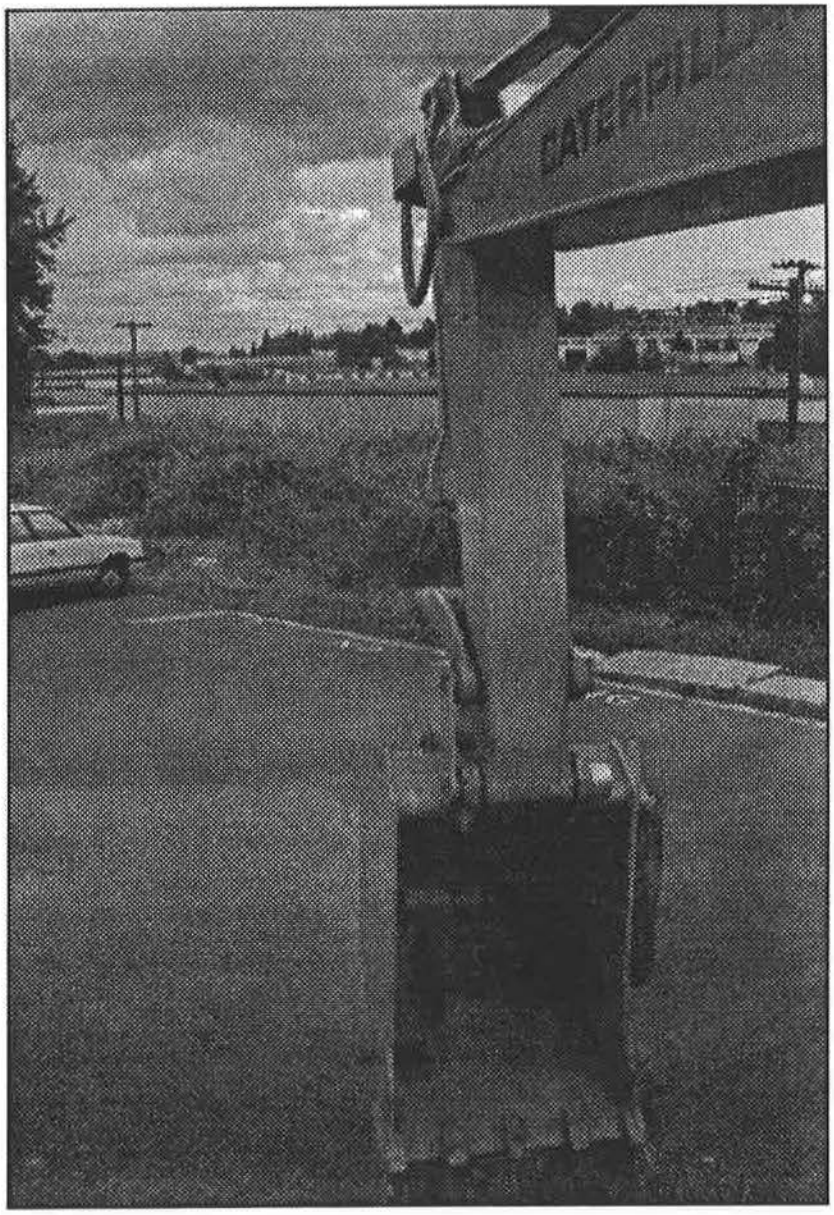

Figure 2: Image 6

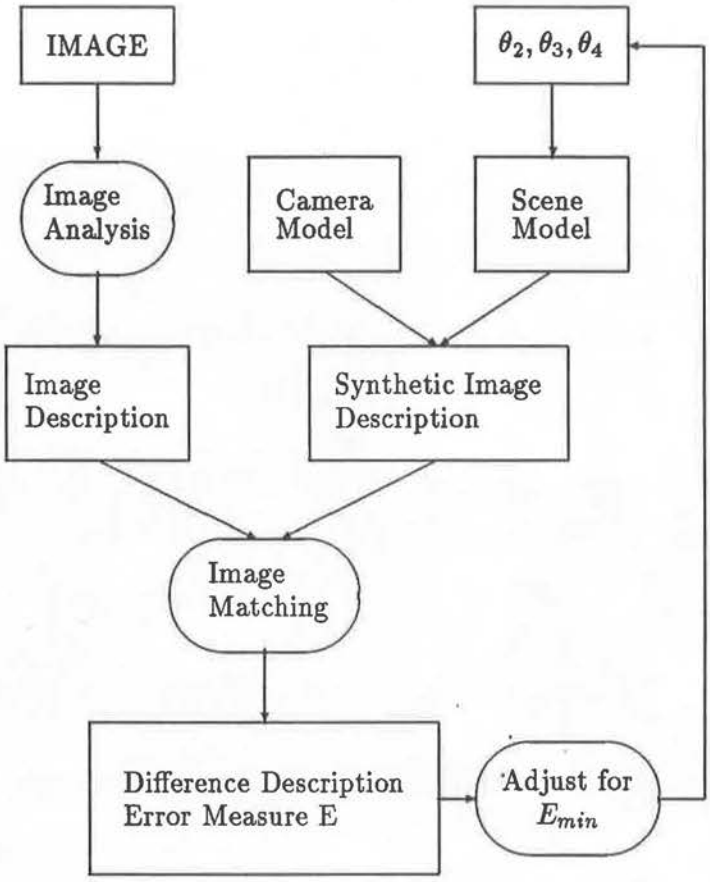

Figure 3: Model-based Analysis-by-synthesis: General framework

3 Model-based Analysis-by-synthesis

The decision to use model-based vision techniques framed the basic *analysis-by-synthesis* approach adopted for this project. An actual image of the manipulator is processed to produce an image description; parameter estimates and arm and camera models are used to produce the description of a *synthetic image*. The descriptions of the two images are compared to provide an error measure E. Parameters determining the synthetic image features are then repeatedly adjusted until E is minimised. The broad outline of this image-based matching scheme is shown in Figure 3.

The question then arises, what are appropriate image descriptions and error measures for such

a framework? From the requirements enumerated in the preceding section the key criterion for an image description is that it be robust over the various environmental variations encountered by the excavator. Perhaps the best such descriptions available to us are edge-based techniques which are fairly stable under such conditions, provided that non-object 'edges' such as highlights and shadows can be explicitly discounted.

There are several characteristics which would be desirable in the error measure E. First low-dimensionality of the parameter space reduces the problem considerably; in other words, we would like to minimise E with respect to as few parameters as possible. For example, if our parameter space were separable, the problem could be broken down into a number of simpler, lower dimensional problems. E should be concave upward, with as few extraneous local minima as possible. Finally we need an E which is fast and cheap to compute, since it will have to be computed for each of several synthetic images until a minimum in E is reached.

4 Chamfer Matching

The chamfer matching process, described by Barrow *et al.* [3], begins by extracting image features, essentially edge feature points, from the input image, retaining shape information while discarding greyscale. Given a *synthetic image* or projection of the 3-D model through the camera model, a comparatively robust measure of similarity between real image edge points $\{\vec{R}\}$ and synthetic image edge points $\{\vec{S}\}$ is E, the sum of the distances between each predicted edge point and the closest image edge point:

$$E = \sum_j \min_i | \vec{S}_j - \vec{R}_i | \tag{1}$$

This measure can be efficiently computed if we transform the image feature array containing $\{\vec{R}\}$, so that each element is assigned a number representing its distance from the nearest real image edge point. The distance values for the pixels where synthetic edge points or predicted features fall can then be summed to produce the desired measure of similarity; this allows comparison of two sets of curve segments at cost proportional to curve length.

The transformation of the $n \times n$ image begins with an array where feature points $\{\vec{R}\}$, are marked with zero, and all other points are marked with infinity. A forward and a backward pass use incremental distances of 2 and 3 to approximate Euclidean distances of 1 and $\sqrt{2}$ computing integral distance values as follows :

```
for i ⇐ 2 step 1 until n do
    for j ⇐ 2 step 1 until n-1 do
        F[i,j] ⇐ minimum (F[i,j],
                  (F[i-1,j] + 2), (F[i-1,j-1] + 3),
                  (F[i,j-1] + 2), (F[i-1,j+1] + 3));

for i ⇐ (n-1) step -1 until 1 do
    for j ⇐ (n-1) step -1 until 2 do
        F[i,j] ⇐ minimum (F[i,j],
                  (F[i+1,j] + 2), (F[i+1,j+1] + 3),
                  (F[i,j+1] + 2), (F[i+1,j-1] + 3));
```

This distance array need only be computed once for the image. Creating the chamfered image requires two passes in serial implementation. The forward raster-order pass propagates distances downward; the backward pass ensures that the distance values for pixels above edge points are given the correct minimum distance, rather than a greater distance propagated down from an edge point above. Relaxation techniques could be used to compute the chamfered image in parallel. The chamfering process actually computes a *medial axis transform* [4], based on the image edge points. Chamfer matching is an unusual application for these transforms which are generally used to represent shapes based on their central skeletons and some weighting information.

The synthetic image is created by projecting the lines of a three dimensional model into the image, using a known camera model and the current estimate for the parameter to be optimised, in this case θ. For each chamfered image several estimates for θ and thus several synthetic images, will be computed and matched, until E is minimised. Standard line scan conversion techniques from computer graphics select pixels in a raster grid which are marked in order to display a line [6]. The same techniques are used here to select the locations in the chamfered image array where the current model line will fall. The chamfer values from these locations are summed to give E

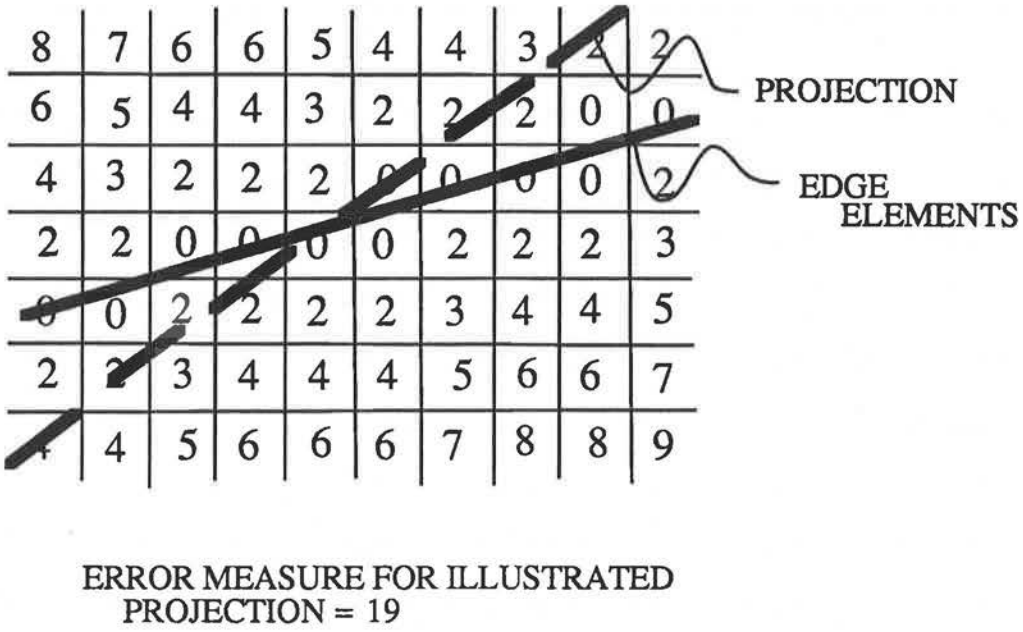

Figure 4: Computing the Error Measure

as illustrated in Figure 4. Given the likelihood of poor image data, another advantage of using E is that it interpolates well over gaps in image lines. E_{min} will have a slightly higher value than it would otherwise, but it will still occur when parameters are close to their actual values.

However, chamfer matching alone only gives us an efficient method to compute E: it does not solve our problem. If we simply chamfer the entire image using all the edge data and search for model components, the system will be very slow to find the solution, and is much more likely to fall into a local minima. The chamfer error measure does not meet our requirements for such a measure as it is relatively high dimensional, it is prone to local minima, and will be slow to converge to the desired solution. As explained in section 6, several constraints from the problem domain must be used to make this technique feasible.

5 Image Processing

The first step in determining arm position in this system is some basic *image processing*, which transforms an eight bit, grey-scale image into a more usable form. Chamfer matching is an *edge based* technique because it compares only the position of edge points in the image to those in the synthetic image. It does not use information based on other image cues such as shading, motion or texture. Methods for extracting edges from the grey scale image must therefore be applied before matching can proceed.

A common strategy, first formalised by Marr and Hildreth [11] is to compute the Laplacian derivative of the image, smoothed with a Gaussian distribution function ($\nabla^2 G$). Since $\nabla^2 G$ is a second derivative operator, the points in the $\nabla^2 G$ image where intensity values change sign (zero crossings), represent points in the original image where intensity changes at a particular scale occurred. Such intensity changes generally arise from edges in the world and thus the zero crossings of $\nabla^2 G$ include the edges projected by objects as illustrated in Figure 5.

More recent work on edge detection by Canny [5] has shown that directional derivatives at varying scales produce better localisation of edges than $\nabla^2 G$. Canny edge detection has the disadvantage of requiring more computation and being only partially parallelisable. In the context of the arm monitoring vision system, the results obtained with Canny edges were not significantly better than those for $\nabla^2 G$, so we chose $\nabla^2 G$ for early image processing in this system.

The known problems of $\nabla^2 G$ include failures of detection and localisation near corners and other regions of high image curvature. It works best on detection and transverse location of straight edges, therefore any process using its results should not require accurate end point locations for image edges. Moreover edge tracking on $\nabla^2 G$ output is unreliable and computationally expensive.

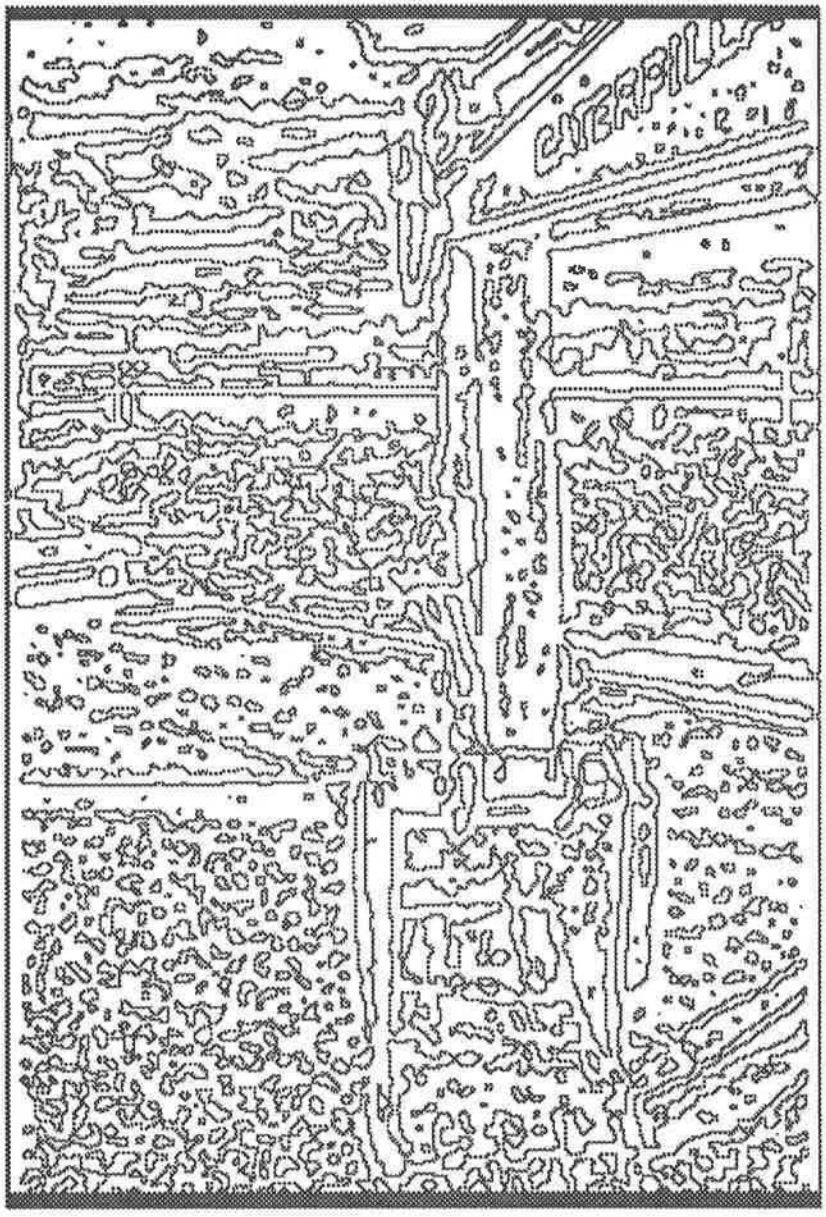

Figure 5: Zero Crossings of $(\nabla^2 G) \otimes I, \sigma = 2$

6 Problem Constraints

The basic geometry of the manipulator itself provides a number of model-based constraints which can be exploited to help determine its position. The model used in the system is as simple as possible: it consists of only the three dominant straight edges for each of the boom, stick and bucket, as shown in Figure 6. The image is constrained by the location of the camera in the cab of the excavator. Only the left upper edge and two lower edges of the boom and stick will ever be visible in the image, and thus only these lines need be used in the model. This has the added advantage that hidden line removal is unnecessary.

6.1 The Quasi-separability Constraint

The nature of serial robot manipulators is such that the position of a link in space is dependent only on the positions and angles of the links and joints that come before it. This is true of the excavator arm and has the useful effect of making the θ parameter space quasi-separable. The position of the boom in the image is dependent only on θ_2, the stick position depends on θ_2 and θ_3 and the bucket on θ_2, θ_3 and θ_4. If we proceed by optimising E for each link sequentially, at each step we are searching for the best value of only one parameter θ_i since:

$$E(\theta_2, \theta_3, \theta_4) = E'(\theta_2) + E''_{\theta_2}(\theta_3) + E'''_{\theta_2,\theta_3}(\theta_4) \tag{2}$$

The parameter space we are searching is essentially reduced to one dimension in θ and one for E as illustrated in Figure 7.

6.2 The Location Constraint

Another constraint of articulated manipulator geometry is that each link has a limited range of motion if, for example, it has a fixed length and the proximal end is constrained to lie at the distal end of the previous link. The excavator arm components each move through their limited angle range in a single plane. These limitations on motion also limit the regions of the image where each component can appear. Attempting to match each manipulator component with edge points

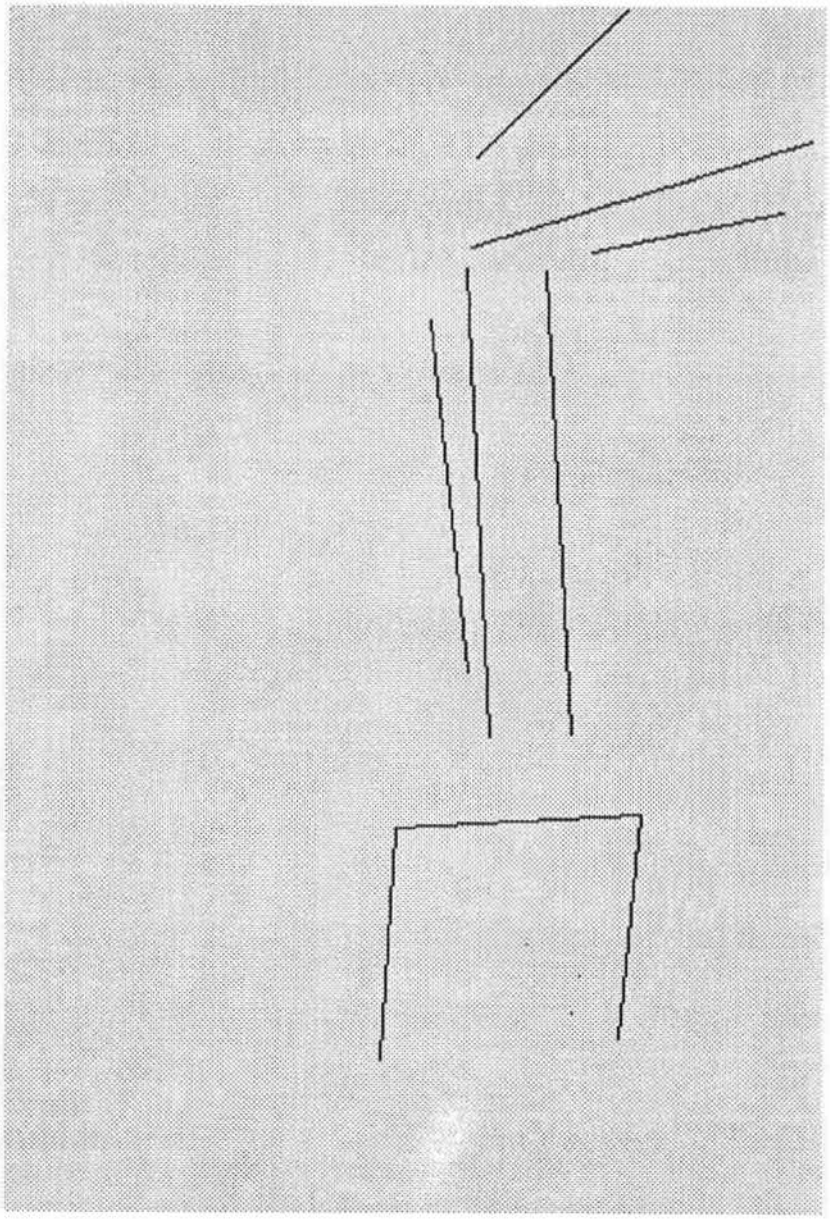

Figure 6: Modelled Arm Components

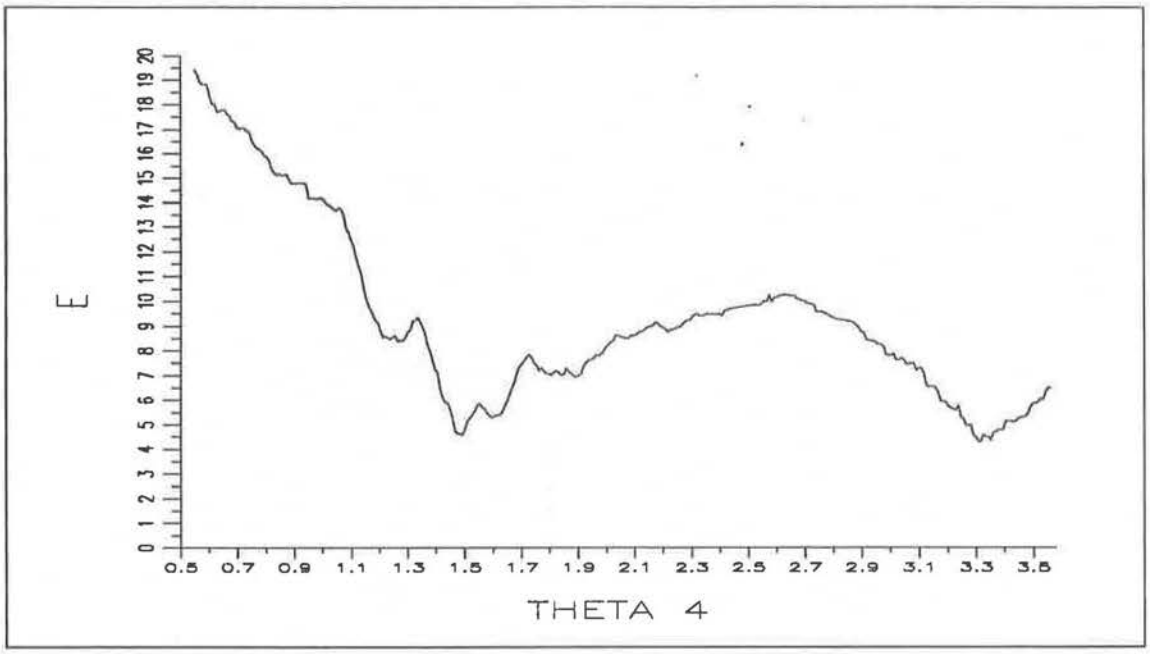

Figure 7: $E - \theta$ Low Dimensional Parameter Space

from the entire image simply did not work, suggesting the use of an essential *location constraint*. For example the range of θ_2 and the shape of the boom, allow it to appear only on the right side of the image, thus we only need chamfer and search this much smaller region in our effort to locate the boom. The stick and bucket windows are similarly limited, but their position and size vary depending on the optimum values for θ_2 and for θ_2 and θ_3 respectively.

6.3 The Orientation Constraint

An attempt to apply chamfer matching to a synthetic image derived from the arm and camera models and the zero crossing edge image of $\nabla^2 G$, using the quasi-separability and location constraints, failed due to the huge number of extraneous edge elements creating local minima in E. It became evident that as many of the irrelevant edge points as possible had to be removed before chamfer matching would become viable. More information about the image had to be included to exploit the problem constraints and refine the edge image before chamfering. The key is to use local context, in a model-based but bottom-up approach, to efficiently eliminate as many as

possible of the image edge elements that could not belong to the component to be located.

Our approach is to use the camera and arm models to create a filter marking each pixel in a component's location window with the overall minimum and maximum orientation of all model lines falling on it. The resulting range can be used to quickly accept or reject edge elements in the window, based on their gradient orientation. The filter is computed by sweeping each model component through its valid range of joint angles at fine increments of θ, updating the range for each filter pixel on which a model line falls. In this way only edge elements which can potentially be part of the arm component are retained.

Figure 8 shows the result of filtering the window in the image where the boom can appear. There is an obvious improvement over the same region of the $\nabla^2 G$ zero crossing image in Figure 5. The boom filter is the most time consuming to compute, but fortunately can be precomputed for a known camera position in the cab. The position of the stick and bucket in the image depend on the proximal joint angles, and their filter size and shape vary depending on image position. At present their filters are computed for each image processed, although potentially a finite number of filters for a range of image positions could be precomputed for these two components.

The nature of the problem then allows us to reduce its complexity, by modelling only easily manipulated straight lines, by determining only one parameter at a time, and by searching for each component in only a limited window in the image. We can further refine the image data by attending only to edge points whose orientation is compatible with that of the current arm component at that position.

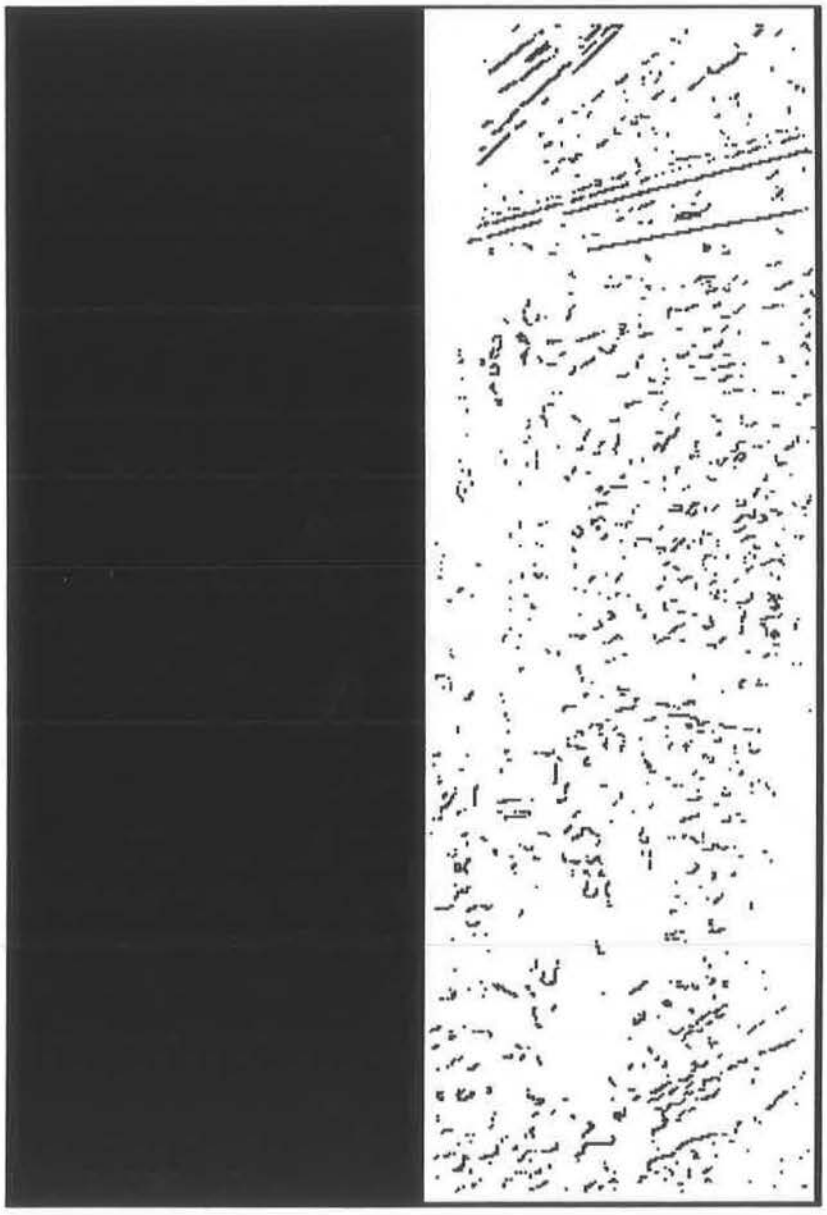

Figure 8: Location and Orientation Filtered Edge Elements for Boom

7 Matching

The input to the implemented system is a digitized image of the excavator arm, taken from the cab. Image processing includes computing $(\nabla^2 G) \otimes I$, $(\nabla_x G) \otimes I$ and $(\nabla_y G) \otimes I$. The zero crossings of $\nabla^2 G$ are marked with their gradient orientation to produce the oriented zero crossing map. For each arm link in turn, given the angle(s) for the previous link(s), an orientation filter is used to select only those zero crossings from the component window whose orientations fall within the allowable range for the component model lines. Using this refined data a chamfered edge window is computed. Optimisation techniques are used to adjust θ until a minimum error measure E, is reached for the current joint angle. The diagram in Figure 9 illustrates this process.

As proposed, the matching process proceeds by finding one angle at a time, beginning with θ_2. Figure 10 illustrates such a solution. Minimising E implies adjusting the current θ_i until the total distance between the synthetic image points and the actual image points is minimised, thus E acts as an L_1 norm. An initial guess for the current joint angle is used to predict the location of component features in the image. The joint angle is then adjusted until the prediction matches the observed image; increases and decreases in the similarity measure indicate how the parameters should be adjusted. For each component edge window chamfered then, several synthetic images will be computed based on adjusted θ values, until the best match (lowest E) is found.

It is important to examine the effect of the constraints the system uses. Basically quasi-separable, local chamfer matching is a single variable optimisation in $E - \theta$ space, where θ is adjusted to minimise E. Without orientation filtering the parameter space is as illustrated in Figure 11, with no clear minimum, particularly at the true θ_3 value of 1.3 radians. The necessity and effectiveness of the orientation filtering process is illustrated by comparing Figure 11 with Figure 12, where there is a clearly defined minimum very close to the true θ_3 value.

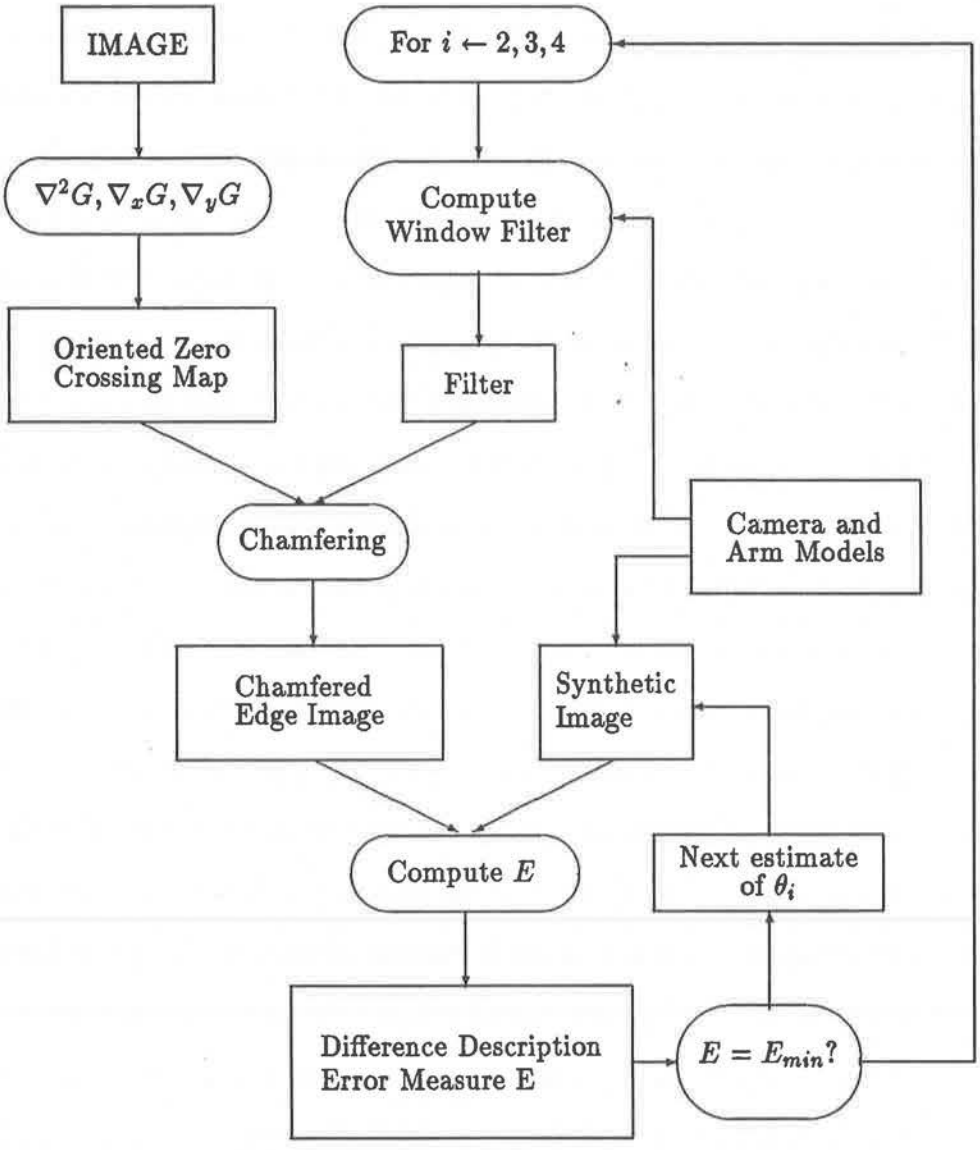

Figure 9: The Algorithm Scheme

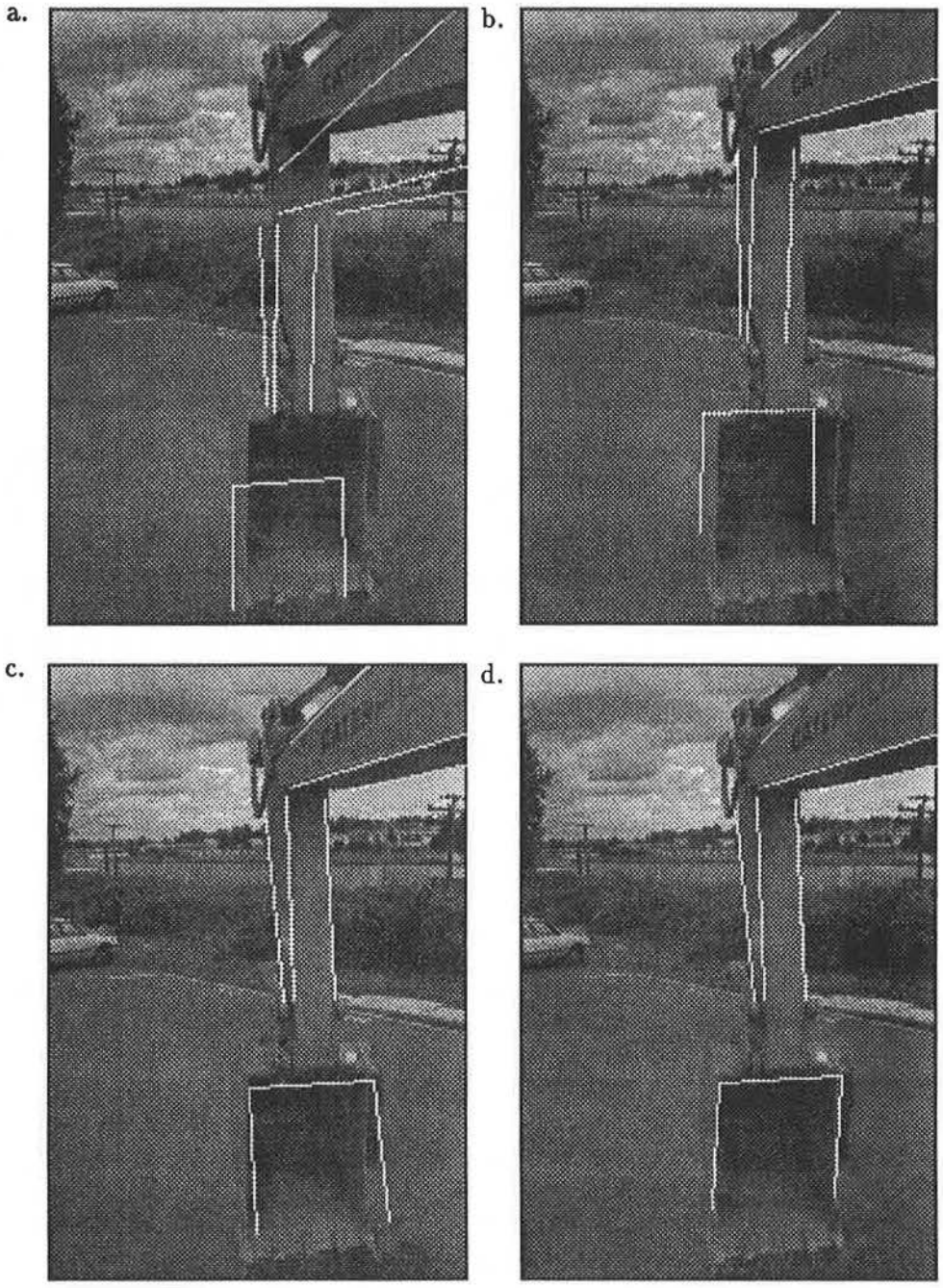

Figure 10: Finding θ_2, θ_3 and θ_4 in sequence. The superimposed models in the above images illustrate a) the initial guess, b) the located boom, c) the located boom and stick, and d) the final computed solution.

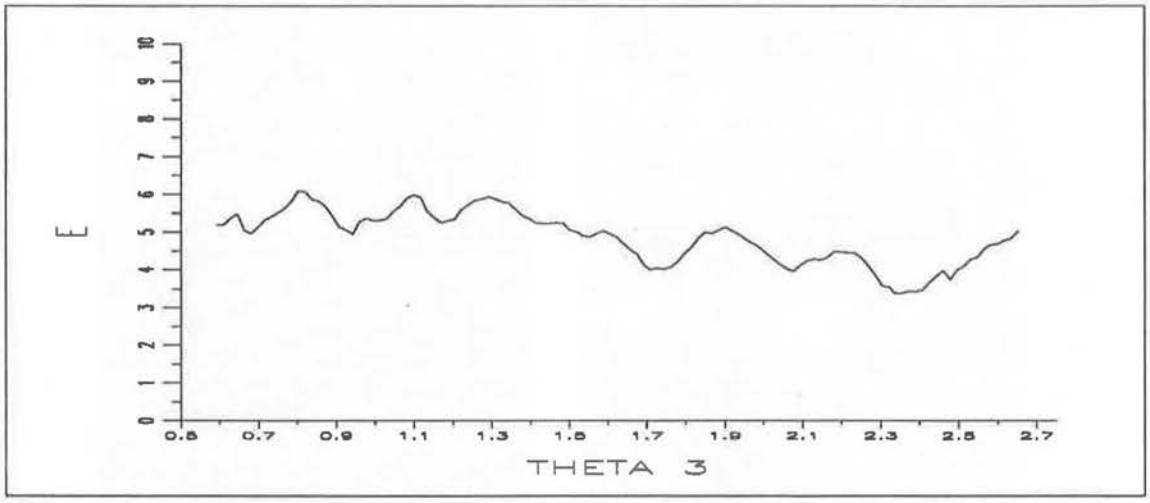

Figure 11: Location Filtered Parameter Space - Image 6, θ_3

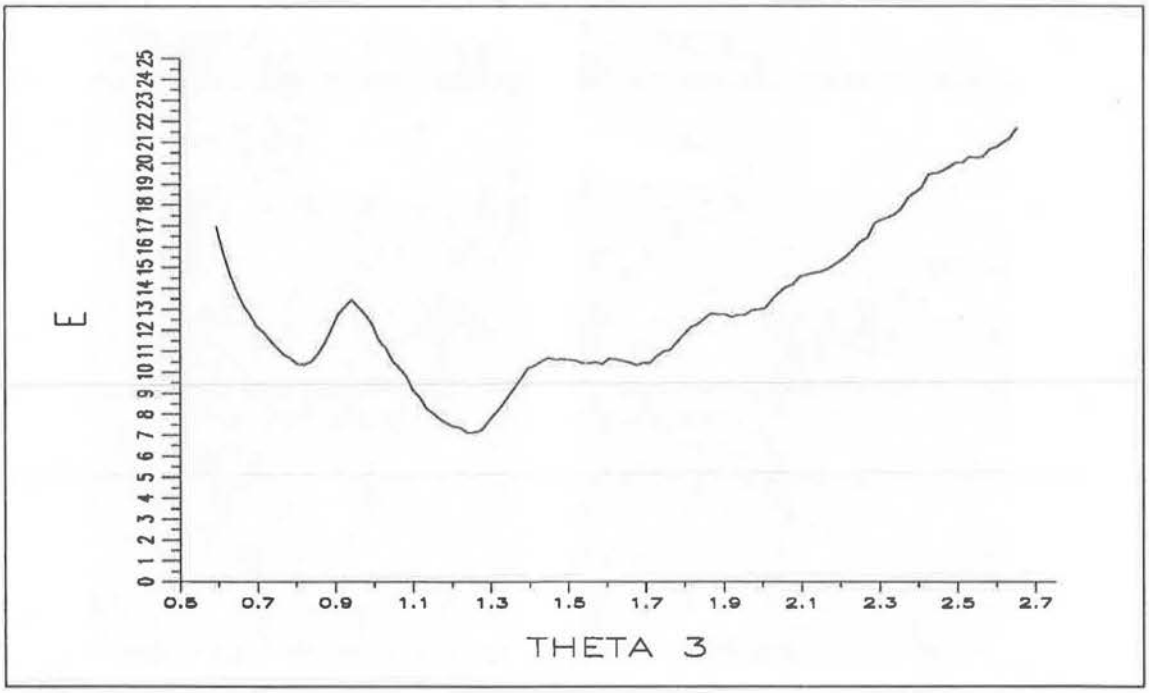

Figure 12: Orientation and Location Filtered Parameter Space - Image 6, θ_3

8 Results

The goal of the arm monitoring vision system is to demonstrate the viability of passive vision techniques for arm position sensing. There are three basic results to be illustrated to establish that the system is useful. Accuracy requires showing that the system will converge to accurate values for the joint angles. Speed is essential if the system is to run at close to frame rates. Although the prototype implementation uses serial techniques, arguments for speedups through the use of equivalent parallel techniques will be presented. Finally, an argument can be made that the system is reliable and robust, although the limited amount of data presented here makes this difficult to verify.

8.1 Accuracy

There are several sources for error in the vision system. First, the data used by the system, including both camera and component models is currently quite crude. Image acquisition is another possible source of error in computing joint angles. In addition, for the distal joint angles any inaccuracy in computing θ_2 affects the accuracy of the estimate for θ_3 and so on, so the greatest error often occurs for the estimated θ_4. Depending on the arm position the bucket is sometimes difficult to locate accurately, particularly when viewed nearly front on. Another factor affecting accuracy is image size. The larger the image the greater the imaged distinction for small changes in θ. The tradeoff, of course, is that large images take longer to process.

Despite all of these potential problems the system performs quite accurately. Using the three joint angles for each of the eight images tested, we have twenty four sample points $(\theta_{true} - \theta_{est})$. The sample mean of this set is 0.0030 radians, the standard deviation is 0.0709 radians. Figure 13 is the histogram of these difference values, and illustrates their roughly normal distribution.

8.2 Speed

In presenting the time required to process an image we note that the boom filter, which is the most costly to compute, is only computed once for each camera model. If we can assume the

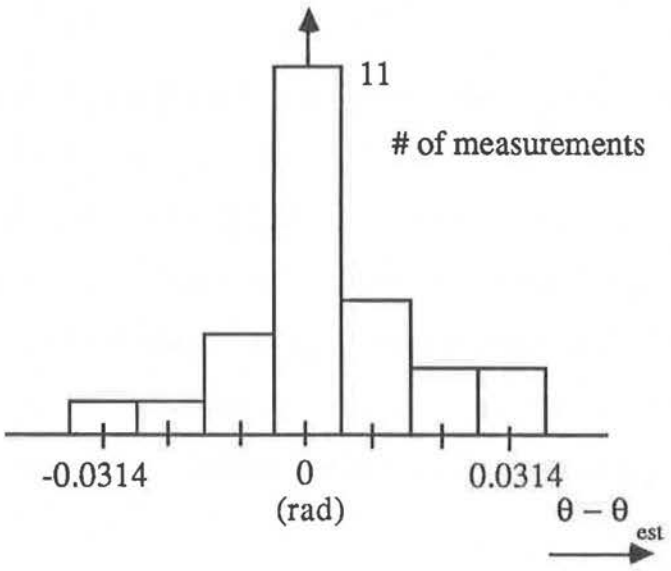

Figure 13: Histogram of Error

camera will require only infrequent adjustment, the boom filter can be precomputed, and not included in the time required to process each image. Without the boom filter then, the entire process in the current implementation takes 92.2 secs per image on a Sun 4/260.

Given our design, it is not surprising that 89% of this time (81.8 secs) is taken up by inherently parallel low level image processing tasks. $\nabla^2 G$, $\nabla_x G$ and $\nabla_y G$ are applied independently to the neighbourhood around each pixel, and given image processing hardware, such as a Datacube system, can be computed at frame rates.

Computing the orientation filters, even excluding the boom, is time consuming: 5.6 secs on the Sun 4/260. Multiple processors would allow each to compute the filter ranges for a very small range of θ. Methods of interpolating between precomputed filters should be investigated. The chamfering process itself requires 4.4 secs on the Sun 4/260. This time can be reduced by parallel implementation so that for window dimension n, the algorithm complexity is reduced from the serial complexity of $O(n^2)$. The parallel complexity is the greater of the maximum distance of any edgepoint to a neighbouring window limit or half the maximum distance between neighbouring edgepoints. The maximum distance for any two such points will be $\sqrt{2}n$ in the worst case. Since

there is still irrelevant data in the filtered edge image the number of iterations will in general be far less than n.

The inner loop of the algorithm, the gradient descent search, requires only 0.4 secs in the prototype. This can be accelerated by computing E for each model line, or even segments of model lines in parallel. At frame rates the arm will not have moved far between images, and the search interval can be made small, which may also improve accuracy, and can be used to limit the extent of chamfering.

In general, if all avenues for parallelising these techniques are exploited, near real time operation seems within reach, but this claim remains to be fully justified.

8.3 Robustness

Robustness is impossible to quantify, but we can say that our system was successful on the eight images tested to date. These are not toy images from a controlled laboratory environment, but natural scenes in a typical excavator environment covering a wide variety of arm configurations and lighting conditions.

The initial angle estimates can deviate, on average, by as much as 0.087 radians for θ_2, 0.227 radians for θ_3, and 0.419 radians for θ_4 and still guarantee convergence to the correct joint angles. Estimates for θ_2 had to be quite accurate because machine geometry caused large changes in image position of the boom for small changes in the joint angle. These ranges seem reasonable for a system running at near frame rates, since the excavator is unlikely to move through larger changes in θ from one image to the next.

Further details on the implementation and results are available [12].

9 Conclusions and Future Work

In this paper a method for arm position sensing using passive model-based vision techniques on a single 2-D image has been described. Using quasi-separability, location constraints and edge based orientation filtering we have provided a simple and efficient method for determining the joint angles of an excavator arm.

The goal of the arm monitoring vision system in the context of telerobotics is to achieve consistent, accurate, real time results. The results for the prototype system are encouraging. Relatively high accuracy and ranges of convergence have been achieved given the quality of measurements and data available to the system. Arguments have also been made that the techniques used are highly parallelizable, and that near real time processing rates can be achieved, given the necessary hardware. In general our techniques have been shown to be fast, accurate, cheap, reliable and, apparently, novel [9].

Possible extensions to the existing prototype include adding a predictor for the joint angles to act as a feedforward component to the system. We could use more image information such as colour and optical flow to help segment out the arm or we could apply spatiotemporal operators to track arm motion. Other enhancements include using multiple orientation maps to separate ranges of valid orientations. The initial goal of this project was to provide a system for position monitoring of the excavator arm, but these techniques could perhaps be extended to track many other kinds of manipulators, and to form the basis for visually-guided manipulation.

10 Acknowledgements

This work was carried out in collaboration with MacMillan Bloedel Research, Robotic Systems International, and Finning Tractor. We are grateful to David Lowe for discussions on model-based vision. Financial support was provided by NSERC, UBC, CICSR, and CIAR. Jane Mulligan is supported by Bell Northern Research, Alan Mackworth is a Fellow of the Canadian Institute for Advanced Research, and Peter Lawrence is a Fellow of the B.C. Advanced Systems Institute.

References

[1] E.W. Aboaf, S.M. Drucker, and C.G. Atkeson. Task-level robot learning: juggling a tennis ball more accurately. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 1290–1295, Scottsdale, Arizona, May 14-19 1989.

[2] R.L. Andersson. *A Robot Ping-Pong Player: Experiment in Real-Time Intelligent Control.* MIT Press, Cambridge, Mass., 1988.

[3] H.G. Barrow, J.M. Tenenbaum, R.C. Bolles, and H.C. Wolf. Parametric Correspondence and Chamfer Matching : Two New Techniques for Image Matching. In *Proceedings of the 5th Anual International Joint Conference on Aritificial Intelligence*, pages 659–663. IJCAI, August 1977.

[4] H. Blum and R. N. Nagel. Shape Description Using Weighted Symmetric Axis Features. *Pattern Recognition*, 10:167–180, 1978.

[5] J.F. Canny. Finding edges and lines in images. Technical Report 720, MIT Artificial Intelligence Laboratory, Cambridge, Mass., June 1983.

[6] J.D. Foley and A. Van Dam. *Fundamentals of Interactive Computer Graphics.* Addison-Wesley Publishing Co., Reading, Mass., first edition, 1982.

[7] W.E.L. Grimson. Recognition of Object Families Using Parameterized Models. *IEEE*, 3:93–101, March 1987.

[8] J.M. Hollerbach. A Review of Kinematic Calibration. In O. Khatib, J.J. Craig, and T. Lozano-Perez, editors, *The Robotics Review 1*, pages 207–242. MIT Press, Cambridge, Mass., 1989.

[9] P.D. Lawrence, A.K. Mackworth, and I.J. Mulligan. Manipulator Arm Position Sensing. Patent, filed February 22, 1988, US Patent allowed. Also filed in Canada, Japan, Europe and Finland.

[10] D.G. Lowe. Three-Dimensional Object Recognition from Single Two Dimensional Images. *Artificial Intelligence*, 31:355–395, 1987.

[11] D. Marr. *Vision.* W.H. Freeman and Co., New York, N.Y., first edition, 1982.

[12] I.J. Mulligan. A computational vision system for joint angle sensing. Master's thesis, University of British Columbia, Vancouver, B.C., October 1988.

[13] H.F.L. Pinkney and C.I. Perratt. A Flexible Machine Vision Guidance System for 3-Dimensional Control Tasks. In *Proceedings of the International Society for Photogrammetry and Remote Sensing, Commision V Symposium, Real-time Photogrammetry - A New Challenge*, Ottawa, Ont., June 1986.