

# Quadratic Surface Reconstruction from Multiple Views Using SQP

Rubin Gong and Gang Xu, Regular Member

**SUMMARY** We propose using SQP (Sequential Quadratic Programming) to directly recover 3D quadratic surface parameters from multiple views. A surface equation is used as a constraint. In addition to the sum of squared reprojection errors defined in the traditional bundle adjustment, a Lagrangian term is added to force recovered points to satisfy the constraint. The minimization is realized by SQP. Our algorithm has three advantages. Firstly, given corresponding features in multiple views, the SQP implementation can directly recover the quadratic surface parameters optimally instead of a collection of isolated 3D points coordinates. Secondly, SQP guarantees that the constraint is strictly satisfied. Thirdly, the camera parameters and 3D coordinates of points can be determined more accurately than that by unconstrained methods. Experiments with both synthetic and real images show the power of this approach.

**Key words:** Quadratic surface reconstruction, constrained minimization, Sequential Quadratic Programming, bundle adjustment, error analysis

## 1. INTRODUCTION

Recovering 3D structures and camera intrinsic and extrinsic parameters from a given sequence of images usually needs bundle adjustment to refine in an optimal way to get a more accurate solution. There is an excellent survey of the theory of bundle adjustment as well as many implementation strategies in [1]. Traditional bundle adjustment aims at recovering isolated 3D features using nonlinear unconstrained optimization methods. Since it does not rely on relations between isolated features, a very wide variety of scenarios can be handled. However, in practice, scenes often contain some prior 3D constraints, such as 3D distances and planar constraints. If applied carefully, more accurate 3D scene structure and camera parameters can be recovered [2,5]. Furthermore, in many scenes such as indoor scenes and man made objects, scenes often contain structures with strong geometry regularities such as floors, walls and globe. And it is more suitable to use parameterized models than isolated features to represent these objects [2,3,4,5]. Some previous work has already noted this. For example, G. Cross et al proposed recovering quadric surfaces from multiple views in [3]. Ying Shan et al utilized the point-on-surface constraint in their model-based bundle adjustment method to directly recover face model from multiple views [2].

When scene constraints are incorporated into bundle adjustment, nonlinear constrained minimization methods are

needed to minimize the objective cost function while keeping the specified constraints strictly satisfied. Previous work has given their own nonlinear constrained minimization methods to incorporate constraints in bundle adjustment. For example, the work in [2] has used a kind of penalty method that converts constrained minimization problem into an unconstrained one. The work in [4] has described a scheme for incorporating surface and other scene constraints into a VSDF filter to directly recover the surfaces and camera motion.

In this paper Sequential Quadratic Programming (SQP) [1,6,7,8,9,10] is used to incorporate scene constraints in bundle adjustment to directly recover quadratic surface parameters from multiple views. SQP is a powerful constrained minimization method and has been successfully applied in a wide variety of industrial fields. It has a concise mathematical formulation and can incorporate a wide variety of constraints. Triggs [10] has used it in camera calibration before. However, it has seldom been used in 3D scene reconstruction so far. In this paper the SQP concept is introduced and a novel implementation that aims at solving constrained bundle adjustment problem is given. The point-on-surface constraint described in [2] is used to directly recover quadratic surfaces from multiple views using SQP. The proposed technique is different from previous quadratic surface reconstruction work [3,11,12,13], where the outlines in multiple views needed to be estimated first to recover the corresponding quadratic surface parameters. In principle the technique introduced in this paper can be used to recover arbitrary parametric models such as lines, planes and freeform surfaces from multiple views. In the experiment section the power of SQP is verified. It can be seen that the specified equality constraints are strictly satisfied. Since external geometry constraints are incorporated, the 3D scene points coordinates and camera parameters are recovered more accurately by SQP than that by unconstrained algorithms.

The paper is organized as follows. In Section 2, we formulate quadratic surface reconstruction problem. In Section 3, we describe SQP nonlinear minimization concept and its novel implementation. In Section 4, we outline the steps in quadratic surface reconstruction. In Section 5, the experimental results with both computer simulation data and real images are shown. And we conclude the paper in Section 6.

## 2. FORMULATION

The quadratic surface reconstruction problem is formulated in

this Section.

### 2.1 Quadratic Surface Representation

A quadratic surface is a second-order algebraic surface given by:

$$h(\mathbf{X}, \mathbf{Q}) = \mathbf{X}^T \mathbf{Q} \mathbf{X} = 0, \quad (1)$$

where  $\mathbf{Q}$  is a  $4 \times 4$  symmetric matrix,  $\mathbf{X}=(x,y,z,1)^T$  is a homogeneous 4-vector which represents a point in 3D. Some instances of quadratic surfaces are shown in Figure 1.

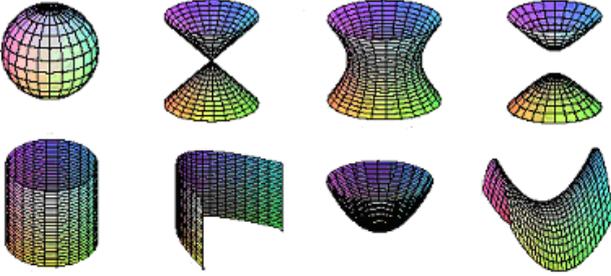


Figure 1. Some instances of quadratic surfaces. First row: ellipsoid, cone, hyperboloid of one sheet and hyperboloid of two sheets. Second row: elliptic cylinder, parabolic cylinder, paraboloid and hyperbolic paraboloid

A quadratic has nine degrees of freedom corresponding to the independent elements of  $\mathbf{Q}$  up to an overall scale. Equation (1) can be rewritten in the form:

$$\mathbf{a}\mathbf{v} = 0, \quad (2)$$

where  $\mathbf{a}$  is a  $1 \times 10$  matrix which can be decided by point  $\mathbf{X}$  only,  $\mathbf{v}$  is a homogeneous 10-vector containing the distinct matrix elements of  $\mathbf{Q}$ . Each 3D point provides a similar constraint, so that from  $N$  points a matrix equation  $\mathbf{A}\mathbf{v}=\mathbf{0}$  can be constructed, where  $\mathbf{A}$  is  $N \times 10$  matrix formed from the stacked matrices  $\mathbf{a}$ . The solution of  $\mathbf{v}$  corresponds to the one dimension null-space of  $\mathbf{A}$ . If  $N \geq 9$  and the  $N$  points are in general position then the quadratic surface parameters can be uniquely determined.

### 2.2 3D Reconstruction from Multiple Views

Suppose we have matched a number of points of interest across  $M$  images using for example the technique described in [14]. Because of occlusion, feature detection failure and other reasons, a scene point can only be observed and detected in a subset of the  $M$  images (c.f. Figure 2).

Suppose a 3D point  $\mathbf{X}$  is observed as  $\mathbf{x}=\mathbf{P}\mathbf{X}$ ,  $\mathbf{x}'=\mathbf{P}'\mathbf{X}$  in arbitrary two images, where image points  $\mathbf{x}, \mathbf{x}'$  are represented by homogeneous 3-vectors,  $\mathbf{x}=(x,y,1)^T$ , and  $\mathbf{P}, \mathbf{P}'$  are  $3 \times 4$  camera projection matrices for the two views. Given the fundamental matrix  $\mathbf{F}$  for the view pair, then from [15,16,17] the camera matrices can be chosen as:

$$\mathbf{P} = [\mathbf{I} | \mathbf{O}], \mathbf{P}' = [[\mathbf{e}']_{\times} \mathbf{F} | \mathbf{e}'],$$

where  $\mathbf{e}'$  is the epipole in the second image ( $\mathbf{F}^T \mathbf{e}' = 0$ ) and  $[\mathbf{e}']_{\times}$  is the  $3 \times 3$  skew matrix such that  $[\mathbf{e}']_{\times} \mathbf{x} = \mathbf{e}' \times \mathbf{x}$ . The 3D

point  $\mathbf{X}$  is then reconstructed from its image correspondence  $\mathbf{x} \leftrightarrow \mathbf{x}'$  by back-projection (via  $\mathbf{P}, \mathbf{P}'$ ) and triangulation[18]. After the projective reconstruction is obtained, the technique described in [19] can be used to upgrade the projective reconstruction to a metric one.

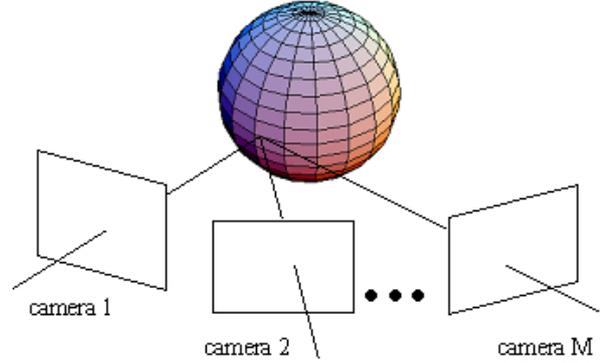


Figure 2. Quadratic surfaces observed by multiple views

### 2.3 Traditional Unconstrained Bundle Adjustment

If initial parameters have been estimated by linear method as illustrated above, a bundle adjustment step is often used to refine initial parameters. A cost function needs to be defined in bundle adjustment to quantify the fitting error of the estimated parameters. In traditional unconstrained minimization method, the cost estimation is often obtained by minimizing the sum of squared errors between the observed image points and the predicted image points. More formally, it can be represented as:

$$C = \sum_{j=1}^M \sum_{i=1}^N \left\{ \left( f_j \frac{X_{ij}}{Z_{ij}} + u_{j0} - u_{ij} \right)^2 + \left( f_j \frac{Y_{ij}}{Z_{ij}} + v_{j0} - v_{ij} \right)^2 \right\}$$

where  $M$  is the total number of cameras,  $N$  is the total number of points.  $f_j$  is the focal length of  $j$ th camera.  $X_{ij}$ ,  $Y_{ij}$  and  $Z_{ij}$  are the  $i$ th point's coordinates under  $j$ th camera's coordinate system.  $u_{ij}$  and  $v_{ij}$  are the coordinates of  $i$ th point observed in  $j$ th image. Since 3D scene constraints are often not enforced in traditional unconstrained bundle adjustment, the optimized isolated features do not satisfy geometry constraints. For example, if a quadratic surface is recovered using traditional unconstrained minimization method, the isolated feature points will not be strictly on the same quadratic surface.

## 3. SEQUENTIAL QUADRATIC PROGRAMMING

In order to impose 3D scene constraints in optimization, constrained minimization is often needed in bundle adjustment algorithm. Sequential Quadratic Programming is a powerful algorithm and has been proved highly effective for solving general constrained optimization problems.

### 3.1 SQP Concept

#### 3.1.1 Problem Formulation

First the basic SQP principle is introduced. Consider the general equality constrained minimization problem  $P$ :

$$(P) \min_{\mathbf{x} \in \mathbf{R}^n} f(\mathbf{x})$$

$$\text{s.t. } h_j(\mathbf{x})=0 \quad j=1, \dots, m.$$

Here  $\mathbf{x}$  is the desired variable vector,  $f(\mathbf{x})$  is the objective function and  $h_j(\mathbf{x})$  is the equality constraint. The Lagrangian function associated with problem (P) is:

$$L(\mathbf{x}, \mathbf{u}) = f(\mathbf{x}) + \mathbf{h}(\mathbf{x})^T \mathbf{u}, \quad \mathbf{u} \in \mathbf{R}^m, \quad (3)$$

where  $\mathbf{u}$  is the corresponding Lagrangian multiplier vector and  $\mathbf{h}(\mathbf{x})$  is the vector of equality constraints..

### 3.1.2 Local Analysis

Given a current solution  $(\mathbf{x}_k, \mathbf{u}_k)$  which is sufficiently close to an optimal solution  $(\mathbf{x}^*, \mathbf{u}^*)$ , we seek to locally approximate problem (P) by a quadratic sub-problem, i.e., an optimization problem with a quadratic objective function and linear constraints. The form of the quadratic subproblem most often found in literature [8], and the one that will be employed here, is

$$(QP) \min_{\mathbf{d}_x} \nabla f(\mathbf{x}_k)^T \mathbf{d}_x + \frac{1}{2} \mathbf{d}_x^T \nabla^2 L(\mathbf{x}_k, \mathbf{u}_k) \mathbf{d}_x \quad (4)$$

$$\text{s.t. } \nabla \mathbf{h}(\mathbf{x}_k) \mathbf{d}_x + \mathbf{h}(\mathbf{x}_k) = \mathbf{0}, \quad (5)$$

where  $\mathbf{d}_x = \mathbf{x} - \mathbf{x}_k$ . From the first order optimality conditions for the quadratic subproblem [8], the following equations can be obtained to compute the update directions  $(\mathbf{d}_x, \mathbf{d}_u)$ :

$$\nabla^2 L(\mathbf{x}_k, \mathbf{u}_k) \mathbf{d}_x + \nabla f(\mathbf{x}_k) + \nabla \mathbf{h}(\mathbf{x}_k)^T (\mathbf{u}_k + \mathbf{d}_u) = 0$$

$$\nabla \mathbf{h}(\mathbf{x}_k) \mathbf{d}_x = -\mathbf{h}(\mathbf{x}_k)$$

The equations can be rewritten in matrix format as:

$$\begin{bmatrix} \nabla^2 L(\mathbf{x}_k, \mathbf{u}_k) & \nabla \mathbf{h}(\mathbf{x}_k)^T \\ \nabla \mathbf{h}(\mathbf{x}_k) & \mathbf{O} \end{bmatrix} \begin{bmatrix} \mathbf{d}_x \\ \mathbf{d}_u \end{bmatrix} = - \begin{bmatrix} \nabla L(\mathbf{x}_k, \mathbf{u}_k) \\ \mathbf{h}(\mathbf{x}_k) \end{bmatrix}, \quad (6)$$

where  $\mathbf{d}_u = \mathbf{u} - \mathbf{u}_k$ . The solution of  $\mathbf{d}_x$  and  $\mathbf{d}_u$  can be used to generate the new iterate. If we choose a suitable step-size  $\alpha_k$ , the new iterate can be defined as:

$$(\mathbf{x}_{k+1}, \mathbf{u}_{k+1}) = (\mathbf{x}_k, \mathbf{u}_k) + \alpha_k (\mathbf{d}_x, \mathbf{d}_u).$$

Once the new iterate is constructed, a set of new linear equations can be built and solved at point  $(\mathbf{x}_{k+1}, \mathbf{u}_{k+1})$ . In the analysis above it has been assumed that the current solution  $(\mathbf{x}_k, \mathbf{u}_k)$  is sufficiently close to the optimal solution and the quadratic subproblem is always feasible. For the quadratic sub-problem to be solved, four conditions [8] should be satisfied. It has also been proved that if the initial solution is sufficiently close to the optimal  $\mathbf{x}^*$ , the algorithm has quadratic local convergence property [8].

### 3.1.3 Global Analysis

If the current solution  $(\mathbf{x}_k, \mathbf{u}_k)$  is not sufficiently close to an optimal solution  $(\mathbf{x}^*, \mathbf{u}^*)$ , the questions of whether the

sequences generated by quadratic programming will converge must then be resolved. To ensure global convergence, SQP needs to be equipped with a measure of progress, a merit function  $\phi$ , whose reduction implies progress towards an optimal solution[7,8]. The merit function used in constrained minimization must blend the need to reduce the objective function while keeping the constraints satisfied. And it is generally different from the unconstrained one.

One commonly used merit function is called  $\ell_1$  penalty merit function[7,8]. It can be written as:

$$\phi_1(\mathbf{x}; \rho) = f(\mathbf{x}) + \rho \sum_i |h_i(\mathbf{x})|,$$

where  $\rho$  is a positive constant to be chosen and  $|\cdot|$  means the absolute value of a function. It is sufficient to note that  $\phi_1$  is an exact penalty function [7,8]; that is, there exists a positive  $\rho^*$  such that for all  $\rho \geq \rho^*$ , an unconstrained minimum of  $\phi_1$  corresponds to a solution of the constrained nonlinear minimization problem.

### 3.2 New SQP Implementation

In practical implementation there are some problems need to be considered. We have assumed that the quadratic subproblem always has a feasible solution in the analysis above. To have a feasible solution, it has been illustrated in [7,8] that the system of constraints of the quadratic subproblem must have a nonempty feasible set and the quadratic objective function should be bounded below on that set. If the initial solution  $\mathbf{x}_k$  is sufficiently close to the optimized solution, the above consistence conditions can be guaranteed. For nonlocal points, it is not necessarily true. An appropriate estimate of  $\nabla^2 L(\mathbf{x}_k, \mathbf{u}_k)$  can assure that a consistent quadratic problem will always have a solution. Some implementations have used BFGS algorithm to approximate the Hessian matrix. In our work, a novel implementation is used to avoid the infeasibilities. Consider function (4) in QP, for nonlocal point  $(\mathbf{x}_k, \mathbf{u}_k)$ , it may be a poor local approximation to solve the original problem (P). In that case, the original problem can only be described as below:

$$(SP) \min_{\mathbf{d}_x} L(\mathbf{x}, \mathbf{u}) \quad (7)$$

$$\text{s.t. } \nabla \mathbf{h}(\mathbf{x}_k) \mathbf{d}_x + \mathbf{h}(\mathbf{x}_k) = \mathbf{0}.$$

Since solution  $(\mathbf{x}_k, \mathbf{u}_k)$  is a remote point, to minimize function (7), about all we can do is to take a step down the gradient, as in the steepest descent method[20]. It can be formally represented as:

$$\mathbf{x}_{k+1} - \mathbf{x}_k = -\eta \nabla L(\mathbf{x}_k, \mathbf{u}), \eta > 0.$$

It can be rewritten as:

$$\lambda \mathbf{d}_x + \nabla \mathbf{h}(\mathbf{x}_k)^T \mathbf{d}_u = -\nabla L(\mathbf{x}_k, \mathbf{u}_k), \lambda = \frac{1}{\eta}. \quad (8)$$

Here  $\lambda$  is a suitable value and it should not exhaust the downhill direction. Equation (5) and (8) can be combined and rewritten in matrix format as:

$$\begin{bmatrix} \lambda \mathbf{I} & \nabla \mathbf{h}_k^T \\ \nabla \mathbf{h}_k & \mathbf{O} \end{bmatrix} \begin{bmatrix} \mathbf{d}_x \\ \mathbf{d}_u \end{bmatrix} = - \begin{bmatrix} \nabla L(\mathbf{x}_k, \mathbf{u}_k) \\ \mathbf{h}_k \end{bmatrix}, \quad (9)$$

where  $\mathbf{I}$  is a  $n \times n$  diagonal matrix. Combine equation (6) and (9), we get the following equation (10):

$$\begin{bmatrix} \mathbf{B}_k & \nabla \mathbf{h}_k^T \\ \nabla \mathbf{h}_k & \mathbf{O} \end{bmatrix} \begin{bmatrix} \mathbf{d}_x \\ \mathbf{d}_u \end{bmatrix} = - \begin{bmatrix} \nabla L(\mathbf{x}_k, \mathbf{u}_k) \\ \mathbf{h}_k \end{bmatrix}, \quad (10)$$

where  $\mathbf{B}_k = \nabla^2 L(\mathbf{x}_k, \mathbf{u}_k) + \lambda \mathbf{I}$ .

The new equation has combined the merits of steepest descent method and Newton method. When  $\mathbf{x}_k$  is sufficiently close to  $\mathbf{x}^*$ ,  $\lambda$  can be adjusted to be very small, the modified matrix  $\mathbf{B}_k$  is very close to the Hessian matrix. The Newton direction is used to approximate the next QP step. When  $\mathbf{x}_k$  is not sufficiently close to  $\mathbf{x}^*$ ,  $\lambda$  can be adjusted to be very large, the matrix  $\mathbf{B}_k$  is forced to be diagonally dominant, the steepest descent direction is mainly used to approximate the next step. It has been proved that add a strictly positive diagonal matrix to  $\mathbf{B}_k$  can produce generally more robust results than by basic SQP implementation [8]. Given an initial guess for parameters  $\mathbf{x}$ , our SQP implementation can be described as below:

#### SQP Implementation

1. Compute  $\phi_1(\mathbf{x}; \rho)$ .
2. Pick a modest value for  $\lambda$ , say  $\lambda = 0.001$ .
3. Solve the linear equations in (10) for  $(\mathbf{d}_x, \mathbf{d}_u)$  and evaluate  $\phi_1(\mathbf{x} + \mathbf{d}_x; \rho)$ .
4. If  $\phi_1(\mathbf{x} + \mathbf{d}_x; \rho) \geq \phi_1(\mathbf{x}; \rho)$ , increase  $\lambda$  by a factor of 10 and go back to step 3.
5. If  $\phi_1(\mathbf{x} + \mathbf{d}_x; \rho) < \phi_1(\mathbf{x}; \rho)$ , decrease  $\lambda$  by a factor of 10, update the trial solution  $(\mathbf{x}, \mathbf{u}) = (\mathbf{x}, \mathbf{u}) + (\mathbf{d}_x, \mathbf{d}_u)$ , and go back to step 3.

For the algorithm to stop, the same strategy employed in Levenberg Marquardt algorithm [20] has been used. The loop is stopped at the first occasion where  $\phi_1(\mathbf{x}; \rho)$  decreases by a negligible amount. Once the acceptable minimum has been found, we set  $\lambda = 0$  and compute the matrix  $\mathbf{B}_k^{-1}$ , the upper left part of which is the standard covariance matrix of the standard

errors in the fitted parameters  $\mathbf{x}$  [20].

### 3.3 Application In Quadratic Surface Recovering

In quadratic surface recovering applications, the quadratic surface equation is used as the constraint function. The corresponding Lagrangian function associated with our problem is:

$$LC = C + \sum_{j=1}^N u_j h(\mathbf{X}_j, \mathbf{Q}),$$

where  $u_j$  is the Lagrange multiplier,  $\mathbf{X}_j$  is  $j$ th point coordinate vector in the world coordinate system,  $\mathbf{Q}$  is the quadratic model matrix. The objective is to minimize function LC under constraints  $h(\mathbf{X}_j, \mathbf{Q}) = 0$ . The merit function used in our implementation is defined as:

$$\phi_1 = C + \sum_{j=1}^N |h(\mathbf{X}_j, \mathbf{Q})|,$$

where  $\rho > \max\{|u_j| | 1 \leq j \leq N\}$ ,  $N$  is the total number of constraints,  $||$  means the absolute value of a variable or a function.

## 4. OUTLINE OF THE METHOD

The algorithm can be outlined as follows:

1. Data Preparation: Collect matched image points across multiple views.
2. Compute the initial 3D points coordinates, intrinsic and extrinsic camera parameters and quadratic surface parameters using linear method.
3. Optimize the initial 3D point coordinates using the SQP implementation.
4. Build corresponding VRML model using the refined camera parameters and sphere parameters only.

## 5. EXPERIMENTAL RESULTS

In this section, we provide experimental results of our algorithm with both synthetic and real data for globe.

### 5.1 Synthesized Data

For the synthetic data, 3 views and total 16 points on a sphere surface are used. The 3 images have the same focal length of 1000 pixels. For each image, 16 image points are generated with isotropic uniform Gaussian noise of  $\delta = 1.0$ . In Table 1 eight of the sixteen points' true 3D coordinates are listed. We first calculate the initial camera parameters and points coordinates using linear methods.

#### Comparison between SQP and Levenberg Marquardt algorithm

After we have calculated the initial solutions, SQP and Levenberg Marquardt algorithm are used to optimize the initial solutions. SQP optimization converges within 10 steps. The constraint functions become strictly satisfied after SQP

optimization step. The maximum absolute error of the constraint function is not more than  $5.0E-7$ . The cost function  $C$  has the value 8.32247. For each pixel, the mean error is 0.416395 pixels. The optimized 3D coordinates of the eight points are listed in Table 1. It can be seen that the solutions computed by SQP are generally more close to the true ones. The true intrinsic and extrinsic camera parameters and the optimized values are listed in Table 2. Here the rotation is represented in angle/axis format. The translation vector between the first camera and the second camera is normalized to 1. So the translation vector has only two free parameters. It can also be seen that the camera parameters computed by SQP are generally more close to the true ones. The true sphere parameters and the optimized values are listed in Table 3. Here the sphere parameters of Levenberg Marquardt algorithm are estimated from the optimized 3D coordinates using the technique described in Section 2. It can also be seen that the sphere parameters computed by SQP are more close to the true parameters. The differences of the sphere surface constraints between SQP and Levenberg Marquardt algorithm are shown in Table 4. The solutions computed by SQP strictly satisfy the point-on-surface constraint. But the solutions computed by Levenberg Marquardt algorithm often deviate from the sphere surface.

		X	Y	Z
True Coordinates	1	1.41421	-1.41421	8.48528
	2	-1.41421	-1.41421	8.48528
	3	-1.41421	1.41421	8.48528
	4	1.41421	1.41421	8.48528
	5	1.41421	-1.41421	5.65685
	6	-1.41421	-1.41421	5.65685
	7	-1.41421	1.41421	5.65685
	8	1.41421	1.41421	5.65685
SQP Optimized Coordinates	1	1.40814	-1.41648	8.52811
	2	-1.40798	-1.4096	8.52583
	3	-1.41873	1.39237	8.52222
	4	1.39431	1.41909	8.52914
	5	1.40114	-1.42392	5.70297
	6	-1.42115	-1.41615	5.72465
	7	-1.41586	1.40802	5.72112
	8	1.40422	1.39093	5.68323
LM method	1	1.40811	-1.41665	8.54785
	2	-1.40574	-1.40838	8.53672
	3	-1.41823	1.39036	8.53317
	4	1.38928	1.41385	8.53411
	5	1.40048	-1.42255	5.72538
	6	-1.42194	-1.41778	5.73604
	7	-1.41282	1.40503	5.73891
	8	1.41259	1.39568	5.68705

Table 1. True coordinates of 8 points and the optimized coordinates calculated by SQP and Levenberg Marquardt algorithm

### Standard Deviation Comparison between SQP and Levenberg Marquardt algorithm

The square roots of the diagonal items of the upper left part of the inversed optimized gradient matrix  $\mathbf{B}_k^{-1}$  represent the standard deviations of the fitted parameters [20]. If any diagonal item is negative, it means that the corresponding parameter has

		First Camera	Second Camera	Third Camera
Focal Length (pixel)	True Value	1000	1000	1000
	SQP	1011.45	1001.88	990.836
	LM Method	1014.226	997.807	985.468
Rotation Vector (Angle/Axis Representation)	True Value	NULL	0 , 1.5708, 0	0., 3.14159, 0.
	SQP	NULL	0.00673965, 1.56541, -0.000108669	-0.00706039, 3.14312, -0.00467069
	LM Method	NULL	0.00701384, 1.56397, 0.000972825	-0.00718367, 3.14376, -0.00434953
Translation vector	True Value	NULL	-0.707107, 0.00000188, 0.707107	0., 0., 1.41421
	SQP	NULL	-0.711301, 0.00238217, 0.702883	0.00107775, 0.0025828, 1.41158
	LM Method	NULL	-0.712693, 0.00304105, 0.701469	0.00159841, 0.00259099, 1.40893

Table 2. True camera parameters and the optimized solutions calculated by SQP and Levenberg Marquardt algorithm

	Sphere center	Sphere radius
True Sphere parameters	0 , 0., 7.07107	2.44949
SQP Optimized sphere parameters	0.00465979, -0.0049935, 7.11528	2.44093
LM method Sphere parameters	0.00236943, -0.00711532, 7.12735	2.43807

Table 3. True sphere parameters and the optimized solutions calculated by SQP and Levenberg Marquardt algorithm

Point number	SQP method	LM method
1	8.32667E-7	0.00380352
2	-8.32667E-7	-0.0011578
3	1.80411E-7	0.000314387
4	1.38778E-7	-0.00225693
5	-2.77556E-7	-0.00205238
6	1.387778E-7	0.00101718
7	5.55112E-7	-0.00195141
8	4.16334E-7	0.00868387

Table 4. The comparison of the sphere surface constraints between SQP and Levenberg Marquardt algorithm

low confidence. The standard deviations of the 3D coordinates of the points are listed in Table 5. It can be seen that the standard deviation values of the coordinates of the points computed by SQP are generally smaller than that by Levenberg Marquardt algorithm. The standard deviations of the camera intrinsic and extrinsic parameters are listed in Table 6. It can also be seen that the standard deviations of the camera parameters computed by

SQP are generally smaller than that by Levenberg Marquardt algorithm. It means that the camera parameters computed by SQP generally have higher confidence than that by Levenberg Marquardt algorithm. The standard deviations of the sphere parameters computed by SQP are listed in Table 7.

Point No.		X std.	Y std.	Z std.
1	SQP Algorithm	0.00681054	0.00607517	0.0260046
2		0.00650413	0.00660919	0.0287544
3		0.00653493	0.0065542	0.0287934
4		0.00673846	0.00608316	0.0260732
5		0.00549643	0.00512055	0.0261408
6		0.00567624	0.00584895	0.0251242
7		0.00566292	0.00582977	0.0251111
8		0.00551058	0.00503517	0.0261631
1	LM Method	0.00850022	0.00860365	0.027719
2		0.00919474	0.00856876	0.0324558
3		0.00925611	0.00848638	0.0325092
4		0.00836699	0.00857979	0.0277633
5		0.00641225	0.0063829	0.0283795
6		0.00752631	0.00723495	0.0268025
7		0.00749629	0.00719085	0.0268104
8		0.00641929	0.00625313	0.0284283

Table 5. The standard deviations of the eight points computed by SQP algorithm (std means standard deviations)

		First Camera std.	Second Camera std.	Third Camera std.
Focal Length (pixel)	SQP Algorithm	5.38256	5.58726	5.20935
Rotation Vector		NULL	0.0133593 0.0148237 0.0131113	0.0162009 0.0109167 0.0171487
Translation vector		NULL	0.0361276 0.00973273	0.00796632 0.0079622 0.0502892
Focal Length (pixel)	LM algorithm	6.406	5.96666	6.1163
Rotation Vector		NULL	0.0139427 0.0374338 0.0134927	0.0162192 0.0111792 0.0174623
Translation vector		NULL	0.038479 0.0104745	0.00834412 0.00813186 0.0700069

Table 6. The standard deviations of the camera parameters computed by SQP algorithm (std means standard deviations)

Sphere Center std.			Sphere Radius std.
X	Y	Z	
0.0026145	0.00195009	0.0244212	0.0854269

Table 7. The standard deviations of the sphere parameters of SQP method (std means standard deviations)

### Relationship between constraints number and standard deviation

We have also done experiments to verify the impact of the number of constraints on the final standard deviations. Some results are shown in Table 8. When there are no constraints,

SQP degenerates into Levenberg Marquardt algorithm. By experiment we find that at least 4 points are needed in our experiment. When the constraints number is less than 4, SQP will not be able to calculate the sphere parameters correctly. In theory the sphere surface has 4 degrees of freedom and at least 4 points are needed, it is compatible to the experiment. We have also found that when the number of constraints increases, the standard deviations of the parameters decrease. This means that the optimized parameters become more accurate as the number of constraints increases. The standard deviations of the four sample points computed by SQP are listed in Table 8.

Point No.	Constraint Number	X std.	Y std.	Z std.
1	5	0.000851589	0.000860086	0.00278381
2		0.000924755	0.000857985	0.00326882
3		0.000926656	0.000852813	0.00327364
4		0.000839189	0.00086563	0.0027939
1	8	0.000723202	0.000658131	0.00276766
2		0.00068708	0.000688759	0.00302997
3		0.000689652	0.000683205	0.00303401
4		0.000713429	0.000659267	0.00277507
1	12	0.000686748	0.000615457	0.0026474
2		0.000656274	0.000666276	0.002901
3		0.000659328	0.000660684	0.00290518
4		0.000679311	0.000616335	0.00265428
1	16	0.000681054	0.000607517	0.00260046
2		0.000650413	0.000660919	0.00287544
3		0.000653493	0.00065542	0.00287934
4		0.000673846	0.000608316	0.00260732

Table 8. The standard deviations of the four points computed by SQP algorithm with different constraint numbers

### Relationship between cost function and gaussian noise

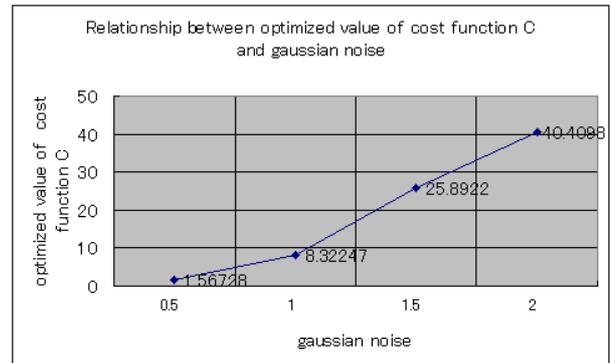


Figure 3. Relation between optimized cost function C and gaussian noise

We have also done experiments with different Gaussian noise parameter  $\delta$ . The constraint functions are strictly satisfied in SQP optimization, the maximum absolute error of the constraint function is not more than  $1.0E-5$ . The SQP implementation converges within 10 steps. The relationship between cost function C and Gaussian noise  $\delta$  is shown in Figure 3. It can be seen from the figure that the value of the cost function C increases when the image Gaussian noise increases.

## 5.2 Real Images

For experiments on real images, 17 pictures have been taken around a globe. The camera is Power Shot Pro 70, a digital camera manufactured by Canon. The images have the same size. The size is 1525\*1021 pixels. Four of them are shown in Figure 4. And the initial 3D points coordinates, camera intrinsic and extrinsic parameters are calculated using the linear methods. The sphere parameters can then be calculated using the technique described in Section 2.

### SQP optimization

We first feed the initial 3D coordinates into SQP optimization loop. The SQP implementation converges within 20 steps. For the real images, the constraints are all strictly satisfied. The maximum absolute error of the constraint function is not more than  $1.0E-6$ . But the points computed by Levenberg Marquardt algorithm deviates from sphere surface by 4.5% or so. Once the refined camera parameters and sphere parameters are computed, the small number of parameters can be used to build the globe VRML model.



Figure 4. Four pictures of sequence of pictures

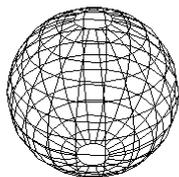


Figure 5. Automatically generated sphere points

### Automatic texture generation

Instead of using traditional methods to match the feature points among different pictures, the sphere parameters are used to generate the arbitrary number of points located on the surface as shown in Figure 5. The triangle patches and quadrilateral patches are then generated and projected onto different images. The normal of the 3D triangle/quadrilateral patches are then calculated. The vectors between sphere center and camera center are also calculated. Then we calculate the smallest angle between the normal of the 3D patch and sphere-camera vectors,

the smallest angle of each 3D patch is found and the corresponding 2D image patch is selected as the texture. Figure 6 shows the selected patches on one picture. By combining the selected texture patches and the automatically generated 3D points coordinates, we then build the VRML model as shown in Figure 7. The model that we built here is fine except some neighbor patches coming from different images have different brightness, and some lines and characters coming from two images cannot connect smoothly. We need another step to blend the texture patches to generate a better surface map around the globe [21]. So that the thin lines and characters can connect smoothly even the corresponding texture patches coming from different source images.

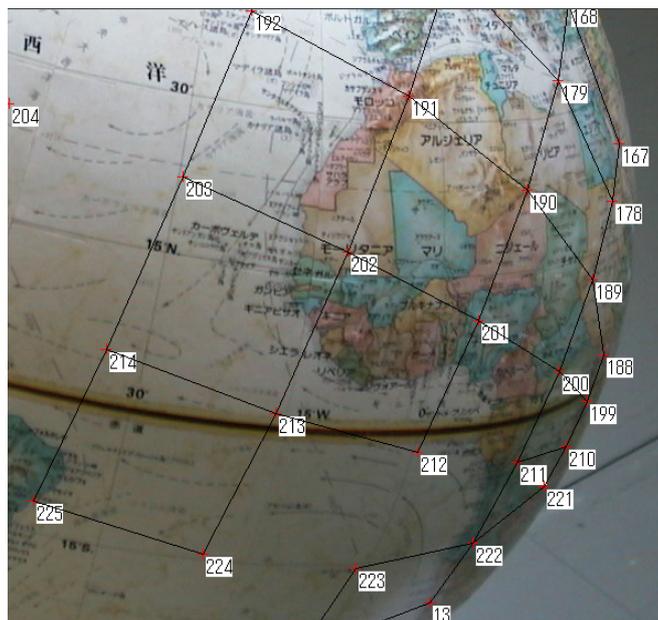


Figure 6. Most suitable texture patch

## 6. CONCLUSIONS

In this paper, we have proposed using SQP to incorporate model knowledge into traditional bundle adjustment step. A novel SQP implementation is used to directly recover quadratic surface models. Our experiment results reveal that sequential quadratic programming can generally generate more accurate results than that by unconstrained minimization methods while keeping the specified equality constraints strictly satisfied. Furthermore, SQP can incorporate arbitrary constraints that can be written in smooth function format. It can be applied in a wide variety of applications, ranging from camera calibration to 3D shape reconstruction. There are some limitations in our work. The major computation cost of our current implementation is the approximated Hessian matrix computation. We are planning to use the sparseness matrix properties to speed up its computation. And we also need to do the texture blending part to have a visually smooth surface map. We would like to apply SQP to model more free form parametric surfaces such as face

modeling, human body and arms.



Figure 7. VRML model of the virtual globe

## REFERENCES

- [1] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. *Bundle adjustment -- A modern synthesis*. In B. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, LNCS, pages 298--375. Springer Verlag, 2000.
- [2] Ying Shan, Zicheng Liu, Zhengyou Zhang: Model-Based Bundle Adjustment with Application to Face Modeling. ICCV:644-651, 2001
- [3] G. Cross and A. Zisserman. Quadratic surface reconstruction from dual-space geometry. In Proc. ICCV, pages 25-31, 1998.
- [4] P. McLauchlan, X. Shen, P. Palmer, A. Manassis, and A. Hilton. Surface-based structure-from-motion using feature groupings. In Proceedings of the Asian Conference On Computer Visions, pages 699-705, Taipei, Taiwan, Jan. 2000. Springer-Verlag.
- [5] R. Szeliski and P. Torr. Geometrically constrained structure from motion: Points on planes. In Proceedings of the European Workshop on 3D structure from Multiple Images of Large-Scale Environments (SMILE), pages 171-186, Freiburg, June 1998.
- [6] Th. F. Coleman and A. R. Conn, "Nonlinear programming via an exact penalty function: Global analysis", *Mathematica Programming* 24(1982), pp. 137-161
- [7] P. E. Gill, W. Murray, and M. A. Saunders, SNOPT: An SQP algorithm for large-scale constrained optimization, Numerical Analysis Report 97-1, Department of Mathematics, University of California, San Diego, La Jolla, CA, 1997.
- [8] P. T. BOGGS and J. W. TOLLE. Sequential quadratic programming. *Acta Numerica*, 4: 1-- 50, 1995.
- [9] Paul T. Boogs, Anthony J. Kearsley, and Jon W. Toole. A global convergence analysis of an algorithm for large scale nonlinear programming problems. *SIAM Journal on Optimizations*, 9(3):755-778, 1999
- [10] B. Triggs. *Autocalibration and the absolute quadric*. In Proc. Conf. Computer Vision and Pattern Recognition, pages 609-- 614, 1997
- [11] Ma S.D. and Li. L. Ellipsoid reconstruction from three perspective views. In Proc. International Conference on Pattern Recognition, Vienna, 1996
- [12] Ma S.D. and Chen X. Quadric surface reconstruction from its occluding contours. In Proc. International Conference on Pattern Recognition, Israel, 1994
- [13] Quan L. Invariant of a pair of non-coplanar conics in space: Definition, geometric interpretation and computations. In Proc. 5<sup>th</sup> Int'l Conf. On computer vision, pages 926-931, 1995
- [14] Z. Zhang, R. Deriche, O. Faugeras, and Q. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78:87--119, 1995
- [15] A. Hartley, R. Zisserman. *Multiple View Geometry in Computer Vision*. MIT Press, 2000
- [16] O. Faugeras, *Three-dimensional computer vision: a geometric viewpoint*, MIT Press, 1993.
- [17] Luong, Q.-T., Vieville, T., *Canonic Representations for the Geometries of Multiple Projective Views*, ECCV'94, Lecture notes in Computer Science, Vol 800. Ed. Jan Olof Eklund, Springer-Verlag, 1994, pp. 589-599.
- [18] Hartley R. and Strum P. *Triangulation*. In Proc. Conference Computer Analysis of Images and Patterns, Prague, Czech Republic, 1995
- [19] Marc Pollefeys and Luc Van Gool. Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. In Proc. Int. Conf. on Computer Vision, 1998.
- [20] W. Press, S. Teukolski, W. Vetterling, B. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 2nd ed., 1992.
- [21] P. Cignoni, C. Montani, C. Rocchini, and R. Scopigno, *Multiple textures stitching and blending on 3d objects*, *Rendering Techniques '99*, Springer-Verlag Wien, pp. 119--130, 1999.