Scene point constraints in camera auto-calibration: an implementational perspective

D.Q. Huynh\textsuperscript{a,}\textsuperscript{*}, A. Heyden\textsuperscript{b}

\textsuperscript{a}School of Computer Science and Software Engineering, The University of Western Australia, Crawley 6009, Australia
\textsuperscript{b}School of Technology and Society, Malmö University, Sweden

Received 5 June 2004; received in revised form 11 April 2005; accepted 5 May 2005

Abstract

We present a scheme for incorporating scene constraints into the auto-calibration process for the structure and motion recovery problem. The steps covered by the scheme include projective factorization of the joint image measurement matrix, recovery of the absolute dual quadric, the upgrade from projective structure to its Euclidean counterpart, and incorporation of constraints from orthogonal scene planes into bundle adjustment. The focus of the paper is on the implementation details of all these steps and discussion of the various issues that arose. We have tested the scheme on both synthetic and real image data and found that it is more advantageous to incorporate into camera auto-calibration and bundle adjustment as many scene constraints as are available rather than performing auto-calibration and bundle adjustment alone.

\textcopyright 2005 Elsevier B.V. All rights reserved.

Keywords: Auto-calibration; Bundle adjustment; Scene constraints; Projective factorization; Euclidean reconstruction

1. Introduction

The structure and motion recovery problem concerns the recovery of both the scene structure and motion of the camera(s) from a set of images. Methods for solving this problem vary largely depending on the camera model chosen, these being, notably, the perspective [1,2] and the affine [3,4] camera models. The type of image features being used, e.g. image lines, points, or conics, on the other hand, determines the minimum number of images required for shape recovery [5–7]. If no special information about the camera or the scene is available then only a projective reconstruction of the scene can be obtained, cf. [1,8,2]. Since projective structures are not suitable for visualization, it is often desirable to carry out camera auto-calibration (see, e.g. [1,2]) to obtain a Euclidean reconstruction up to an unknown similarity transformation.

Traditionally, the Euclidean structure of a scene has been obtained via two different types of methods. The first is often referred to as stratification, since one starts with a projective reconstruction and then finds an affine ‘stratum’ and finally a Euclidean ‘stratum’ to give the desired reconstruction. In the upgrade from projective to Euclidean reconstruction, some scene constraints, e.g. some distance or angular measurements, cf. [9] may be incorporated. The second type of method, often referred to as auto-calibration, takes into account some a priori information about the intrinsic parameters, e.g. known skew and/or aspect ratio (see [10–13]) or constant intrinsic parameters (see [1]). The main focus of this latter type of method is on finding the intrinsic parameters, i.e. auto-calibrating the cameras.

Auto-calibration which involves the recovery of the absolute conic or the absolute dual quadric has been attempted by various researchers. The earliest work on recovery of the absolute conic is the algorithm by Faugeras et al. [14], where one camera is involved and its intrinsic parameters are assumed fixed. By assuming also fixed camera intrinsic parameters, Heyden and Åström [15] later retrieved Euclidean structures via the computation of the absolute dual quadric. With the same assumption, Triggs [11] proposed the use of quasi-linear constraints for
the absolute dual quadric. Pollefeys et al. [16], on the other hand, incorporated the so-called modulus constraint into the stratification approach to upgrade projective structures to affine and finally recover the absolute conic and upgrade the structures to Euclidean. In their other work [13], they first assumed the principal points in all images vanish to linearize the equations for computing the absolute dual quadric, a method that we adopt in this paper. They then fine tuned the absolute dual quadric via a nonlinear optimization process in which the intrinsic parameters of the images are allowed to vary independently. The possibility of recovering camera intrinsic parameters under various conditions (e.g. a certain parameter is known or when they are known to be fixed) was proven by Heyden and Åström [17]. Although Pollefeys et al. [13] have indicated that scene constraints can be incorporated into auto-calibration, they did not carry out further research along this line. Liebowitz and Zisserman [18] later reported their use of the image projections of parallel and orthogonal scene lines to estimate the vanishing points and as constraints for estimating the absolute conic. However, their method requires the computation of the fundamental matrix, and that limits it to working with two images. Now it is recognized that scene constraints can be easily incorporated into various computer vision problems. Triggs et al. [19] report a detailed literature survey on bundle adjustment and the incorporation of scene constraints into the process. Gong and Xu [20,21] have also attempted to incorporate constraints into surface recovery problems for calibrated cameras.

This paper is an extension of our previous work reported in [22,23]. As before, we will use the natural camera model, i.e. a model that has zero skew and unit aspect ratio, and orthogonal scene planes as constraints for the estimation of the absolute dual quadric and bundle adjustment. The use of the natural camera model is justified by the high quality digital and video cameras available today. Even if the skew is non-zero and the aspect ratio of a camera is not unity, these two entities are known to be invariant under change of focus and so they can be pre-calibrated and treated as constant. The use of orthogonal scene planes is justified by the presence of many such planes in environments that contain man-made objects, e.g. indoor scenes, buildings. The scheme of incorporating scene constraints into auto-calibration covers a number of steps, namely, projective structure retrieval, absolute dual quadric estimation, Euclidean structure upgrade, and bundle adjustment. We will discuss each step in detail in the rest of the paper.

The paper is organized as follows. Section 2 recapitulates the background on projective reconstruction and factorization. In this section, the iterative projective factorization will be reviewed and the gauge freedom that we found essential to the subsequent auto-calibration step will be discussed. Section 3 then covers the linear estimation of the absolute dual quadric, the incorporation of orthogonal scene plane constraints, and the upgrade of projective structures to Euclidean. Section 4 discusses bundle adjustment without and with scene constraints imposed and the various issues we have studied during our investigation. Section 5 reports our experiments testing all the steps involved with both simulated and real video images. Section 6 lists other scene constraints that can be incorporated and a few issues on parameter setting. Finally, Section 7 concludes the paper.

2. Projective reconstruction and factorization

2.1. Background on projective reconstruction

Given a scene point \( X' = [x' \ y' \ z' \ 1]^T \), its projection \( u' = [u' \ v' \ 1]^T \) onto an image plane is governed by:

\[
\begin{pmatrix}
    f & 0 & u_0 \\
    0 & f & v_0 \\
    0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
    R \\
    -Rt
\end{pmatrix}
X',
\]

i.e. \( \xi' u' = K \begin{pmatrix} R & -Rt \end{pmatrix} X' \),

where the superscript \( j \) denotes the \( j \)th scene (or image) point from a list of scene (or image) points and \( \xi' \) an unknown scalar known as the relative depth (also referred to as the projective depth). The first matrix is the camera matrix \( K \) which embodies the unknown focal length \( f \) and principal point \((u_0, v_0)\) of the camera. The second matrix is the motion matrix which contains the unknown rotation \( R \) and translation \( t \) of the camera to the scene point. The special form of \( K \) arises from the use of the natural camera model.

When none of these parameters are known a priori, (1) is often put in the compact form \( \xi' u' = P X' \), where \( P \) is a 3\( \times \)4 projection matrix. With the availability of \( m \) images and \( n \) scene points, the \( 3m \times 4 \) joint projection matrix \( P \), the \( 3n \times n \) joint image measurement matrix \( X \), and the \( 4 \times n \) joint shape matrix \( U \) are related by

\[
\begin{bmatrix}
    \xi_1 u_1^1 & \cdots & \xi_n u_n^1 \\
    \vdots & \ddots & \vdots \\
    \xi_1 u_1^m & \cdots & \xi_n u_n^m
\end{bmatrix} =
\begin{bmatrix}
    P_1 \\
    \vdots \\
    P_m
\end{bmatrix}
\begin{bmatrix}
    X^1 \\
    \vdots \\
    X^n
\end{bmatrix}
\]

\( \Leftrightarrow U = PX \),

where the subscript \( i \) denotes the \( i \)th camera. Without any knowledge of the intrinsic and extrinsic parameters of each image, the \( P \) and \( X \) matrices can only be estimated from \( U \) up to a projective transformation.

Tomasi and Kanade [3] pioneered the use of the factorization method to retrieve the joint projection matrix \( P \) and joint shape matrix \( X \) from the joint image measurement matrix \( U \). Since their method is based on the affine camera model, all the \( \xi' \) terms can be set to unity. The choice of the centroid of the image point cluster as the origin of the coordinate system in each image allows one to reduce the dimensions of the \( P \) and \( X \) matrices to \( 2m \times 3 \) and \( 3 \times n \),
respectively. For the perspective camera model, Sparr [24] and Heyden [25] proposed using the subspace method to iteratively refine all the $\xi_i^j$'s. At about the same time, Sturm and Triggs [26] (see also [27]) proposed a different scheme that requires the estimation of the fundamental matrices and the epipoles. Other iterative approaches were recently suggested by Oliensis [28] and Mahamud et al. [29,30] and proofs for the convergence of the iterative approaches were given in their work. In [30], in particular, the estimation of projective depths was eliminated.

2.2. Iterative projective factorization

The factorization method we use to extract the shape and motion matrices is that proposed by Sparr [24], Heyden [25, 31], and Berthilsson et al. [32]. The method is based on the study of the 4-D subspace spanned by the row vectors of the image measurement matrix. Below we summarize the ideas presented in these papers.

Consider the projection of the $n$ scene points onto the $i$th image. Eq. (1) can be written as

$$[u_i^1, u_i^2, \ldots, u_i^n] = P_i \begin{bmatrix} x_i^1 \\ \xi_i^1 \\ \vdots \\ x_i^n \\ \xi_i^n \end{bmatrix} = P_i X_i = P_i \Lambda_i,$$

(3)

where the relative depths, $\xi_i$, embodied in the diagonal matrix $\Lambda_i$, must be estimated in the factorization. In [24,31], it was shown that the 3D-space, $D_i$, spanned by the row vectors in $U_i \Lambda_i$, is a subspace of the 4D-space, $\mathbb{D}$, spanned by the row vectors of $X_i$. By putting all images together, it was then proven that $\sum_{i=1}^{m} \Lambda_i / D_i = D$. Thus, the procedure to follow is to iteratively estimate the relative depths and the shape matrix $X_i$ in an alternating fashion while constraining the subspace $\sum_{i=1}^{m} \Lambda_i / D_i$ to be 4-dimensional, i.e.

- Compute the shape matrix $X_i$ using the current estimates of $\Lambda_i$, for all $i = 1, \ldots, m$.
- Use the $X_i$ matrix obtained from Step 1 to re-estimate the relative depths in each $\Lambda_i$ matrix for the next iteration.

Initially, all the relative depths $\xi_i^j$ were set to unity. Together with the shape matrix $X_i$, they were then refined in each iteration.

We found the iterative method very robust and convergence was always guaranteed after 10–20 iterations (see [28–30] for proofs). In each iteration, an SVD operation is required to compute the $X_i$ matrix and then an additional SVD for the computation of each $\Lambda_i$ matrix. The overall runtime will, of course, depend on both the number of images and number of scene points. If speed is a concern then a recursive approach mentioned in [32,31] can be adopted.

We also examined the reprojection errors of the $P$ and $X_i$ matrices returned by the iterative factorization method and found them very small (less in magnitude than $10^{-1}$ pixels in small noise conditions, i.e. when $\sigma=0.1$ as discussed in Section 5). In addition, we also assessed the kinetic depths (introduced in [33]) and found them very close to the true values. The kinetic depth, denoted by $d_i$, is related to the relative depth $\xi_i$ by the formula $d_i = \xi_i / \xi_i^j$. We note that it is necessary to assess the accuracy of $d_i$ rather than $\xi_i$ because the latter is dependent on the ambiguous scale involved in each reconstructed scene point but the former eliminates such ambiguity.

As the method is based on the search of the 4D-subspace of $(\mathbb{R}^n)$ spanned by the row vectors of $U_i \Lambda_i$, for $i = 1, \ldots, m$, it is independent of the underlying coordinate system used in each image. Furthermore, since the row vectors of the shape matrix, $X_i$, are recovered as the basis of the 4D-subspace of $(\mathbb{R}^n)$, the final $X$ matrix returned by the method has orthonormal row vectors. Once the $X$ matrix is estimated, the recovery of the joint projection matrix $P$ involves solving a set of linear equations.

2.3. Gauge freedom

It is well known that the global coordinate system can be arbitrarily set for the shape and motion recovery problem from images. Such freedom of fixing the coordinate system is known as gauge freedom and studies on this topic have been reported in the literature (e.g. see [19,34–37]). Although, theoretically, the underlying geometry is invariant for different gauges, we found that undesirable numerical problems can often result from a bad choice of coordinate system. As the projective factorization process is iterative and involves complicated operations such as SVD and the computation of the orthonormal basis of the 4D subspace (via the Matlab function orth), it is not straightforward to analyse the gauge transformation involved in the factorization process. We have, however, tested the effect of changing the origin of the global coordinate system by post-multiplying the $P$ matrix returned by the projective factorization method by various arbitrary non-singular $4 \times 4$ matrices. In particular, we studied the consequence of setting the global coordinate system at the first camera’s optical centre. The results were that in most cases the subsequent auto-calibration process either failed (e.g. the values of focal lengths became complex numbers) or the estimated focal lengths deviated from the true values by a factor of 10 or more.

In the following step of estimating the absolute dual quadric, we therefore tried to fix the gauge returned by the iterative projective factorization step.
3. Linear estimation of the absolute dual quadric

The structure contained in the shape matrix $X$ is projective only, since for any joint projection matrix $\mathbf{P}$ and joint shape matrix $\mathbf{X}$ that satisfy (2), $\mathbf{PA}$ and $A^{-1}\mathbf{X}$ also form a solution, where $A$ is any non-singular $4 \times 4$ matrix.

To upgrade the projective structure to Euclidean, we must seek a matrix $A$ for an appropriate change of coordinates in the estimated scene points. That is, we estimate $A$ such that

$$P_iA \sim K_i[\mathbf{R}_i, \mathbf{-R}_i \mathbf{t}_i], \quad \text{for } i = 1, \ldots, m,$$

where $\sim$ denotes equality up to an unknown scale. Let $A$ be of the form

$$\begin{bmatrix} \tilde{A} & 0 \\ \mathbf{a}^T & s \end{bmatrix},$$

(5)

where $\tilde{A} \in \mathbb{R}^{3 \times 3}$, $\mathbf{a} \in (\mathbb{R}^3$, and $s$ is a non-zero scalar, which is often set to unity. This special form of $A$ retains the projection of the origin of the global coordinate system, i.e.

$$P_iA = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & s \end{bmatrix} \sim P_i \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}.$$  

Let $\tilde{A}$ be the $4 \times 3$ sub-matrix that comprises the first three columns of $A$. Then it follows that

$$P_i\tilde{A}\tilde{A}^TP_i \sim K_iR_iR_i^TK_i^T,$$

$$\Rightarrow P_i\begin{bmatrix} \tilde{A} \\ \mathbf{a}^T \end{bmatrix} \begin{bmatrix} \tilde{A}^T & \mathbf{a} \end{bmatrix}P_i \sim K_iR_iR_i^TK_i^T,$$

(6)

$$\Rightarrow P_iQ^*_mP_i^T \sim K_iK_i^T, \quad \text{for } i = 1, \ldots, m.$$  

The $4 \times 4$ matrix, denoted by $Q^*_m$ in (6), is the absolute dual quadric that embodies both the plane at infinity (vector $\mathbf{a}$) for affine reconstruction and the DIAC (dual image of the absolute conic) for Euclidean reconstruction.

Geometrically $Q^*_m$ consists of planes tangential to the absolute conic and the inclusion angle, $\theta$, between any two planes, whose coordinates are $\mathbf{n}$ and $\mathbf{m}$ (both in ($\mathbb{R}^3$)), can be computed via

$$\cos(\theta) = \frac{\mathbf{n}^T Q^*_m \mathbf{m}}{\sqrt{\mathbf{n}^T Q^*_m \mathbf{n} \mathbf{m}^T Q^*_m \mathbf{m}}}.$$  

(7)

A common approach to obtain an initial estimate of the elements of $Q^*_m$ is to set the origin of the image coordinate system of each image at the centre of the image buffer so that the principal point, $(u_0, v_0)$ of each image can be approximated to be $(0, 0)$. This effectively diagonalizes each of the camera matrices, $K_i$, for $i = 1, \ldots, m$, and thus allows the elements of $Q^*_m$ to be computed from a set of linear equations. With the camera matrices $K_i$ in diagonal form, 4 equations are provided by each image, namely:

$$\begin{align*}
(P_iQ^*_mP_i^T)_{(1,1)} &= (P_iQ^*_mP_i^T)_{(2,2)}, \\
(P_iQ^*_mP_i^T)_{(1,2)} &= 0, \quad (P_iQ^*_mP_i^T)_{(1,3)} = 0, \quad (P_iQ^*_mP_i^T)_{(2,3)} = 0,
\end{align*}$$

(8)

where $(\cdot)_{ij}$ denotes the $(i, j)$-element of the matrix in consideration.

Since the absolute dual quadric, $Q^*_m$, has rank equal to 3 and is defined only up to a scale, it really has only 8 degrees of freedom. To obtain an initial estimate of $Q^*_m$ linearly (without using the condition $\det(Q^*_m) = 0$), 3 images are required, which provide a total of 12 equations for the unknown elements of $Q^*_m$.

3.1. Incorporation of orthogonal scene plane constraints

If image feature points from two orthogonal planes in the scene are identified then an additional linear constraint on $Q^*_m$ can be imposed. That is, by setting $\theta = 90^\circ$ in (7) and temporarily ignoring the normalization factor in the denominator, we obtain

$$\mathbf{n}^T Q^*_m \mathbf{m} = 0.$$  

(9)

As mentioned in Section 3, the linear estimation of the absolute dual quadric requires the projection matrices of 3 images. If the condition $\det(Q^*_m) = 0$ is not considered in the estimation then $Q^*_m$ has 9 degrees of freedom. The provision of one scene constraint, as given in (9), will reduce the number of images required to 2.

In our implementation, we incorporated as many images and scene constraints as are available for estimating $Q^*_m$, so as to avoid any degenerate configuration or critical camera motion that might be present in some subsets of the images taken. For instance, if there are three mutual orthogonal scene planes whose coordinates are $\{\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3\}$ then these scene constraints, one for each pair of planes, would be used, although not all of these constraints are independent.

Whether scene constraints are or are not incorporated into the system, the elements of the $Q^*_m$ matrix can be computed either via SVD or eigen-decomposition. It is practical to examine the magnitude of the smallest singular (or eigen) value and the ratio between the two smallest singular (or eigen) values. These two values give some indication about the accuracy of the estimated $Q^*_m$ and whether camera parameters can be retrieved later.

After obtaining the initial estimate of $Q^*_m$, one can enforce constraints from the specific camera model (e.g. zero skew, unit aspect ratio) used and formulate a non-linear system of equations to further refine the estimate of $Q^*_m$. This can be done iteratively by
where \( \| \cdot \|_F \) denotes the Frobenius norm of the matrix concerned. However, we found that the formulation becomes rather cumbersome when the exact form of the \( \hat{A} \) matrix in (5) that comprise \( Q^*_n \) is not known. For instance, if the global coordinate system is set at the optical centre of the first image and \( P_1 = [I \ 0] \) then \( \hat{A} = K_1 \). Since our \( P_1 \) matrix does not take the above form and since the iterative refinement of \( Q^*_n \) in (10) involves minimizing the algebraic errors only (see [38], Section 18.3.3), we have eliminated this step from our current work (c.f. our previous work [22,23]). Instead, we shift our focus to the final bundle adjustment step with scene constraints incorporated.

### 3.2. Initial linear Euclidean upgrade

Not only does the step described above give an estimate of the \( Q^*_n \) matrix, it also allows the projective structure of the scene to be upgraded to Euclidean.

From (6), we have

\[
\hat{A}\hat{A}^T \sim Q^*_n.
\]

Let \( USUT = Q^*_n \) be the singular value decomposition of \( Q^*_n \). We can then compute \( \hat{A} \) by letting \( \hat{A} = U_3 S_3^{1/2} \), where \( U_3 \) and \( S_3 \) are the matrices containing the first 3 columns of \( U \) and \( S \). The upgrade of the projective structure, \( X \), to its Euclidean counterpart, \( X_e \), is simply:

\[
X_e = X\hat{A}^{-1}.
\]

For the intrinsic parameters, \( \{f_i, u_{0,i}, v_{0,i} \mid i=1,\ldots,m\} \), we chose to recover them by applying the Cholesky decomposition on the \( P_i Q^*_n P_i^T \) matrix. As mentioned in [38] (Page 453), this step will fail if \( P_i Q^*_n P_i^T \) is not positive definite. However, we have never encountered this problem in our experiments using the projection matrices \( P_i \) estimated by the iterative factorization method. Only when changing the global coordinate system would this problem sometimes occur (see also Section 6). Examination of the singular (or eigen) values (see the subsection above) involved in the computation of \( Q^*_n \) also serves as an indicator whether auto-calibration would succeed at this stage.

For the retrieval of the extrinsic parameters \( R_i \) and \( t_i \), for each \( i=1,\ldots,m \), the process is straightforward. By enforcing the condition that each \( R_i \) is a rotation matrix, i.e. \( R_i R_i^T = R_i^T R_i = I \) and \( \det(R_i) = 1 \), we can find the optimal rotation matrix \( R_i \) and translation vector \( t_i \) that satisfy

\[
[R_i \ t_i] \sim K_i^{-1} P_i, \quad \text{for } i = 1, \ldots, m.
\]

### 4. Bundle adjustment

The parameters obtained from the auto-calibration and Euclidean upgrade processes above are not optimal and an iterative refinement on the scene point coordinates and intrinsic and extrinsic parameters is required. This process of iteratively refining all these parameters simultaneously with the objective of minimizing the reprojection errors of feature points is known as bundle adjustment. In this section, we will discuss in detail how scene constraints are incorporated in our bundle adjustment process and various issues that we studied.

#### 4.1. Bundle adjustment without scene constraints

For our camera model, each \( i \)th image has 9 unknowns to be recovered, namely 3 intrinsic parameters, \( f_i, u_{0,i}, v_{0,i} \), and the 6 extrinsic parameters, which comprise the 3 rotation angles, \( \alpha_i, \beta_i, \gamma_i \), embodied in the rotation matrix \( R_i \), and the 3 components, \( t_{x,i}, t_{y,i}, t_{z,i} \), of the translation vector \( t_i \). Switching to working in inhomogeneous coordinates, each \( j \)th scene point has three unknowns: \( x^j, y^j, z^j \). Thus, the total number of parameters to be refined is \( 9m + 3n \). As each image point provides two equations, the total number of equations available is \( 2mn \) for \( n \) points in \( m \) images.

Let \( \theta \in \mathbb{R}^{9m+3n} \) be the long vector formed by the concatenation of all these parameters. Let \( \hat{u}_i = (\hat{u}_i^u, \hat{v}_i^v)^T \) denote the coordinates of the detected image points and \( \hat{u}_i = (\hat{u}_i^u, \hat{v}_i^v)^T \) the reprojected image coordinates of point \( j \) in image \( i \). So, \( \hat{u}_i \) satisfies the 3D to 2D projection given in (1) and is a function, denoted by \( f \) in (11) below, of the parameter \( \theta \). Let \( u \) and \( \hat{u} \) denote, respectively, the concatenation of all the \( u_i \) and \( \hat{u}_i \) entities. Let \( C \) denote the covariance matrix of the noisy image point coordinates \( u \).

Without the use of scene constraints, bundle adjustment is an unconstrained optimization problem to minimize the residual (i.e. the reprojection errors) in the objective function:

\[
\min_{\theta} \frac{1}{2} \| f(\theta)^T C^{-1} f(\theta) \| = \min_{\theta} (u - \hat{u})^T C^{-1} (u - \hat{u}). \tag{11}
\]

Since there are \( 2mn \) terms in \( f \), the Jacobian matrix, which stores the derivative of each of these terms with respect to \( \theta \), is of dimensions \( 2mn \times (9m+3n) \). Although this matrix is relatively large in real applications, it is sparse in nature, making it worthwhile to use methods specifically designed for the least-squares problem to solve (11). For instance, as shown in Fig. 1, when \( m=2 \) and \( n=10 \), the Jacobian matrix is \( 40 \times 48 \) in dimensions, providing 40 equations from the \( u \)- and \( v \)-components of the 20 image coordinates in two images. The first 18 columns then correspond to the derivatives with respect to the parameters \( f_i, u_{0,i}, v_{0,i}, \alpha_i, \beta_i, \gamma_i, t_{x,i}, t_{y,i}, t_{z,i} \) for \( i=1, 2 \). The remaining 30 columns correspond to the derivatives with respect to the three components of the 10 scene points.
where the unknown Lagrange multiplier vector $\lambda \in \mathbb{R}^n$ must be estimated also.

We investigated two different methods for estimating the unknown $\theta$ in (14). Each of these methods is described below.

### 4.2.1. Sequential quadratic programming (SQP)

The approximation of $f$ and $c$ using Taylor expansions permits the SQP method to compute the Lagrange multiplier $\lambda$ and the estimate of $\theta$ by means of the solution of the equation

$$
\begin{bmatrix}
H_k & a_k^T \\
a_k & 0
\end{bmatrix} \begin{bmatrix}
\delta \theta_k \\
\lambda_k
\end{bmatrix} = - \begin{bmatrix}
g_k^T \\
c_k
\end{bmatrix},
$$

(15)

where subscript $k$ denotes the iteration number; $g_k \in \mathbb{R}^1 \times (9m+3n)$ and $H_k \in \mathbb{R}^{(9m+3n) \times (9m+3n)}$ are, respectively, the gradient and Hessian estimates of $f$ with respect to $\theta$ evaluated at $\theta_k$; $a_k \in \mathbb{R}^{(9m+3n)}$ denotes the gradient of $c$ with respect to $\theta$ evaluated also at $\theta_k$. Thus, given that the estimate $\theta_k$ at the $k$th iteration is known, one can use (15) to solve for $\delta \theta_k$, and then compute $\theta_{k+1}$ for the next iteration using the update formula $\theta_{k+1} = \theta_k + \delta \theta_k$.

SQP requires knowledge of the inverse of the Hessian matrix of $f$. For bundle adjustments, the parameter space is often very large and, even if the Hessian matrix is known to be sparse, it is often impossible to compute this matrix analytically. We therefore have investigated several methods for estimating it and have studied their performance:

- **Method 1:** at iteration $k=0$, $H_k$ was initialized using finite differences and the Broyden-Fletcher-Goldfarb-Shanno (BFGS) update (see Appendix) was used to compute $H_{k+1}$ whose inverse was then computed for the next iteration.
- **Method 2:** at iteration $k=0$, $H_k$ was initialized the same way as in method 1 and the inverse of $H_k$ was computed. The Davidon-Fletcher-Powell (DFP) update (see Appendix) was then used to compute the inverse of $H_{k+1}$ directly.
- **Method 3:** at each iteration $k \geq 0$, $H_k$ was approximated as $J_k^T C^{-1} J_k$ and its inverse was computed. This update formula can be easily derived by differentiating (11) twice and discarding the term involving the derivative of $J$. This formula has been used in the popular Gauss-Newton method in the literature on optimization.

In all of the above methods, if $H_k$ was singular then the pseudo-inverse of the matrix was computed instead. Unfortunately, we found that none of these methods gave satisfactory results for SQP in our simulated tests. Although SQP has been shown to be successful for quadratic surface reconstruction [20] and structure recovery from single calibrated views [21], the $\theta$ vector in our simulated tests were of dimensions close to 200 (see Section 5) and were
much higher for real video data. These space dimensions may be too large for SQP to show satisfactory convergence using any inexact estimates of the Hessian of the objective function. Other methods for estimating Hessian matrices may improve the performance of SQP and further investigation would be required.

4.2.1. The penalty function method.

The penalty function method uses a standard idea to turn the original constrained optimization problem into a sequence of unconstrained ones. In each iteration, the unconstrained subproblem can be solved using any of the standard methods, such as Levenberg-Marquardt or Gauss-Newton method, and the iteration terminates when the solution satisfies the optimality conditions. The penalty function method involves constructing the quadratic penalty function:

\[ P_Q(\theta, \rho) = \min_\theta \frac{1}{2} f(\theta)^T C^{-1} f(\theta) + \frac{\rho}{2} c(\theta)^T c(\theta), \tag{16} \]

where \( \rho \) is a non-negative scalar known as the penalty parameter and \( c(\theta)^T c(\theta) \) is called the penalty term.

It has been proven that, under mild conditions, as \( \rho \to \infty \), the estimate of \( \theta \) approaches the optimal solution \( \theta^* \). However, one cannot achieve this optimal \( \theta^* \) by setting \( \rho \) to an infinitely large value and then solve (16) as a single unconstrained optimization problem. The reason is that if \( \rho \) is set to a large value initially, the condition number of the Hessian of \( P_Q(\theta, \rho) \) would be large, making it difficult for any robust unconstrained algorithms to attain \( \theta^* \). Thus, the original constrained optimization problem must be formulated as a sequence of unconstrained optimization subproblems, each of which has a fixed value of \( \rho \). The value of \( \rho \) should be set to a small value (e.g., \( \rho = 0.5 \)) initially and then increased gradually over the sequence (e.g., increased by a factor of 5 to 10). The iterations terminate when acceptable convergence criteria for the solution of the original problem are satisfied.

In comparison with the SQP, the penalty function method performed much better in our experiments and it always gave significant improvement to bundle adjustment without scene constraints. Another advantage of using the penalty function method rather than SQP is that the Hessian of \( f \) is not required. The trade-off of this method is the computation time—it requires solving a sequence of unconstrained minimization subproblems, each of which is equivalent to performing a bundle adjustment without scene constraints incorporated.

4.3. A note on the constraints from orthogonal scene planes

This subsection discusses how we formulate the scene constraints arising from a set of orthogonal scene planes. We introduce the discussion on a pair of orthogonal scene planes. The idea generalizes to two or more pairs of such scene planes.

Let \( \{X^1, \ldots, X^n\} \) and \( \{Y^1, \ldots, Y^n\} \) be two sets of scene points lying on two planes that are orthogonal to each other. We introduce the ‘nuisance’ parameters \( n_x, n_y \in \mathbb{R}^d \) that denote the coordinates of these two planes. Then \( n_x \) and \( n_y \), respectively, satisfy (17) and (18) below.

\[ \psi_x(n_x, \{X^1, \ldots, X^n\}) \triangleq An_x = 0, \tag{17} \]

\[ \psi_y(n_y, \{Y^1, \ldots, Y^n\}) \triangleq Bn_y = 0, \tag{18} \]

where \( A = [X^1 \ldots X^n]^T \in \mathbb{R}^{n \times 3} \) and \( B = [Y^1 \ldots Y^n]^T \in \mathbb{R}^{n \times 3} \).

The constraint arising from this pair of orthogonal scene planes can be formulated in terms of the two nuisance parameters as follows:

\[ c(n_x, n_y) \triangleq (n_x^T n_x) \sqrt{(n_x^T n_x)(n_y^T n_y)} = 0, \tag{19} \]

where \( I = \text{diag}(1, 1, 1, 1) \). For bundle adjustment that employs the Levenberg-Marquardt algorithm, we need only compute the gradient of \( c \) with respect to the parameters to be refined, in this case, \( \{X^1, \ldots, X^n\} \) and \( \{Y^1, \ldots, Y^n\} \).

We focus on a particular scene point \( X^l = (X^l_1, X^l_2, X^l_3, 1)^T \). Differentiating (17) with respect to \( X^l \), for \( l = 1, \ldots, 3 \), gives

\[ \frac{\partial \psi_x}{\partial X^l} + \frac{\partial c}{\partial n_x} \frac{\partial n_x}{\partial X^l} = 0 \Rightarrow \hat{\lambda}^l n_x + A \frac{\partial n_x}{\partial X^l} = 0, \tag{20} \]

where \( \hat{\lambda}^l \) denotes a zero matrix having the same dimension as \( A \) and its \( (j, l) \)-element has the value 1.

Finally, the derivative of \( c \) with respect to \( X^l \) can be estimated using the chain rule:

\[ \frac{\partial c}{\partial X^l} = \frac{\partial c}{\partial n_x} \frac{\partial n_x}{\partial X^l} \tag{21} \]

where \( \frac{\partial c}{\partial n_x} \) and \( \frac{\partial c}{\partial n_y} \) can be computed from (20) and (19) respectively. The derivative of \( c \) with respect to \( Y^l \) can be similarly derived.

In our implementation, the nuisance parameters \( n_x \) and \( n_y \) were recomputed in each iteration using the latest estimates of coordinates of the scene points.

4.4. A note on the implementation

We implemented the bundle adjustment step for both the cases where scene constraints are absent and present. In the former case, bundle adjustment is just a nonlinear minimization process that minimizes the projection errors. In the latter case, if two sets of scene points fall onto a pair of orthogonal scene planes then a scene constraint of the form given in (19) can be incorporated into (16) for the penalty function method. Note that as the scene constraint (see (19)) is formulated in terms of two nuisance parameters
(n_i and n_j) which are computed via (17) and (18), the gradient vector of the scene constraint with respect to the parameters stored in ∂θ to be optimized is computed via the chain rule (see (20)). For both cases, the criterion used for terminating the iterations is when ∥Δθ∥<ε for some predefined ε.

5. Results

We tested the entire procedure described in the previous three sections on synthetic image data as well as real image data.

For all the experiments on both the synthetic and real image data, the iterative projective factorization method described in Section 2.2 was first applied. Two independent processes were then carried out and their outputs were compared:

- Process (i): The absolute dual quadric was estimated using the linear method with no scene constraints added to the linear system. After the reconstruction was upgraded to Euclidean, the bundle adjustment step described in Section 4.1 was carried out.
- Process (ii): The absolute dual quadric was estimated using the linear method and scene constraints were added to the linear system. After the reconstruction was upgraded to Euclidean, the bundle adjustment step using the penalty function method described in Section 4.2 was carried out. The penalty parameter, ρ, was set to 0.5 initially. Its value was then increased by a factor of 10 in each of the 8 iterations.

In all the experiments, the covariance matrix C was set to the identity matrix due to the lack of knowledge of the exact distribution of the noise on the image point coordinates.

For the synthetic image data, we simulated the motion of a camera viewing 48 scene points, lying on three mutually orthogonal planes in the scene with 16 points on each plane. The focal length of the camera in each image was varied between 900 to 1500 pixels and the 3 focal lengths (in pixels), principal points (in pixels), and orthogonal angles (in degrees) in 100 runs of the simulation. The RMS errors of the initial parameter estimates, the final parameter estimates with and without bundle adjustment are labelled as squares (□), circles (○), and stars (*), respectively. It is clear from the figure that the RMS errors of all these entities are smaller after the refinement of bundle adjustment. It is also interesting to see that, even under noisy free conditions (when σ=0.0), the focal length estimates have a RMS error of 20–30 pixels (see Fig. 3(d)). This is related to the assumption about the position of the principal points (u_0,v_0) = (0, 0) when they were, in fact, ±(25, 25) pixels from the origin, as the principal point estimates (see Fig. 3(c)) also show a non-zero RMS error under noise free conditions. From the plots in Fig. 3(b) and (e), it appears that the errors of the principal point estimates do not have much effect on the estimates of reconstruction and orthogonal angle.

As the Euclidean structure from auto-calibration is known only up to a similarity transformation, we would like to briefly mention how our reconstruction was evaluated against the known 3D ground truth. Suppose that {X_i|i=1,...,n} and {X̂_i|i=1,...,n} are the sets of estimated and true 3D point coordinates. We first computed the centroids of the two sets of points and temporarily changed the gauge by aligning the two centroids, giving two new sets of points. If the reconstruction was good then the two sets of points should then have differed only by a νR transformation, where ν∈ℝ and R is a rotation matrix. The entities ν and R could be computed using least-squares and the RMS error was then obtained as the average distance between corresponding points in the two sets after applying the νR transformation to one of the sets.

Fig. 2. An example of the five images in the simulation. All images are of size 800×800 pixels.
Fig. 3. (a)–(e) The RMS errors in 100 runs of simulation when different levels of Gaussian noise, $\sigma = 0.0, \ldots, 0.4$ pixels, were added to the image point coordinates.

Fig. 4. Five images of an indoor scene. All images are of dimensions 768 × 576 pixels. Superimposed onto each image are the 35 manually detected image feature points (marked as black dots).
The results of two real experiments are reported here. In the first experiment, five images of dimensions 768 × 576 pixels of an indoor scene (see Fig. 4) were captured. The origin of the image coordinate system in each image was set at the centre of the image buffer, (384, 288). A total of 35 key image feature points were identified and the correspondences were manually established for auto-calibration. A few pairs of orthogonal scene planes were present in the scene and two of them were used in this experiment, giving 2 scene constraint equations. The first selected pair of orthogonal scene planes was formed by image feature points on the two pieces of A4-sized paper on the door and the wall. Each scene plane had 9 points. The second pair of orthogonal scene planes was formed by image feature points on the top and front faces of the two boxes. Each plane had 6 points.

Using the coordinates of all the image feature points, the same procedure as for the synthetic data was carried out. The estimated values of focal lengths and principal points of the five images are given in Table 1. Unfortunately, there is no ground truth to assess the accuracy of these estimates. From the table, it shows that the principal point of the camera varied quite randomly around the centre of the image buffer from one image to the next and, even when scene constraints were imposed, the camera’s principal point could be as far as 80 pixels from where we set it initially.

For process (i), the RMS reprojection errors before and after bundle adjustment were 0.92 and 0.88 pixels; the two inclusion angles of the pairs of orthogonal planes before and after bundle adjustment were estimated to be \{88.67°, 87.87°\} and \{88.13°, 87.82°\}. For process (ii), these figures were 0.92 and 0.90 pixels and \{89.07°, 89.22°\} and \{90.00°, 90.00°\}, respectively. The results show that it is more advantageous to impose scene constraints, if they are available, in that known orthogonal angles in the scene can be maintained while the reprojection errors are also smaller. As expected, one may notice that the RMS reprojection errors for process (ii) after bundle adjustment became slightly larger than those before bundle adjustment in order to satisfy the scene constraint. Also noticeable is that in process (i), because the objective of bundle adjustment is to minimize reprojection errors and because of the absence of scene constraints, the inclusion angles of the two pairs of planes deviate more from 90° after bundle adjustment.

Table 1
The estimated focal lengths and principal points for the input indoor scene images after bundle adjustment. The origin of the image coordinate system was set at the centre of the image buffer, so the principal point coordinates should all be small.

<table>
<thead>
<tr>
<th>Image</th>
<th>Process (i): no scene constraints</th>
<th>Process (ii): with scene constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Focal length</td>
<td>Principal point</td>
</tr>
<tr>
<td>1</td>
<td>1562.14</td>
<td>(−92.41, −32.39)</td>
</tr>
<tr>
<td>2</td>
<td>1736.65</td>
<td>(−100.76, −145.31)</td>
</tr>
<tr>
<td>3</td>
<td>1619.29</td>
<td>(−135.61, −138.91)</td>
</tr>
<tr>
<td>4</td>
<td>1091.97</td>
<td>(−59.26, −13.08)</td>
</tr>
<tr>
<td>5</td>
<td>1027.74</td>
<td>(−32.83, −4.85)</td>
</tr>
</tbody>
</table>

Fig. 5. Euclidean reconstruction of the indoor scene. The recovered optical centres are labelled as black stars in the frontal view.
The Euclidean reconstruction from process (ii) is shown in Fig. 5. The Euclidean reconstruction from process (i) is visually not differentiable from that from process (ii) and is therefore not shown. From the top view (Fig. 5(b)), it is clear that the reconstructed wall and the door are at right angle to each other. Similarly, from the side view (Fig. 5(c)), the top and front faces of boxes also form a right angle. From the top view, it can be seen that, although no orthogonality constraints were imposed between the top face of the boxes and the wall and door, these reconstructed faces are also orthogonal. Even though the Euclidean reconstructions from processes (i) and (ii) are not visually differentiable, quantitatively, these angles were estimated to be 87.98° and 87.81° for process (i) and 89.99° and 90.00° for process (ii).

The second experiment, a video sequence of images of a wall scene was captured by a hand-held digital video camcorder (Sony DCR-PC100). A number of image frames at the beginning and near the end of the video sequence were discarded due to unstable camera motion. The camera motion was mainly lateral (panning the scene from left to right) but some small up and down camera motion was also involved. In the last 50 frames, while the camera remained stationary, it zoomed slowly into the scene. Tests on this video sequence were carried out for our earlier work [22,23] where a less robust bundle adjustment process was implemented at that time. We tested this video data set again in the hope of seeing an improvement in the recovered 3D shape when bundle adjustment with scene constraints was included into the system.

The KLT feature tracker [40,41] was applied to the entire video sequence to extract image feature points. Out of the video sequence, 7 images at 20 frames apart were selected. As the full treatment of outlier detection and elimination (e.g. see [42–45]) is outside the scope of this paper and as only a few outliers were detected, we manually removed them and added a few image feature points used for the scene constraints, resulting in 140 image feature points in each image. Fig. 6 shows 5 of the 7 images of the video sequence. Seven image points on the lower face of the concrete stairs and twelve image points were randomly selected to form two sets of points for a pair of orthogonal scene planes. The same procedure as for the experiment on the indoor scene was carried out.

The final estimates of focal lengths and principal points of all the images are shown in Table 2. For process (i), the RMS reprojection errors before and after bundle adjustment were 1.25 and 1.16 pixels; the angles between the reconstructed wall and stair before and after bundle adjustment were estimated to be 94.33° and 91.26° respectively. For process (ii), these figures were 1.20 and 1.13 pixels and 89.69° and 90.00°, respectively. As the angle between the two orthogonal planes was estimated to be 90.16° in our previous work [22], bundle adjustment had indeed given an improvement to the reconstruction.

The Euclidean reconstruction from process (ii) is shown in Fig. 7. Feature points used in the scene constraint are shown as circles (O) and diamonds (◊) on the concrete stair and on the wall. The recovered optical centres of the images are labelled as stars (*). It can be seen from the figure that the wall and the stair form a right angle.

Since no ground truth of the camera intrinsic parameters is available, the accuracy of the estimates shown in the table cannot be assessed. However, the estimated values of focal length of the camera from image 229 to 269 do show an

<table>
<thead>
<tr>
<th>Image</th>
<th>Process (i): no scene constraints</th>
<th>Process (ii): with scene constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Focal length</td>
<td>Principal point</td>
</tr>
<tr>
<td>149</td>
<td>1249.51</td>
<td>(−55.66, 22.89)</td>
</tr>
<tr>
<td>169</td>
<td>1236.57</td>
<td>(−41.47, 22.85)</td>
</tr>
<tr>
<td>189</td>
<td>1206.02</td>
<td>(−4.97, 9.93)</td>
</tr>
<tr>
<td>209</td>
<td>1230.86</td>
<td>(−3.37, −9.63)</td>
</tr>
<tr>
<td>229</td>
<td>1224.18</td>
<td>(12.17, −17.25)</td>
</tr>
<tr>
<td>249</td>
<td>1226.58</td>
<td>(6.18, −29.37)</td>
</tr>
<tr>
<td>269</td>
<td>1363.23</td>
<td>(14.72, −22.68)</td>
</tr>
</tbody>
</table>
increasing trend, which concurs with the zooming-in of the camera.

6. Discussion

We found that the final Euclidean structure depended significantly on the initial estimate of the absolute dual quadric obtained by the linear method and that the gauge of the system also played an important part on the estimation of absolute dual quadric and the initial Euclidean structure. Other factors, such as the distribution of corresponding points in the images and the depth variation of objects in the scene, all make a contribution to the initial estimation of the absolute dual quadric.

As discussed in Section 2.2, the setting of the global coordinate system at the optical centre of the first image may not be an ideal choice. The reason is that video sequences are often taken by panning the camera across the scene and that such a global coordinate system will significantly amplify any small errors on the camera parameter estimates near the end of the video sequence. We found that choice of the upgrade matrix $A$ in the form given in (5) put the origin of the global coordinate system after the Euclidean reconstruction very close to, but not exactly at, the centroid of the camera optical centres of all the images. Further studies will be required to verify whether fixing the gauge at the optical centres’ centroid is the ideal choice for camera auto-calibration.

In the bundle adjustment steps of both processes (i) and (ii), convergence can be further sped up if the value of $\mu_k$ in (12) is carefully adjusted in each iteration for each data set. In our implementation, we followed the updates for $\mu_k$ as discussed in [38] (Page 569). Similarly, the value of $\rho$ in the penalty function method can be fine tuned for different data sets. For instance, if the initial Euclidean reconstruction is good and the value of $c(\theta)$ is small then a larger initial value of $\rho$ may be used.

While only constraints from orthogonal scene plane can be easily incorporated into the recovery of the absolute dual quadric for camera auto-calibration, many other scene constraints can be included into bundle adjustment with ease. Some of these are summarized below.

6.1. Line segments of equal length

Line segments of equal length often appear in scenes that have man-made objects, e.g. windows of the same width or height. Suppose that two line segments whose endpoints are $\{X_1^1, X_2^1\}$ and $\{X_3^3, X_4^3\}$ are known to have the same length. Then a scene constraint, $c_1$, can be formulated as

$$||c_1(X_1^1, X_2^1, X_3^3, X_4^3)||^2 \triangleq ||(X_2^1 - X_1^1)^T (X_2^1 - X_1^1) - (X_4^3 - X_3^3)^T (X_4^3 - X_3^3)||^2$$

$$= 0,$$

(22)

6.2. Coplanar scene points

Given that a number of scene points, $\{X_1^1, \ldots, X^n\}$, are known to be coplanar, by introducing the nuisance parameter $n$ which denotes the estimate of the plane normal in each iteration, we can formulate this scene constraint $c_2$ as

$$c_2(n, \{X_1^1, X_2^2, \ldots, X^n\}) \triangleq A n = 0,$$

(23)

where $A = [X_1^1, X_2^2, \ldots, X^n]^T \in \mathbb{R}^{p \times 4}$.

6.3. Collinear scene points

Suppose that three scene points, $X_1^1, X_2^2, X_3^3$, are known to be collinear. This constraint can be formulated as

$$||c_3(X_1^1, X_2^2, X_3^3)||^2 \triangleq ||(X_1^1, X_2^2, X_3^3)^T X^1_1||^2 = 0,$$

(24)

where $[\cdot]^\times$ denotes the skew symmetric matrix formed by the vector.
Appendix. The BFGS and DFP updates

Given the gradient and the Hessian approximations, \( \mathbf{g}_k \) and \( \mathbf{H}_k \), at iteration \( k \), the Broyden-Fletcher-Goldfarb-Shanno (BFGS) formula is widely used for estimating the Hessian matrix for the next iteration. The update is given by [39] (page 119):

\[
\mathbf{H}_{k+1} = \mathbf{H}_k - \frac{1}{\mathbf{y}_k^T \mathbf{H}_k \mathbf{y}_k} \mathbf{H}_k \mathbf{y}_k \mathbf{y}_k^T \mathbf{H}_k + \frac{1}{\mathbf{y}_k^T \mathbf{y}_k} \mathbf{y}_k \mathbf{y}_k^T,
\]

where \( \mathbf{y}_k = \mathbf{g}_{k+1} - \mathbf{g}_k \). If no additional information is available, the initial Hessian approximation, \( \mathbf{H}_0 \), is usually taken as an identity matrix.

The Davidon-Fletcher-Powell (DFP) update on the other hand is defined as [39] (page 119):

\[
\mathbf{H}_{k+1} = \mathbf{H}_k - \frac{1}{\mathbf{y}_k^T \mathbf{H}_k \mathbf{y}_k} \mathbf{H}_k \mathbf{y}_k \mathbf{H}_k^T + \frac{1}{\mathbf{y}_k^T \mathbf{y}_k} \mathbf{y}_k \mathbf{y}_k^T \mathbf{H}_k + \mathbf{y}_k \mathbf{y}_k^T \mathbf{H}_k \mathbf{y}_k \mathbf{y}_k^T
\]

where

\[
\mathbf{w}_k = \frac{1}{\mathbf{y}_k^T \mathbf{y}_k} \mathbf{y}_k - \frac{1}{\mathbf{y}_k^T \mathbf{H}_k \mathbf{y}_k} \mathbf{H}_k \mathbf{y}_k \mathbf{y}_k^T \mathbf{H}_k \mathbf{y}_k.
\]

References


