STABILIZED SOLUTION FOR 3-D MODEL PARAMETERS

David G. Lowe*
Computer Science Dept., Univ. of British Columbia
Vancouver, B.C., Canada V6T 1W5

One important component of model-based vision is the ability to solve for the values of all viewpoint and model parameters that will best fit a model to some matching image features. This is important because it allows some tentative initial matches to constrain the locations of other features, and thereby generate new matches that can be used to verify or reject the initial interpretation. The reliability of this process can be greatly improved by taking account of all available quantitative information to constrain the unknown parameters during the matching process. In addition, parameter determination is necessary for identifying object sub-categories, for interpreting images of articulated or flexible objects, and for robotic interaction with the objects.

Our solution for unknown viewpoint and model parameters is based on Newton's method of linearization and iteration to perform the non-linear minimization. This is augmented by a stabilization method that incorporates a prior model of the range of uncertainty in each parameter and estimates of the standard deviation of each image measurement. This allows useful approximate solutions to be obtained for problems that would otherwise be underdetermined or ill-conditioned. In addition, the Levenberg-Marquardt method is used to always force convergence of the solution to a local minimum. These techniques have all been implemented and tested as part of a system for model-based motion tracking, and have been found to be reliable and efficient.

Previous approaches

Attempts to solve for viewpoint and model parameters date back to the work of Roberts [14], but his solution methods were specialized to certain classes of objects such as rectangular blocks. In 1980, the author [8] presented a general technique for solving for viewpoint and model parameters using Newton's method for nonlinear least-squares minimization. Since that time the method has been used successfully in a number of applications, and it also provides the starting point for the work presented in this paper. The application of the method to robust model-based recognition has been described by Lowe [9, 10], McIvor [12], and Worrall, Baker & Sullivan [16]. Verghese & Dyer [15] have used this method for model-based motion tracking. Ishii *et al.* [5] describe the application of this work to the problem of tracking the orientation and location of a robot hand from a single view of LED targets mounted on the wrist. Their paper provides a detailed analysis that shows good accuracy and stability. Recently, Liu *et al.* [7] and Kumar [6] have examined alternative iterative approaches to solving for the viewpoint parameters by separating the solution for rotations from those for translations. However, Kumar shows that this approach leads to worse parameter estimates in the presence of noisy data, so he adopts a similar simultaneous minimization as is used in the work above.

Much work has been published on characterizing the minimum amount of data needed to solve for the six viewpoint parameters (assuming a rigid object) and on solving for each of

* Contact the author (lowe@cs.ubc.ca) for a full-length version of this paper. The author is supported as a Scholar of the Canadian Institute for Advanced Research.

the multiple solutions that can occur when only this minimum data is available. Fischler and Bolles [2] show that up to four solutions will be present for the problem of matching 3 model points to 3 image points, and they give a procedure for identifying each of these solutions. A solution for the corresponding 4-point problem, which can also have multiple solutions under some circumstances, is given by Horaud *et al.* [3]. Huttenlocher and Ullman [4] show that the 3-point problem has a simple solution for orthographic projection, which is a sufficiently close approximation to perspective projection for some applications. In the most valuable technique for many practical applications, Dhome *et al.* [1] give a method for determining all solutions to the problem of matching 3 model lines to 3 image lines. This could be particularly useful for generating starting positions for the iterative techniques used in this paper when there are multiple solutions.

While this work on determining all possible exact solutions will no doubt be important for some vision applications, it is probably not the best approach for practical parameter determination in general model-based vision. One problem with these methods is that they do not address the issue of ill-conditioning. Even if a problem has only one analytic solution, it will often be sufficiently ill-conditioned in practice to have a substantial number and range of solutions. Secondly, all these methods deal with specific properties of the six viewpoint parameters, and there is little likelihood that they can be extended to deal with an arbitrary number of internal model parameters. In addition, these methods fail to address the problem of what to do when the solution is underconstrained. The stabilization methods described in this paper allow an approximate solution to be obtained even when a problem is underconstrained, as will often be the case when models contain many parameters. Possibly the most convincing reason for believing that it is not necessary to determine all possible solutions is the fact that human vision apparently also fails to do so. The well-known Necker cube illusion illustrates that human vision easily falls into a local minimum in the determination of viewpoint parameters, and seems unable to consider multiple solutions at one time.

Stabilizing the solution

As long as there are significantly more constraints on the solution than unknowns, Newton's method will usually converge to a stable solution from a wide range of starting positions. However, in both recognition and motion tracking problems, it is often desirable to begin with only a few of the most reliable matches available and to use these to narrow the range of viewpoints for later matches. Even when there are more matches than free parameters, it is often the case that some of the matches are parallel or have other relationships which lead to an ill-conditioned solution. These problems are further exacerbated by having models with many internal parameters.

All of these problems can be solved by introducing prior constraints on the desired solution that are used in the absence of further data. In many situations, the default solution will simply be to solve for zero corrections to the current parameter estimates. However, for certain motion tracking problems, it is possible to predict specific final parameter estimates by extrapolating from velocity and acceleration measurements, which in turn imply non-zero preferences for parameter values in later iterations of non-linear convergence. The general form of this process for motion tracking would be equivalent to the use of the extended Kalman filter [17], but the predictive component does not play a role in recognition applications.

Any of these prior constraints on the solution can be incorporated by simply adding rows to the linear system constructed on each iteration of Newton's method. Let $\mathbf{J}$ be the Jacobian matrix of partial derivatives with respect to each model or viewpoint parameter, and $\mathbf{e}$ be the vector of error measurements from the current solution to corresponding image features (see the full-length version of this paper for efficient methods for calculating these). Then Newton's method solves the following matrix equation on each iteration for the vector of parameter corrections, $\mathbf{x}$:

$$\mathbf{J}\mathbf{x} = \mathbf{e}$$

The solution can be stabilized by adding rows to this equation specifying prior desired parameter values, $\mathbf{d}$, in the absence of constraints from the image. If we assume that the errors, $\mathbf{e}$, have unit standard deviation, then the prior estimates of the parameter values should be weighted by a diagonal matrix $\mathbf{W}$ in which each weight is inversely proportional to the standard deviation, σ_i, for parameter i:

$$W_{ii} = \frac{1}{\sigma_i}$$

Incorporating these new constraints, we wish to minimize the following stabilized system:

$$\begin{bmatrix} \mathbf{J} \\ \mathbf{W} \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{e} \\ \mathbf{Wd} \end{bmatrix}$$

We will minimize this system by solving the corresponding normal equations (see full-length paper for discussion of the numerical stability of the normal equations):

$$\begin{bmatrix} \mathbf{J}^{\mathrm{T}} & \mathbf{W}^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \mathbf{J} \\ \mathbf{W} \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{J}^{\mathrm{T}} & \mathbf{W}^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \mathbf{e} \\ \mathbf{Wd} \end{bmatrix}$$

which multiplies out to

$$\left(\mathbf{J}^{\mathrm{T}}\mathbf{J} + \mathbf{W}^{\mathrm{T}}\mathbf{W}\right)\mathbf{x} = \mathbf{J}^{\mathrm{T}}\mathbf{e} + \mathbf{W}^{\mathrm{T}}\mathbf{Wd}$$

Since $\mathbf{W}$ is a diagonal matrix, $\mathbf{W}^{\mathrm{T}}\mathbf{W}$ is also diagonal but with each element on the diagonal squared. This means that the computational cost of the stabilization is trivial, as we can first form $\mathbf{J}^{\mathrm{T}}\mathbf{J}$ and then simply add small constants to the diagonal that are the inverse of the square of the standard deviation of each parameter. If $\mathbf{d}$ is non-zero, then we add the same constants multiplied by $\mathbf{d}$ to the right hand side. If there are fewer rows in the original system than parameters, we can simply add enough zero rows to form a square system and add the constants to the diagonals to stabilize it.

Forcing convergence

Even after incorporating this stabilization based on a prior model, it is possible that the system will fail to converge to a minimum due to the fact that this is a linear approximation of a non-linear system. We can force convergence by adding a scalar parameter λ that can be used to increase the weight of stabilization whenever divergence occurs. The new form of this system is

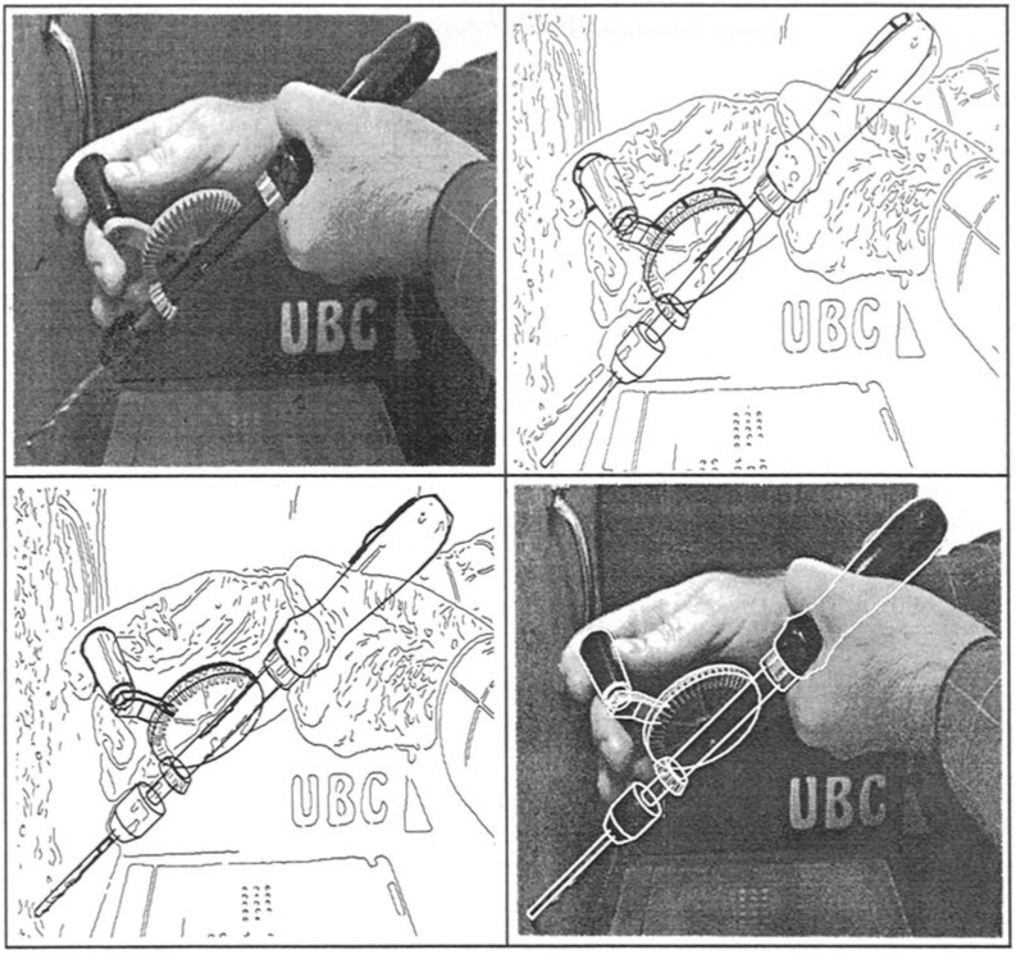
$$\begin{bmatrix} \mathbf{J} \\ \lambda\mathbf{W} \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{e} \\ \lambda\mathbf{Wd} \end{bmatrix}$$

This system minimizes

$$\|\mathbf{Jx} - \mathbf{e}\|^2 + \lambda^2 \|\mathbf{W}(\mathbf{x} - \mathbf{d})\|^2$$

This as an example of regularization using a Tikhonov stabilizing functional, as has been applied to many areas of low-level vision (Poggio *et al.* [13]). In this case, the parameter λ controls the trade-off between approximating the new data, $\|\mathbf{Jx} - \mathbf{e}\|^2$, and minimizing the distance of the solution from its original starting position, $\mathbf{d}$, prior to non-linear iteration, $\lambda^2\|\mathbf{W}(\mathbf{x} - \mathbf{d})\|^2$.

The use of this parameter λ to force iterative convergence for a non-linear system was first studied by Levenberg and later reduced to a specific numerical procedure by Marquardt [11]. Marquardt did not assume any prior knowledge of the weighting matrix $\mathbf{W}$, but instead estimated each of its elements from the euclidean norm of the corresponding column of $\mathbf{J}^{\mathrm{T}}\mathbf{J}$. In our case, the availablity of $\mathbf{W}$ allows the algorithm to perform much better when a column of $\mathbf{J}^{\mathrm{T}}\mathbf{J}$ is near zero. When the solution fails to improve on any iteration, increasing the value of λ (by factors of 10, as suggested by Marquardt) will essentially freeze the parameters having the lowest standard deviations and therefore solve first for those with higher standard deviations.



Figures 1–4: Parameter solving during a motion tracking sequence.

Results of implementation

One initial application of these methods has been to the problem of motion tracking. A Datacube image processor is used to implement Marr-Hildreth edge detection in real time on 512 by 485 pixel images. The image containing these edge points is transferred to a Sun 3/260, where the edges are linked into lists on the basis of local connectivity. A fairly simple matching technique is used to identify the image edges that are closest to the current projected contours of a 3-D model. The few best initial matches are used to perform one iteration of the viewpoint solution, then further matches are generated from the new viewpoint estimate. Up to 5 iterations of this procedure are performed. For simple models with straight edges, all of these steps can be performed in less than 1 second, resulting a system that can perform robust but rather slow real-time motion tracking. Full details of the components of this system other than parameter solving will be published in a separate paper.

Figures 1–4 show the operation of the system for one frame of motion tracking. However, due to the complexity of the model, this version requires about 6 seconds of processing per frame and does not operate in real time. Figure 1 shows an image of a hand drill from which edges are extracted. A simple matching algorithm is used to identify image edges that are close to the projected model curves. These matches are ranked according to their length and average separation, and the best ones are chosen for minimization. The selected matches are shown with heavy lines in Figure 2 along with the perpendicular errors between model and image curves that are minimized. After one iteration of model fitting, the new model position is shown in Figure 3 along with a new set of image matches generated from this position. Note that the rotation of the handle is a free parameter along with the viewpoint parameters. After this second iteration of convergence, the final results of model fitting are shown superimposed on the original image in Figure 4. Note that due to occlusion and errors in low-level edge detection, this final result is based on only a small subset of the predicted image edges. However, due to the overconstrained nature of the problem, in which far more measurements are available than unknown parameters, the final result can be reliable and accurate.

References

[1] Dhome, M., M. Richetin, J.T. Lapresté, and G. Rives, "Determination of the attitude of 3-D objects from a single perspective view," *IEEE PAMI,* **11,** 12 (1989), 1265–78.

[2] Fischler, Martin A. and Robert C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM,* **24,** 6 (1981), 381-395.

[3] Horaud, R., B. Conio, O. Leboulleux, and B. Lacolle, "An analytic solution for the perspective 4-point problem," *Proc. Conf. Computer Vision and Pattern Recognition,* San Diego (June 1989), 500–507.

[4] Huttenlocher, Daniel P., and Shimon Ullman, "Object recognition using alignment," *Proc. First Int. Conf. on Computer Vision,* London, England (June 1987), 102–111.

[5] Ishii, M., S. Sakane, M. Kakikura and Y. Mikami, "A 3-D sensor system for teaching robot paths and environments," *The International Journal of Robotics Research,* **6,** 2 (1987), pp. 45–59.

[6] Kumar, Rakesh, "Determination of camera location and orientation," *Proc. DARPA Image Understanding Workshop,* Palo Alto, Calif. (1989), 870–879.

[7] Liu, Y., T.S. Huang and O.D. Faugeras, "Determination of camera location from 2-D to 3-D line and point correspondences," *IEEE PAMI,* **12,** 1 (1990), 28–37.

[8] Lowe, David G., "Solving for the parameters of object models from image descriptions," *Proc. ARPA Image Understanding Workshop* (College Park, MD, April 1980), 121–127.

[9] Lowe, David G., *Perceptual Organization and Visual Recognition* (Boston, Mass: Kluwer Academic Publishers, 1985).

[10] Lowe, David G., "Three-dimensional object recognition from single two-dimensional images," *Artificial Intelligence,* **31,** 3 (March 1987), pp. 355-395.

[11] Marquardt, Donald W., "An algorithm for least-squares estimation of nonlinear parameters," *Journal. Soc. Indust. Applied Math.,* **11,** 2 (1963), 431–441.

[12] McIvor, Alan M., "An analysis of Lowe's model-based vision system," *Proc. Fourth Alvey Vision Conference,* Univ. of Manchester (August 1988), 73–78.

[13] Poggio, Tomaso, Vincent Torre and Christof Koch, "Computational vision and regularization theory," *Nature,* **317,** 6035 (Sept. 1985), 314–319.

[14] Roberts, L.G., "Machine perception of three-dimensional solids," in *Optical and Electro-optical Information Processing,* eds. J. Tippet *et al.* (Cambridge, Mass.: MIT Press, 1965), 159-197.

[15] Verghese, G. and C.R. Dyer, "Real-time model-based tracking of three-dimensional objects," *Univ. of Wisconsin, Computer Sciences TR 806* (Nov. 1988).

[16] Worrall, A.D., K.D. Baker and G.D. Sullivan, "Model based perspective inversion," *Image and Vision Computing,* **7,** 1 (1989), 17–23.

[17] Wu, J.J., R.E. Rink, T.M. Caelli, and V.G. Gourishankar, "Recovery of the 3-D location and motion of a rigid object through camera image," *Inter. Journal of Computer Vision,* **3** (1988), 373–394.