An Object-Oriented Approach to Scene Reconstruction

Richard I. Hartley
G.E. CRD,
Schenectady, NY, 12301.
email : hartley@crd.ge.com

Abstract

A program is described for carrying out least-squares camera modelling and scene reconstruction from a set of image and scene measurements of geometric features. Because of the object-oriented nature of the program, it is easily extendible to include very general types of camera, image feature or measurement. A Levenberg-Marquardt parameter estimation algorithm is used to optimize the choice of camera and feature parameters to fit the measurements. A structured sparse technique is used to obtain speediest performance on large problems.

1. Introduction

This paper describes a program called Carmen for carrying out camera modelling and scene reconstruction using an iterative least-squares method. Carmen has gone through several different versions over several years. This paper describes the latest version. A basic requirement of Carmen is to be capable of handling many different sorts of camera models simultaneously. These should include perspective (pinhole) cameras, linear pushbroom cameras ([6]), orbiting pushbroom cameras, panoramic cameras ([4]), but it should be possible to add any other camera model with minimum of effort, and without recompilation of the previously existing code. In addition, recent work by several authors ([3, 14]) has dealt with reconstruction from points as well as lines. Furthermore, recent work of Quan raised the possibility of using conics in the reconstruction process. The program therefore had to be able to deal with these object types while at the same time maintaining the possibility of extending to images containing other geometric features.

Many special methods have been developed for reconstruction and camera modelling from several views. A lot of effort has been expended on finding efficient linear or non-iterative algorithms for reconstruction ([2, 3, 11, 9]). However, in most cases ultimate precision is obtained by following up the linear methods with a full-scale least-squares minimization technique, based on the Levenberg-Marquardt or some other least squares algorithm. The object of Carmen was to provide a framework in which least-squares minimization can be carried out on a wide variety of cameras viewing a wide variety of different geometric features.

2. Program Structure

Overview. The overall purpose of Carmen is to carry out an estimate of camera (or sensor) parameters and 3D spatial feature parameters by analyzing measurements taken on images of those spatial objects. Measurements of the 3D spatial objects and the cameras themselves are also taken into account.

Parameters of cameras typically include the location and orientation of the camera (external parameters) as well as camera calibration information (internal parameters), though different types of sensor may be parametrized differently. In some instances, some of the parameters of the cameras may be known. For instance, if the camera is calibrated, then the internal parameters may be known, but the external parameters may not be. Alternatively, it may be assumed that some parameters of the cameras are shared by several or all the cameras. For instance, a common assumption is that all the cameras have the same (unknown) internal parameters.

For 3D spatial features, examples of parameters are the location of points, Grassman or Plucker coordinates of lines, or position and radius of spheres. As with camera parameters, some of the parameters of the 3D spatial features may be known within specified tolerances.

Measurements are made in the images and used as a basis for estimating the unknown parameters of the cameras and 3D features. Measurements in the images may be of quite a general nature, such as measurements of coordinates of the image of a point, identification of a point that lies on the image of a line, or the outline of the image of a sphere. More complex measurements are possible, and allowed for

in the program design. Measurements may also be made directly of the 3D spatial features or the cameras.

Measurements are made with a specified tolerance or error distribution, and in general, the measurements will not coincide exactly with any realizable configuration of cameras and spatial features. Each given value of the variable parameters of the cameras and spatial objects leads to a corresponding predicted configuration of image features in each of the images. Comparing each predicted 2D feature with the specified measurements of that feature leads to an error (in general a vector). The same is true of measurements of cameras and spatial objects. The goal of the program is to find that set of parameters for the cameras and spatial features that minimizes the total squared error of all the measurements. The exact formula is given below in (1). Thus, the problem is formulated as a standard parameter minimization problem, and is carried out using the Levenberg-Marquardt method ([10]).

Sparse Techniques. As has been observed previously ([12, 5]) the special nature of the camera modelling problem allows sparse matrix techniques to be used with enormous savings running time and storage. The general model is as follows. There exist a number of geometric features in space, denoted $\mathbf{x}_i$. These are viewed by a set of cameras, each camera projecting a feature $\mathbf{x}_i$ into an image. The j-th camera is represented by a mapping P^j which takes a 3D feature to a 2D feature $\mathbf{u}_i^j = P^j\mathbf{x}_i$ in the j-th image. The special nature of the camera estimation problem is that images of individual points are independent. That is, the parameters of the feature $\mathbf{x}_i$ do not have any effect on $\mathbf{u}_{i'}^j$ for any i' not equal to i. It is this sparse nature of the data that differentiates this particular problem from a general least-squares parameter optimization problem, and makes the application of structured sparse techniques effective. How this is done is described in detail in [12, 5].

Extensibility. An object-oriented structure of Carmen was chosen in order to allow easy extensibility of its capabilities. The language used was C++. It is clearly not possible to write a program that will handle all types of cameras, geometric features, or measurements of those features. However, a user may easily extend the program to add new camera types, spatial objects, or new ways of measuring these objects just by adding new subclasses derived from generic object classes.

There are several important classes of object : geometric features; cameras or sensors ; mappings from 3D to 2D geometric features; measurements and measurement mappings. These object types will now be considered one by one.

Geometric features. The geometric features are objects such as points, lines or conics in the image, or in space. More complicated geometric features may be added at will by defining a new derived class. Examples are spheres in space, parametrized curves, or solids of revolution. Geometric features are mathematically defined quantities, represented by a set of parameters. For instance, points are represented either as homogeneous or non-homogeneous vectors. Lines are represented by pairs of points, or Plucker coordinates. Plane conics are represented by symmetric matrices. Each different type of feature may be described by a different number of parameters.

Camera Models. Sensors are characterized by a set of parameters. For instance, the common pinhole camera model is defined by a specification of its external parameters (location and orientation) and internal parameters (calibration). More exact models for perspective cameras may include parameters for radial distortion, or other effects. Other camera models may similarly be described by sets of parameters. For instance, the linear pushbroom model ([6]) is described by a set of 11 parameters.

Camera Mappings. A camera mapping is a specification of how a given camera model maps geometric features in space to geometric features in an image. For instance for the pinhole camera model and for space points $\mathbf{x}$ represented by homogeneous vectors the corresponding image point $\mathbf{u}$ is given in homogeneous coordinates by the formula $\mathbf{u} = P\mathbf{x}$, where P is a 3×4 matrix which may be written in terms of the camera parameters. It is necessary to provide a camera mapping for each camera/geometric feature pair of interest. New mappings are created as a subclass derived from a generic camera mapping class.

Measurements. Measurements may be taken of any 3D or 2D geometric feature, or of any camera. The measurements may be directly of the parameters of the object, or other more general measurements. Measurements are rarely absolutely precise, so provision is given for specifying a standard deviation for such measurements. As an example, if the 3D or 2D features are points in space and the image, then it is possible to measure the coordinates of these points

directly. Similarly, it may be possible to measure the focal length or principal point of a pinhole camera. In these cases, the parameters are measured directly.

In other cases, measurements are not made directly on the parameters. For instance a 2D line will normally be represented as a homogeneous 3-vector, the projective coordinates of the line. Instead of measuring the vector coordinates directly, typically one measures the location of a line in an image by identifying a number of points on the line. Alternatively, a set of directed edgels may be provided by a edge extractor program. A similar situation is true with conics in an image. A conic may be parametrized by the entries of the symmetric 3×3 matrix C that determines the conic. One does not measure entries of the symmetric matrix directly in the image, however. Instead, a set of points supposed to lie on the conic may be identified.

In order to handle quite general forms of measurement, a measurement class is defined, along with a measurement mapping class. The measurement mapping compares a measurement with a "measurable entity" (predicted feature or camera model) and returns an error vector. The goal of the parameter minimization procedure is to minimize the sum of squares of the entries of this error vector, summed over all measurements.

The general measurement setup is shown in Fig 1. As an example, consider the case of a line in the image. Such a line is represented by a vector $\boldsymbol{\lambda} = (\lambda, \mu, \nu)^\top$, which denotes the line in the image with equation $\lambda u + \mu v + \nu = 0$, where u and v are image coordinates. A measurement of this line may consist of a point $\mathbf{u} = (u, v)^\top$ intended to lie on the line. An appropriate error vector (in this case a 1-vector) returned by the measurement mapping is the perpendicular distance of the point to the line, namely

$$d(\mathbf{u}, \boldsymbol{\lambda}) = (\lambda u + \mu v + \nu)/\sqrt{\lambda^2 + \mu^2} \ .$$

Measurement of point location. As another example consider the measurement of the location of a point in an image. In many cases, the position of the point may be measured directly. The error may be supposed to have circularly symmetric (isotropic) gaussian probability distribution, with variance, v. In this case, the measurement $\hat{\mathbf{u}}$ and measurable feature $\mathbf{u}$ are both points, and the measurement mapping returns a vector equal to the weighted difference between these two points, $v^{-1/2}.(\hat{\mathbf{u}} - \mathbf{u})$. More generally the location of the point in the image is not measured with an isotropic error distribution. For instance, if the point lies on an edge its position may

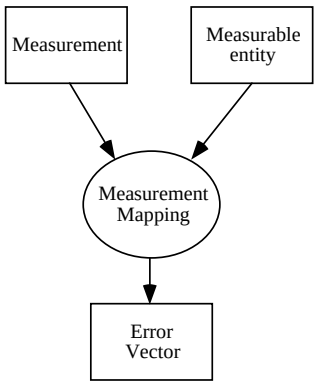


Figure 1: Measurement of a Geometric feature of Camera

be measured with greater precision in the direction of the edge normal than in the direction along the edge. In this case, one might assume a gaussian error distribution with covariance matrix C. One may write $C = U^\top D U$, where D is a diagonal matrix with entries representing the variance of the distribution in two principal axial directions, and U is a rotation matrix specifying the orientation of the axes of the distribution. In this case, the measurement mapping returns the vector $D^{-1/2}U(\hat{\mathbf{u}} - \mathbf{u})$. In the limiting case, the location of the point in the along-edge direction may be completely unknown, having infinite variance. This case is handled by letting $D^{-1/2}$ have a zero entry in the corresponding axial direction.

To add new ways of measuring geometric features, one defines a new measurement class, and the appropriate measurement mapping as a subclass of the generic mapping class.

Primaries and Secondaries. Geometric features are divided into two types, represented by classes called Primary and Secondary. A primary geometric feature is one for which the parameters are independently varying values. Generally, this is synonymous with the concept of a 3D or object-space geometric feature. Such a primary feature is mapped by a camera into an image. The resulting geometric feature is an object of class Secondary. The parameters of such a secondary feature are not independent variables, but rather depend on the parameters of the camera model and the primary feature from which it is derived. The structure of the relationships between the different types of classes of objects is illustrated in Fig 2.

2D mappings. It should be noted that although the discussion so far has been in terms of camera

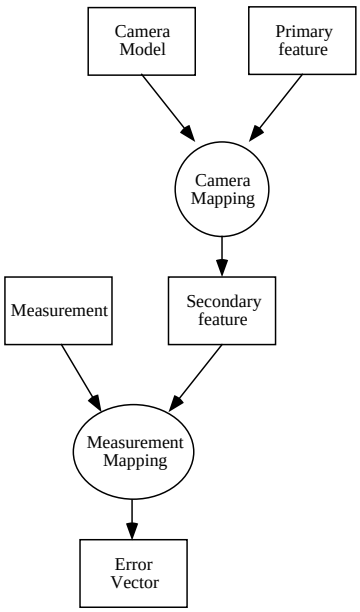


Figure 2: Class Structure. Measurements of the camera model and primary feature are also possible, though not shown here.

mappings from 3D object space to 2D image space, this is in no way enforced by the structure of Carmen. As an example, consider the problem of projective alignment of two 3D or 2D data sets. Consider a set of points A and another set A' which differ by an unknown projective transformation. The point-to-point correspondence between the two sets is assumed known. Thus, there exists a projective transformation that takes points in A' as nearly as possible to corresponding points in A. The task is to find the unknown projective transformation. This problem may be handled by Carmen. The set A' is the set of primaries. The set A is the set of measurements, and the unknown projective transformation is the camera mapping, parametrized by the unknown entries of the matrix representing the projective transformation. The measurement mapping returns the difference between the measured points in A and the points in A' transformed by the projective mapping. It is worth noting that there exists a linear solution to this problem essentially similar to the Direct Linear Transformation camera resectioning method ([13]) but it does not minimize geometric distances. This method may be used to initialize the iterative method used in Carmen.

3. Minimization

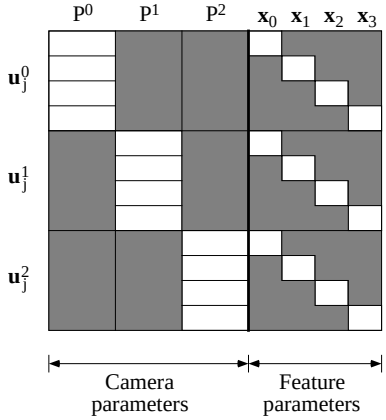
The quantity that we wish to minimize may be expressed as follows Let $\{\mathbf{x}_i\}$ be the set of primary features and $\{P^j\}$ be the set of cameras. For simplicity, we assume (in this discussion only) that each primary is imaged by each camera. For each primary/camera pair, a suitable camera mapping exists. Let $\mathbf{u}_i^j$ be the secondary feature obtained by applying the appropriate mapping for camera P^j to primary $\mathbf{x}_i$. Suppose that there are N_{ij} measurements of the feature $\mathbf{u}_i^j$, (The value of N_{ij} will be zero if feature $\mathbf{x}_i$ is not seen by camera P^j.) Let $\delta_k(\mathbf{u}_i^j)$ be the error vector corresponding to the k-th measurement of secondary $\mathbf{u}_i^j$. In addition, suppose that there are L_j measurements of the camera model P^j and M_i measurements of the primary $\mathbf{x}_i$ and let $\delta_k(P^j)$ and $\delta_k(\mathbf{x}_i)$ be the corresponding k-th error vectors. One seeks to minimize the value of

$$\sum_j \sum_{k=1}^{L_j} ||\delta_k(P^j)||^2 \quad + \quad \sum_i \sum_{k=1}^{M_i} ||\delta_k(\mathbf{x}_i)||^2 \quad + \quad \sum_i \sum_j \sum_{k=1}^{N_{ij}} ||\delta_k(\mathbf{u}_i^j)||^2 \quad (1)$$

where $||\cdot||$ represents the Euclidean norm of a vector. The minimum is sought by varying the parameters of primary features and cameras.

4. Sparse Techniques

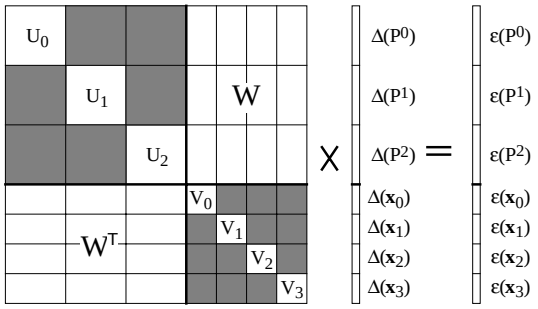
Let a set of primary features $\{\mathbf{x}_i\}$ be visible from several cameras, and let $\mathbf{u}_i^j$ be the error vector related to a measurement of the image of $\mathbf{x}_i$ as seen by the camera with parameters P^j. Given the error vectors $\mathbf{u}_j^i$, Carmen attempts to retrieve the parameters of the cameras and the geometric features, $\mathbf{x}_i$. This is done using a Levenberg-Marquardt (LM) algorithm. A key step in the LM algorithm is to compute the matrix of partial derivatives J of the error vectors (dependent parameters) with respect to the independent parameters. In the present case, the independent parameters fall into two classes, namely the camera parameters and the feature parameters $\mathbf{x}_i$. Altering the parameters of $\mathbf{x}_i$ will cause a change in the coordinates of only those error vectors $\mathbf{u}_i^j$ with the same index i as the primary feature. Similarly, altering the parameters of a camera P^j will lead to a change in only those $\mathbf{u}_i^j$ with the same index j as the camera. Consequently, the matrix J has a particular sparse structure as shown in the following diagram.



The diagram shows the case for three camera and four geometric features, but the general scheme is easily extended to any number of features and cameras. In the cases where some features are not visible in some views, then the corresponding rows are missing from the matrix. This makes very little difference to the following discussion.

The iterative step in a Newton parameter estimation algorithm is to solve the over-determined set of equations $J\Delta = \epsilon$, where J is the Jacobian, Δ is a vector of increments to be applied to the variable parameters, and ϵ is the residual error vector. The equation is to be solved in the least-squared sense. Using the method of normal equations ([10]) to solve these equations involves finding the solution of the square set of equations $J^\top J\Delta = J^\top \epsilon$. In the Levenberg-Marquardt algorithm the normal equation matrix $J^\top J$ is augmented by incrementing the diagonal elements before solution. Otherwise, the principal is the same.

Because of the structure of the matrix J, the normal equations (augmented or not) have a special block structure as follows



The important point about the matrix on the left is that the bottom right-hand block has a sparse structure, consisting of diagonal blocks V_j. One may use this fact to eliminate the top right hand block, and then solve for the $\Delta(P^j)$ all at once. Subsequently, the $\Delta(x_i)$ may be computed one feature at a time. Details of this procedure are omitted, but are given in [5]. Using this sparse technique gives an enormous saving in computation time compared with a straight-forward approach.

Figure 3: Linear-pushbroom image with spherical marker balls

In the above description, it was assumed that there was only one measurement of each secondary feature. A brief check reveals that this is not a necessary restriction, since the special form of the normal equations does not rely on this. It is important in this method that parameters of each primary geometric feature should be separate, and that there should be no explicit relations between these parameters. This is necessary to preserve the block-diagonal structure of the lower right-hand block of the normal equation matrix. It is not, however essential that the parameters of the cameras should be independent, since the upper left block does not need to be strictly block-diagonal. This observation allows us to share parameters between different cameras. For instance one may specify that all cameras have the same internal parameters. This sort of parameter sharing is allowed in Carmen.

5. Example

As an example of the use of Carmen consider Fig 3.

This is one of a sequence of X-ray images of a turbine blade taken from many angles. The purpose is to reconstruct features of the blade. The application is described in greated details in the papers [8, 7, 1]. The type of sensor used is of the linear-pushbroom type ([6]) which does a central projection in one axial direction and orthographic projection in the other.

In order to estimate the positions of the features (in this case, drill-holes) with maximum precision, it is necessary to estimate very accurately the orientation of the sensor for each view. To this end, a series of calibration markers are included in a fixture attached to the part. These calibration markers are spheres, and are visible as medium-grey circles (more precisely ellipses) in the image. The approximate location of these markers is determined through a priori knowledge of the approximate sensor orientation for each view. Carmen is then used to determine the precise sensor locations.

In order to extend the basic capability of Carmen for this application, it was necessary to write specific derived classes as follows.

1. A parametrization of the linear pushbroom model. This had 11 parameters.

Figure 4: Drill holes in turbine blade.

2. A camera-mapping specifying how the marker spheres are mapped into the image by a linear-pushbroom mapping.

3. A measurement mapping for comparing the predicted locations of the imaged marker features with the measured feature position.

Each of these tasks is a relatively minor programming task. Once this was done, least-squares estimation of all sensor locations could be carried out iteratively. After that, the positions of the features of interest (drill holes) could be accurately determined. Figure 4 shows the detected drill-holes in one of the images.

6. Conclusion

The camera modelling and feature reconstruction program, Carmen, described here gives the capability of minimizing squared sum error using a notion of true geometric distance, or other error model. It allows simultaneous use of several geometric feature types and camera models, and is arbitrarily extendible to new camera models or geometric features.

References

[1] Rajiv Gupta, Alison Noble, Richard Hartley, Joseph Mundy, and Andrea Schmitz. Camera calibration for 2.5-D X-ray metrology. In *Proc. International Conference on Image Processing ICIP-95, Volume III*, pages 37 – 40, 1995.

[2] R. Hartley, R. Gupta, and T. Chang. Stereo from uncalibrated cameras. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 761–764, 1992.

[3] R. I. Hartley. A linear method for reconstruction from lies and points. In *Proc. International Conference on Computer Vision*, pages 882 – 887, 1995.

[4] Richard Hartley. Photogrammetric techniques for panoramic cameras. In *Proc. SPIE Conf. on Integrating Photogrammetric Techniques with Scene Analysis and Machine Vision, Vol. 1994*, pages 127–139, April 1993.

[5] Richard I. Hartley. Euclidean reconstruction from uncalibrated views. In *Applications of Invariance in Computer Vision : Proc. of the Second Joint European - US Workshop, Ponta Delgada, Azores – LNCS-Series Vol. 825, Springer Verlag*, pages 237–256, October 1993.

[6] Richard I. Hartley and Rajiv Gupta. Linear pushbroom cameras. In *Computer Vision - ECCV '94, Volume I, LNCS-Series Vol. 800, Springer-Verlag*, pages 555–566, May 1994.

[7] A. Noble, R. Gupta, J. Mundy, A. Schmitz, R. Hartley, and W. Hoffman. CAD-based Inspection using X-ray Stereo. In *Proc. IEEE Robotics and Automation Conference*, pages 2361–2365, Japan, May 1995.

[8] Alison Noble, Richard Hartley, Joseph Mundy, and James Farley. X-ray metrology for quality assurance. In *Proc. IEEE Robotics and Automation Conference*, 1994.

[9] Jean Ponce, David H. Marimont, and Todd A. Cass. Analytical methods for uncalibrated stereo and motion reconstruction. In *Computer Vision - ECCV '94, Volume I, LNCS-Series Vol. 800, Springer-Verlag*, pages 463–470, 1994.

[10] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1988.

[11] Amnon Shashua. Projective depth: A geometric invariant for 3D reconstruction from two perspective/orthographic views and for visual recognition. In *Proc. International Conference on Computer Vision*, pages 583–590, 1993.

[12] C. C. Slama, editor. *Manual of Photogrammetry*. American Society of Photogrammetry, Falls Church, VA, fourth edition, 1980.

[13] I.E. Sutherland. Sketchpad: A man-machine graphical communications system. Technical Report 296, MIT Lincoln Laboratories, 1963. Also published by Garland Publishing Inc, New York, 1980.

[14] T. Viéville and Q.T. Luong. Motion of points and lines in the uncalibrated case. Report RR-2054, INRIA, 1993.