Image Processing Algorithms For Deep-Space Autonomous Optical Navigation

Shuang Li[1], Ruikun Lu[1], Liu Zhang[2] and Yuming Peng[3]

[1] (*College of Astronautics, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China*)
[2] (*Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China*)
[3] (*Shanghai Institute of Satellite Engineering, Shanghai 200240, China*)
(E-mail: lishuang@nuaa.edu.cn)

As Earth-based radio tracking navigation is severely limited because of communications constraints and low relative navigation accuracy, autonomous optical navigation capabilities are essential for both robotic and manned deep-space exploration missions. Image processing is considered one of the key technologies for autonomous optical navigation to extract high-precision navigation observables from a raw image. New image processing algorithms for deep-space autonomous optical navigation are developed in this paper. First, multiple image pre-processing and the Canny edge detection algorithm are adopted to identify the edges of target celestial bodies and simultaneously remove the potential false edges. Secondly, two new limb profile fitting algorithms are proposed based on the Least Squares method and the Levenberg-Marquardt algorithm, respectively, with the assumption that the perspective projection of a target celestial body on the image plane will form an ellipse. Next, the line-of-sight (LOS) vector from the spacecraft to the centroid of the observed object is obtained. This is taken as the navigation measurement observable and input to the navigation filter algorithm. Finally, the image processing algorithms developed in this paper are validated using both synthetic simulated images and real flight images from the *MESSENGER* mission.

1. INTRODUCTION. Traditional spacecraft navigation has been primarily done by means of range and range-rate measurements from the deep-space network (DSN). The major drawbacks to this approach are the high cost and the low update frequency. Both future robotic and human space exploration missions are expected to rely on autonomous operations more than ever before. Of particular interest here is the

ability of a spacecraft to navigate autonomously independent of contact with Earth-based DSN or similar assets (Owen et al., 2008; Riedel et al., 1997; 2000; 2008; Wang and Huang, 2009). For robotic missions, especially in the approach and landing phases, autonomous navigation is indispensable because Earth-based radio tracking navigation operations are severely limited due to communications constraints and low relative navigation accuracy. During the planetary approach, rendezvous and landing phases, some spacecraft have also relied on optical observations to improve the estimation of the spacecraft's situation with respect to the target planet or moon (Rush and Synnott, 2002; Bhaskaran et al., 2011; Li et al., 2005). For manned missions, autonomous navigation would enable the crew to safely guide the spacecraft's return to Earth in the event of a radio tracking system failure. Autonomous navigation not only helps to reduce the communications demands on ground-based antennas but also helps provide the high-precision relative navigation. The latter is of great importance for landing and fly-by missions and is beyond the capability of the current ground-based tracking navigation (Owen et al., 2008; Riedel et al., 2008; Owen, 2011; Paluszek et al., 2010). Although, in the last two decades, several kinds of autonomous navigation methods and technologies have been proposed, autonomous optical navigation is considered the most feasible solution to the problem of spacecraft navigation without requiring ground-based tracking or status update (Owen, 2011).

The feasibility of optical navigation (OPNAV) has been preliminarily confirmed by many deep-space missions. The *NEAR Shoemaker* mission depended heavily on an optical navigation system, which located many craters on the surface of the asteroid Eros as navigation landmarks and enabled rapid orbit determination around Eros (Owen, 2001). For onboard navigation *Hayabusa* employed wide-angle cameras in conjunction with Light Detecting and Ranging (LIDAR) for measurement of altitude with respect to the asteroid Itokawa (Kominato, 2006). The OPNAV images captured by the *NEAR* and *Hayabusa* spacecraft were sent back to the Earth for image processing. These results were integrated with ground-based tracking data and then a status update was uploaded to the spacecraft. *Deep Space 1*, as part of the NASA New Millennium Program, used optical images of distant asteroids for orbit determination and was the first deep-space mission that used spacecraft autonomous navigation throughout the mission phases (Riedel et al., 1997; 2000; 2008). The *Deep Impact* (DI) mission navigated the impactor spacecraft to successfully hit an illuminated area on the surface of comet Tempel 1, while also performing a fly-by of the main spacecraft to track the impact site (Riedel et al., 2008; Mastrodemos, 2006; Owen, 2006). From *NEAR* to *Deep Impact*, the most promising advance in optical navigation technology has been the migration from ground image processing to onboard processing, which consequently presents a huge challenge to the onboard computational capability.

The ultimate goal of image processing algorithms in optical navigation is to extract the various available navigation observables from raw images taken by a spacecraft, which usually include; the line-of-sight (LOS) vector, apparent diameter and centroid, angle between horizon and reference star, etc. (Owen, 2008; 2011). Among them, the LOS measurement is the most important. With the development of various deep-space flight missions, there have been many academic papers published on the subject of autonomous optical navigation. However, most of them are focused on the system dynamics, navigation measurement modeling and navigation filters; significantly few reported the onboard image processing algorithms and the procedure for the

extraction of navigation observables (Giannatrapani et al., 2011; Ma et al, 2010; Li and Cui, 2008; Sui, 2008; Li et al., 2006; Sui et al., 2007; Thompson et al., 2012; Christian and Lightsey, 2012). Traditional image processing algorithms face various difficulties and cannot be directly applied to an autonomous navigation system due to the low onboard computation capacity and the particularity of the deep-space environment (Owen, 2011; Paluszek et al., 2010). The core requirement of image processing technologies for deep-space autonomous optical navigation lies in extracting navigation measurement information from various optical observations quickly and accurately. With the rapid development of optical sensors, image resolution is increasing exponentially, which usually results in a bigger computation burden and degrades real-time performance of autonomous optical navigation. Therefore, simple and effective image processing algorithms with a high degree of robustness are indispensable for the success of deep-space autonomous optical navigation.

The aim of this paper is to develop suitable image processing algorithms used for extracting navigation observables from raw grayscale images. The basic measurement obtained from this process is the line-of-sight (LOS) vector from the optical sensor to the centroid of the target celestial body, which is then input to a navigation filter in order to estimate the spacecraft status in combination with system dynamics equations. Image processing algorithms developed in this paper place emphasis on how to obtain the LOS measurement from a raw image regardless of other measurement types. This paper is structured as follows. Section 2 briefly presents the optical navigation measurement model and image processing algorithms involved in this paper. Section 3 describes the basic image processing techniques and Canny edge detection algorithm. Section 4 develops two new methodologies for fitting an ellipse to a set of limb data points. The centroid computation formula and the extraction procedure for navigation observables are given in Section 5. In section 6, simulation scenarios are described and results discussed in detail. Finally, Section 7 presents our conclusions and future work.

2. FUNDAMENTALS OF AUTONOMOUS OPTICAL NAVIGATION

2.1 *Optical Navigation Measurement Model.* The lowest order approximation to the optical sensor system is a pinhole camera model, which assumes every point on the target emits a single ray and that each ray maps to a point on the focal plane. A point $P(X,Y,Z)$ is mapped to the imaging plane by the following relationship (Rafael and Richard, 2003):

$$u = \frac{X}{Z}f$$
$$v = \frac{Y}{Z}f$$

(1)

where: u and v are coordinates in the focal plane, f is the focal length of navigation camera. X, Y, Z are defined in the camera-fixed frame, and Z is the distance from the pinhole to the point $P(X,Y,Z)$ along the axis normal to the focal plane. This assumes that the Z-axis of the coordinate frame X, Y, Z is aligned with the boresight of the navigation camera.

The pinhole camera model allows for a simple relationship between a point on the detector plane and the corresponding line-of-sight unit vector. The relationship

between the focal length, a measured point on the focal plane, and the line-of-sight unit vector to the image source is given by (Yi et al., 2004):

$$\mathbf{e}_i^C = \frac{1}{\sqrt{u_i^2 + v_i^2 + f^2}} \begin{bmatrix} -u_i \\ -v_i \\ f \end{bmatrix} \tag{2}$$

where the variable u_i and v_i are the coordinates of the point where the central ray of the i–th observed point source pierces the image plane.

The line-of-sight vector is then rotated from the camera frame to the inertial frame:

$$\mathbf{e}_i^I = T_B^I T_C^B \mathbf{e}_i^C \tag{3}$$

where T_B^I is the transformation matrix from body frame to inertial frame, and T_C^B is the transformation matrix from camera frame to body frame.

Imaging error sources, such as stellar aberration, parallax, stray light, misalignment and diffraction, are not included in the pinhole camera model. In order to get a real math model, both the external error sources and the internal camera error sources must be taken into account (Li and Lu, 2013).

2.2. *Image Processing Procedure*. In this paper, the image processing for autonomous optical navigation is a multi-step procedure. The first step is to simplify the raw image by using image pre-processing technologies to facilitate the subsequent steps. The second step is to identify the edge data points and remove the fake limb points. The third step is to find the limb profile in the image of a celestial object. For this third step, two different ellipse-fitting algorithms will be discussed: (1) the Least Squares based ellipse-fitting algorithm and (2) the Levenberg-Marquardt based ellipse-fitting algorithm. The last step is to compute the centroid according to the ellipse equation of the fitted limb profile. Thereafter, useful navigation measurements can be extracted. Figure 1 depicts a sketch of the image processing procedure; the details of these four steps are provided in the following subsections.

3. EDGES IDENTIFICATION ALGORITHM

3.1. *Image Pre-Processing*. The aim of image pre-processing technology is to make an image more suitable for subsequent processing. This is achieved by enhancing areas of interest and suppressing uninteresting regions without adding additional information. The method of pre-processing includes, among others, threshold segmentation, image smoothing, image sharpening and image correction technology (Rafael and Richard, 2003). In this paper they are directly applied to raw grayscale images taken by an optical navigation camera. Threshold segmentation operations can be used to simplify the raw image into a binary image and improve the running speed of subsequent image processing. Image smoothing operations can suppress imaging noise and other small fluctuations in the image which could cause blurring to sharp edges that bear important information about the image. On the other hand, image sharpening operations strengthen the edge features. Individually, each of these image processing techniques is rather simple. When properly combined, however, they can be a powerful tool and facilitate the following image processing. Figures 2 and 3, respectively, show the raw gray image of Mercury and the segmentation image with the threshold value $\tau = 0{\cdot}1$.
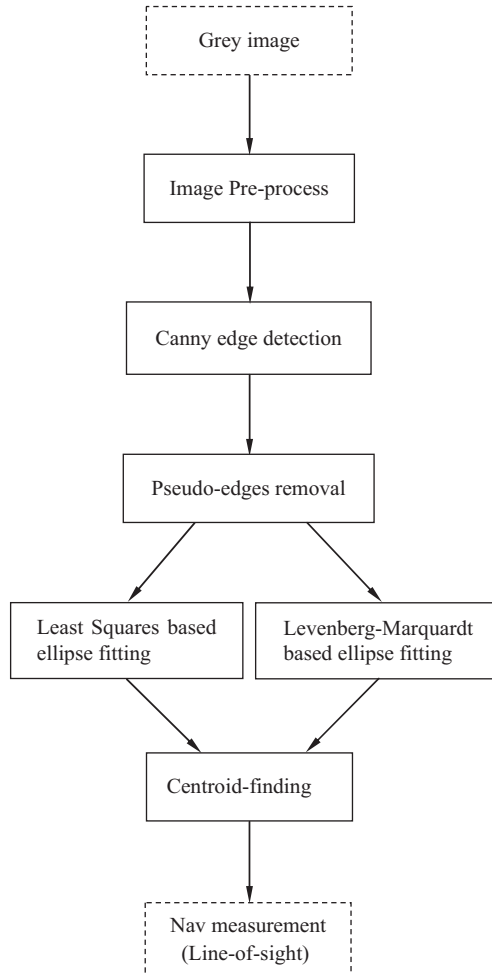
Figure 1. Sketch of the image processing procedure.

Figure 2. Raw gray image of Mercury.

3.2   Edge Detection

3.2.1.   *Canny Edge Detection*.   Edge points are defined as the points where the gradient magnitude assumes a local maximum in the gradient direction. An ideal edge

Figure 3. Image after threshold segmentation ($\tau = 0 \cdot 1$).

detection algorithm is required to meet the following criteria: (1) noise should be effectively suppressed and (2) edge points should be accurately positioned. However, noise suppression and accurate positioning of edge points cannot be easily satisfied simultaneously. That is, the smoothing operation in edge detection can effectively remove the imaging noise; at the same time, it also weakens the main edge information of the original image. On the other hand, improving the sensitivity to the edge leads to the same adverse effects for noise suppression.

The first-order derivative of the Gaussian function is chosen here as a linear operator, which provides the best compromise between removing noise interference and accurate positioning of edge points (Canny, 1986). In order to perform image smoothing using a Gaussian filter, we first introduce the Gaussian function as follows:

$$G(i, j) = e^{\left(-\frac{i^2 + j^2}{2\sigma^2} \right)} \tag{4}$$

First, a convolution operation is implemented between the Gaussian function and the image intensity values:

$$S(i, j) = G(i, j) * I(i, j) \tag{5}$$

where: * denotes the convolution operator and $I(i,j)$ stands for the intensity values of the raw image.

Next, the first-order derivative finite difference of the image convolution in the horizontal direction ($P[i,j]$) and the vertical direction ($Q[i,j]$) can be obtained:

$$P[i, j] \approx (S[i, j + 1] - S[i, j] + S[i + 1, j + 1] - S[i + 1, j])/2 \tag{6}$$

$$Q[i, j] \approx (S[i, j] - S[i + 1, j] + S[i, j + 1] - S[i + 1, j + 1])/2 \tag{7}$$

Then, the edge gradient magnitude $M[i,j]$ and direction $\theta[i,j]$ can be determined as follows:

$$M[i, j] = \sqrt{P[i, j]^2 + Q[i, j]^2} \tag{8}$$

$$\theta[i, j] = \arctan(Q[i, j]/P[i, j]) \tag{9}$$

As the gradient magnitude of an edge point $M[i,j]$ fulfills the condition of being a local maximum in the gradient direction, we use the Non-Maxima Suppression (NMS) technique to determine whether these local maximum data points belong to the edge points. We establish whether each pixel point $[i,j]$ is a local maximum of the
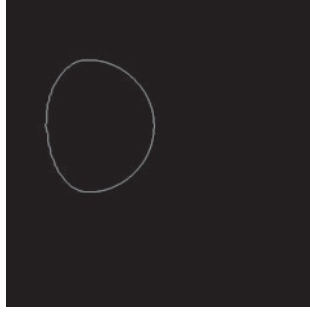
Figure 4. Canny edge detection result.

gradient magnitude among its adjacent pixels in the gradient direction and, if not, the gradient magnitude $M[i,j]$ is set to zero. This screening process is sometimes referred to as edge thinning.

The final step of the Canny edge detection is detecting and connecting edge points using the double-threshold segmentation algorithm. Two threshold values are defined τ_1, τ_2 ($\tau_1 > \tau_2$), and the magnitude of all the data points are set to zero when they are lower than the prescribed thresholds. Then, we can acquire two binary segmentation edge images T_1 and T_2, which correspond to the two different threshold values τ_1 and τ_2, respectively. If the threshold value is set too low, the segmentation result will be increasingly susceptible to imaging noise and will include some false edges. Conversely, segmentation with a high threshold may miss subtle edges and result in a fragmented edge profile. In order to overcome the problem of finding a suitable threshold, we first link the edge points obtained from higher threshold segmentation, and then the breakpoints are supplemented by edge points from lower threshold segmentation using the eight neighborhood search algorithm. Figure 4 depicts the result from Canny edge detection, which, obviously, includes the pseudo-edge. The removal of these pseudo-edges is described in the following section.

3.2.2. *Pseudo-edge Removal*. Taking into account the influence of the sun's azimuth and camera viewing angles, edge detection results, such as those from section 3.2.1., commonly contain some pseudo-edges due to the presence of backlight shaded areas. Pseudo-edges usually occur at the junction between the sunlight face and backlight face (Thompson et al., 2012). In order to extract navigation information correctly, real edges must be distinguished from false edges.

False edges can be distinguished by the calculation of the angle between the lighting direction and the gradient direction of edges. The distribution of gradient direction is depicted in Figure 5. In fact, the angle between the gradient vectors of a planet's real edges and the sun lighting direction is less than 90°. Here, we assume that the sun lighting direction is provided by a sun sensor, though it can be obtained from a sophisticated image algorithm (Rafael and Richard, 2003). Given the planet's edge gradient vector $\boldsymbol{g}$ and the sun lighting direction $\boldsymbol{n}$, the following relation should be satisfied:

$$\frac{\boldsymbol{g} \cdot \boldsymbol{n}}{|\boldsymbol{g}||\boldsymbol{n}|} > 0 \qquad (10)$$

The Inequality constraint (10) is the important criterion for identifying and eliminating the fake edges at the interface between the light and dark sides.
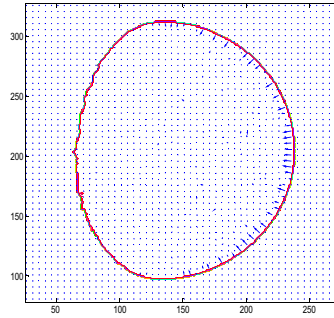
Figure 5. Edge gradient direction distribution.

Figure 6. Edge detection result without pseudo-edge.

Figure 6 shows that the real edges can be successfully identified after removing the pseudo-edges.

4. LIMB PROFILE FITTING. The perspective projection of a sphere or ellipsoid will form an ellipse on the image plane. That is normally the case if the objective is to locate a planet or moon in an image. Once candidate edge points are found, an ellipse must be fitted to these data points. Most early solutions to the problem of ellipse fitting were based on clustering and voting techniques, such as the Hough transform (Zhang, 1997; Theodoridis and Koutroumbas, 2009). These techniques are naturally robust to noise and outliers; however, they require large amounts of memory and time-consuming computations, which leads to these algorithms being difficult to apply in an onboard computer.

4.1. *Least Squares-Based Ellipse Fitting.* All conic sections can be described by the following implicit quadratic equation (Brannan et al., 1999):

$$F(x_i, y_i) = Ax_i^2 + Bx_iy_i + Cy_i^2 + Dx_i + Ey_i + F = 0 \qquad (11)$$

where: $[x_i, y_i]$ denotes the data point lying on the conic section. A detailed discussion of the properties of this implicit equation is given in (Brannan et al., 1999) and it is well known that the conic will be an ellipse if the coefficients of equation (11) satisfy:

$$4AC - B^2 > 0 \qquad (12)$$

In general, the data points from the edge detection algorithm developed in Section 3 do not lie exactly on an ellipse and $F(x_i, y_i) \neq 0$ in the presence of imaging noise.

Therefore, the following optimization problem based on the square of the model fitting residuals is proposed:

$$S(\boldsymbol{a}) = \sum_{i=1}^{n} [f(\boldsymbol{a}, \boldsymbol{x}_i)]^2 \tag{13}$$

where: $\boldsymbol{a} = [A, B, C, D, E, F]^T$ are the ellipse equation parameters to be estimated, $f(\boldsymbol{a}, \boldsymbol{x}_i) = Ax_i^2 + Bx_iy_i + Cy_i^2 + Dx_i + Ey_i + F$ stands for the estimated value from the ith observation point $\boldsymbol{x}_i = (x_i, y_i)$, whose theoretical value should be zero, and n is the number of observation points.

Here, the ellipse fitting problem can be described as an optimization problem with the objective function to be minimized as follows:

$$\min\{S(\boldsymbol{a})\} = \min\left\{\sum_{i=0}^{m} [f(\boldsymbol{a}, \boldsymbol{x}_i)]^2\right\} \tag{14}$$

According to extreme principle, the minimum value of $S(\boldsymbol{a})$ can be obtained when the value of first order partial derivatives is zero:

$$\frac{\partial S}{\partial A} = \frac{\partial S}{\partial B} = \frac{\partial S}{\partial C} = \frac{\partial S}{\partial D} = \frac{\partial S}{\partial E} = \frac{\partial S}{\partial F} = 0 \tag{15}$$

$$\frac{\partial f}{\partial A} = \frac{\partial f}{\partial B} = \frac{\partial f}{\partial C} = \frac{\partial f}{\partial D} = \frac{\partial f}{\partial E} = \frac{\partial f}{\partial F} = 0 \tag{16}$$

Then the parameters of the ellipse can be determined when the objective function has a minimum:

$$\sum_{i=1}^{n} [f(\boldsymbol{a}^*, \boldsymbol{x}_i)]^2 = \sum_{i=1}^{n} (A^*x_i^2 + B^*x_iy_i + C^*y_i^2 + D^*x_i + E^*y_i + F^*)^2 \tag{17}$$

The first order partial derivatives $\dfrac{\partial f}{\partial A}$ can be rewritten as follows:

$$\frac{\partial f}{\partial A} = \frac{\partial\left(\sum_{i=1}^{n} [f(\boldsymbol{a}, \boldsymbol{x}_i)]^2\right)}{\partial A} = \sum_{i=1}^{n} \frac{\partial [f(\boldsymbol{a}, \boldsymbol{x}_i)]^2}{\partial A}$$

$$= \sum_{i=1}^{n} 2(Ax_i^2 + Bx_iy_i + Cy_i^2 + Dx_i + Ey_i + F) \cdot x_i^2 = 0 \tag{18}$$

Similarly, we have

$$\frac{\partial f}{\partial B} = \sum_{i=1}^{n} 2(Ax_i^2 + Bx_iy_i + Cy_i^2 + Dx_i + Ey_i + F) \cdot x_iy_i = 0 \tag{19}$$

$$\frac{\partial f}{\partial C} = \sum_{i=1}^{n} 2(Ax_i^2 + Bx_iy_i + Cy_i^2 + Dx_i + Ey_i + F) \cdot y_i^2 = 0 \tag{20}$$

$$\frac{\partial f}{\partial D} = \sum_{i=1}^{n} 2(Ax_i^2 + Bx_iy_i + Cy_i^2 + Dx_i + Ey_i + F) \cdot x_i = 0 \tag{21}$$

$$\frac{\partial f}{\partial E} = \sum_{i=1}^{n} 2(Ax_i^2 + Bx_iy_i + Cy_i^2 + Dx_i + Ey_i + F) \cdot y_i = 0 \tag{22}$$

$$\frac{\partial f}{\partial F} = \sum_{i=1}^{n} 2(Ax_i^2 + Bx_iy_i + Cy_i^2 + Dx_i + Ey_i + F) = 0 \tag{23}$$

Linear equations (18)–(23) can be rewritten as follows:

$$\begin{bmatrix} \sum_{i=1}^{n} x_i^4 & \sum_{i=1}^{n} x_i^3y_i & \sum_{i=1}^{n} x_i^2y_i^2 & \sum_{i=1}^{n} x_i^3 & \sum_{i=1}^{n} x_i^2y_i & \sum_{i=1}^{n} x_i^2 \\ \sum_{i=1}^{n} x_i^3y_i & \sum_{i=1}^{n} x_i^2y_i^2 & \sum_{i=1}^{n} x_iy_i^3 & \sum_{i=1}^{n} x_i^2y_i & \sum_{i=1}^{n} x_iy_i^2 & \sum_{i=1}^{n} x_iy_i \\ \sum_{i=1}^{n} x_i^2y_i^2 & \sum_{i=1}^{n} x_iy_i^3 & \sum_{i=1}^{n} y_i^4 & \sum_{i=1}^{n} x_iy_i^2 & \sum_{i=1}^{n} y_i^3 & \sum_{i=1}^{n} y_i^2 \\ \sum_{i=1}^{n} x_i^3 & \sum_{i=1}^{n} x_i^2y_i & \sum_{i=1}^{n} x_iy_i^2 & \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_iy_i & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i^2y_i & \sum_{i=1}^{n} x_iy_i^2 & \sum_{i=1}^{n} y_i^3 & \sum_{i=1}^{n} x_iy_i & \sum_{i=1}^{n} y_i^2 & \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_iy_i & \sum_{i=1}^{n} y_i^2 & \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} y_i & \sum_{i=1}^{n} 1 \end{bmatrix} \cdot \begin{bmatrix} A \\ B \\ C \\ D \\ E \\ F \end{bmatrix} = 0 \tag{24}$$

Defining the coefficient matrix $M = \begin{bmatrix} x_1^2 & x_1y_1 & y_1^2 & x_1 & y_1 & 1 \\ x_2^2 & x_2y_2 & y_2^2 & x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n^2 & x_ny_n & y_n^2 & x_n & y_n & 1 \end{bmatrix}$, then these

linear equations can be simplified into the following form:

$$M^TMa = 0 \tag{25}$$

Generally speaking, for nonlinear equations, it is rather difficult, and sometimes even impossible, to directly find exact analytic solutions. However, an approximate solution can be obtained instead of the analytic solution by using numeric iteration methods such as Newton-Raphson iteration methods (Crassidis and Junkins, 2011). It should be noted that the equations simplified at (25) are linear equations, and, normally, the analytic solution can be easily obtained without complex iteration steps. Here, the Gaussian elimination method is utilized to solve the ellipse parameters A,B, C,D,E,F. It should be pointed out that the validity of these fitting parameters a should be checked according to the ellipse inequality constraint $4AC - B^2 > 0$.

4.2. *Levenberg-Marquardt-Based Ellipse Fitting*. The Levenberg-Marquardt (LM) algorithm, also known as the damped least-squares method, provides a numerical solution to the problem of minimizing a nonlinear function. The Levenberg-Marquardt algorithm can effectively overcome: the difficulties of the nonlinear least squares approach, when an accurate initial estimate is not given; and the slow convergence problems of the steepest descent method, when the solution is close to minimizing the cost function (Crassidis and Junkins, 2011; Levenberg, 1944). As with other numeric minimization algorithms, the Levenberg-Marquardt algorithm is an iterative procedure.

As before, the goal is to optimize the parameters a of the model curve $f(a,x_i)$ so that the sum of the squares of the deviations becomes minimal:

$$\min\{S(a)\} = \min\left\{\sum_{i=1}^{n}\left[f(a, x_i)\right]^2\right\} \tag{26}$$

To start a minimization, an initial guess for the parameter vector $a=[A,B,C,D,E,F]^T$ is required. In each iteration step, the parameter vector a is then replaced by a new estimate $a+\delta$. To determine the value of δ, the functions $f(a+\delta,x_i)$ are approximated by their linearization:

$$f(a + \delta, x_i) \approx f(a, x_i) + J_i\delta \tag{27}$$

where: $J_i = \dfrac{\partial f\,(a, x_i)}{\partial a}$ is the gradient of f with respect to a.

At the minimum of the sum of squares $S(a)$, the gradient of S with respect to δ should be zero. The above first-order approximation of $f(a+\delta,x_i)$ gives:

$$S(a + \delta) \approx \sum_{i=1}^{n}\left[f\,(a, x_i) + J_i\delta\right]^2 \tag{28}$$

or in vector notation:

$$S(a + \delta) \approx \left\|f(a) + J\delta\right\|^2 \tag{29}$$

Taking the derivative with respect to δ and setting it to zero $\dfrac{\partial S}{\partial \delta} = 0$ gives:

$$(J^TJ)\delta = J^Tf(a) \tag{30}$$

where J is the Jacobian matrix whose i-th row equals J_i, and where $f(a)$ is the vector with i-th component $f(a,x_i)$.

In order to control the convergence rate and accuracy, a damping factor λ is introduced here. Equation (30) can be transformed as:

$$(J^TJ + \lambda I)\delta = J^Tf(a) \tag{31}$$

To avoid the defect of slow convergence in the direction of a small gradient, the identity matrix I is replaced with the diagonal matrix consisting of the diagonal elements $diag(J^TJ)$ (Crassidis and Junkins, 2011; Marquardt, 1963).

$$(J^TJ + \lambda diag(J^TJ))\delta = J^Tf(a) \tag{32}$$

All that is left to do is to solve the above linear equations, and then get the value of δ and the new ellipse parameter estimated value $a+\delta$. With repeatedly iterated calculations, we can obtain the optimal estimate of $a_{final}=(A,B,C,D,E,F)^T$. If these parameters meet the constraint condition $4AC-B^2>0$, then we can conclude that $a_{final}=(A,B,C,D,E,F)^T$ is the optimal estimated ellipse parameters.

5. CENTROID EXTRACTION. The equation of the planet profile can thus be fitted by using the aforementioned methods of edge detection and ellipse fitting. Then the centre point of an ellipse can be assumed to be the centroid of a planet. Compared with a method directly extracting the centre-of-brightness (COB), the centroid approach achieves a better accuracy of navigation measurements with various influences, such as the shadows in an image of a planet which depend on the

illumination and observation angles, and the interference of craters, texture, and atmosphere when photographed from a close range.

The centre of an ellipse can be obtained by the transformation from the implicit coefficients to standard ellipse parameters, which is a well-known result from classical geometry and is given by (Rafael and Richard, 2003; Yi et al., 2004):

$$x_0 = \frac{2CD - BE}{B^2 - 4AC} \tag{33}$$

$$y_0 = \frac{2AE - BD}{B^2 - 4AC} \tag{34}$$

$$a = \sqrt{\frac{2\left[AE^2 + CD^2 - BDE + F\left(B^2 - 4AC\right)\right]}{\left(B^2 - 4AC\right)\left[\sqrt{(A-C)^2 + B^2} - A - C\right]}} \tag{35}$$

$$b = \sqrt{\frac{2\left[AE^2 + CD^2 - BDE + F\left(B^2 - 4AC\right)\right]}{\left(B^2 - 4AC\right)\left[-\sqrt{(A-C)^2 + B^2} - A - C\right]}} \tag{36}$$

$$\phi = \begin{cases} 0 & B = 0 \ and \ A < C \\ \dfrac{\pi}{2} & B = 0 \ and \ A > C \\ \dfrac{1}{2}\cot^{-1}\left(\dfrac{A-C}{B}\right) & B \neq 0 \ and \ A < C \\ \dfrac{\pi}{2} + \dfrac{1}{2}\cot^{-1}\left(\dfrac{A-C}{B}\right) & B \neq 0 \ and \ A > C \end{cases} \tag{37}$$

where: (x_0, y_0) is the coordinate of the ellipse centre, a and b stand for the semi-major axis and the semi-minor axis, and ϕ is the inclined angle from the x-axis to the ellipse major axis.

The ellipse parameters obtained in Equations (33)–(37) include the coordinates of the ellipse centre, as well as the dimensions and orientation of the ellipse in the image plane. With these parameters we can easily compute the line-of-sight vector from the spacecraft to the centroid of a target celestial body according to Equations (2) and (3).

6. SIMULATION RESULTS AND ANALYSIS. In order to test the effectiveness and robustness of the image processing algorithm developed in this paper, real images obtained from deep-space flight missions and synthetic simulated images were utilized to test the performance of the image processing algorithm.

The real raw images used here come from the *MESSENGER* mission, which include a Mercury image (Product ID EN0108892844M) and a Venus image (Product ID N0089716371M) taken by the *MESSENGER* spacecraft's narrow angle camera on 15 January 2008 and 6 June 2007, respectively (Hash, 2008). In order to find the potential edge profiles of the target celestial bodies in the image, the pre-processing and edge identification algorithms described in Section 3 were firstly applied to both the Mercury and Venus images. Then, the Least Squares-based ellipse fitting and Levenberg-Marquardt-based ellipse fitting algorithms were respectively implemented to fit the planet's limb profile. The simulation results are shown in Figures 7 and 8 respectively.
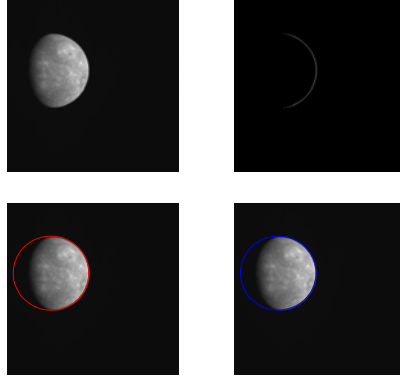
Figure 7. Image processing algorithm applied to real image containing Mercury. (Taken by *MESSENGER* on 15 January 2008).
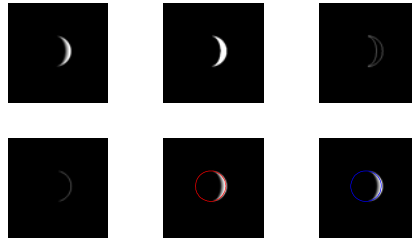

Figure 8. Image processing algorithm applied to real image containing Venus. (Taken by *MESSENGER* on 6 June 2007).

Note that Figures 7 and 8 also contain two fitted ellipses, from the image processing algorithms developed in Section 4, superimposed in red or blue outlines on top of the raw Mercury and Venus images from the *MESSENGER* mission. The red outlines indicate the fitting ellipses obtained by the Least Squares-based ellipse fitting algorithm, and the blue outlines indicate the fitting ellipses obtained by the Levenberg-Marquardt-based ellipse fitting algorithm. It can be seen in these plots that both ellipse fitting algorithms accurately acquire the limb profiles of Mercury and Venus. It can also be observed that the pseudo-edge data points in the interior of the planet limb are all eliminated by the threshold segmentation operation, which is adopted before the Canny edge detection and aims at simplifying the raw image. At the same time, the fake edges produced at the interface between the light and dark areas can be effectively removed by considering the angle between the planet's edge gradient vectors and the lighting direction, which greatly contributes to the success of the planet profile fitting algorithm.

Synthetic images were also used in our simulation analysis to test the robustness of image processing algorithms developed in this paper. When the solar azimuth angle is provided by an onboard attitude sensor, synthetic images of a planet or moon can be generated using the suitable optical imaging theories and models. The principle and process of how to produce real simulated images including various imaging errors will be described in detail in a companion paper (Li and Lu, 2013) and, thus, are not detailed here. Simulated planet images with various surface textures and solar azimuth
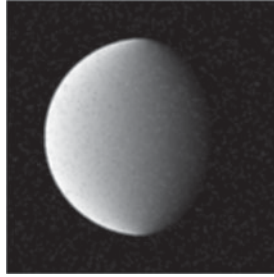
Figure 9. Simulated image of a planet with flat surface.

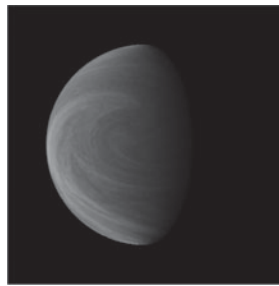Figure 10. Simulated image of a planet with crater texture.

Figure 11. Simulated image of a planet with atmosphere.

angles can be found in Figures 9 through 11. Figure 9 shows the simulated flat surface image of a distant planet with a 60 degrees solar azimuth angle. Figure 10 depicts the simulated image of a planet covered with crater texture at 45 degrees solar azimuth angle. A synthetic image of a planet covered with atmosphere is represented in Figure 11 at an illumination phase angle of 30 degrees.

Then, the complete image processing algorithms developed in this paper, including both the Least Squares and the Levenberg-Marquardt ellipse fitting algorithms, are applied to the three simulated images of a planet with different surface textures and sun azimuth angles, and the results are plotted in Figure 12 for comparison. It should be noted that the image processing algorithms can accurately obtain the limb profiles of the planets with different textures and lighting conditions. The effectiveness and robustness of the image processing algorithm developed in this paper can, therefore, be preliminarily confirmed.
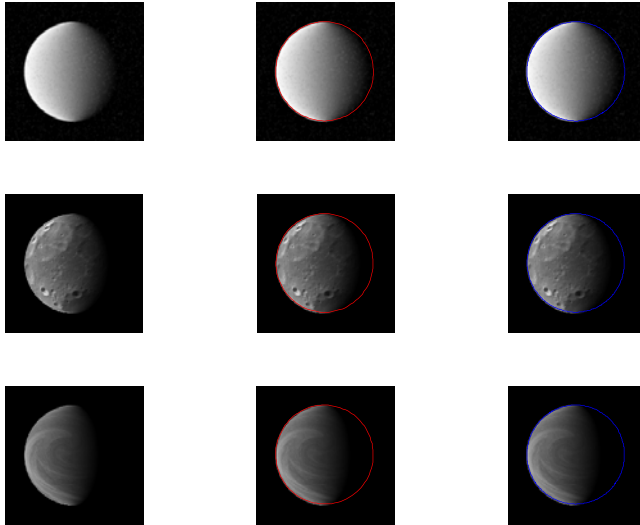
Figure 12. Image processing algorithm applied to simulated images of a distant planet.

To further verify the performance of the ellipse fitting algorithms described in Section 4, simulation and analysis using different fitting algorithms and different simulated images are performed for comparison. It is emphasized that all the simulations adopted the same image processing algorithm developed in this paper except for the ellipse fitting portion. The method of improved ellipse fitting with direct Least-Squares estimation presented by Christian and Lightsey (2012) is used as a reference compared with the two ellipse fitting algorithms described in Section 4. All three simulated images (covered with a flat surface, crater texture and atmosphere) are included in our simulation and test. As the ellipse parameters of the limb profile in the simulated images can be accurately known in advance, the fitting errors can be easily obtained after the profile ellipses are fitted. The fitting errors of the three ellipse fitting algorithms are shown in Tables 1 through 3, respectively. It can be seen from these tables that the maximum fitting error with the method of improved ellipse fitting with direct least-squares estimation is larger than three pixels, while the maximum errors with both the Least Squares-based ellipse fitting and the Levenberg-Marquardt-based ellipse fitting algorithms, are less than three pixels. It should also be noted that the inclined angle error in Table 1 is equal to zero, which indicates that the Least Squares-based ellipse fitting algorithm is insensitive to errors in the ellipse orientation angle.

Tables 1 to Table 3 set out the fitting errors of three ellipse fitting algorithms, but it is still difficult to make useful comparisons of their performance from just these tables. When the relevant parameters of the camera system are given, we can easily convert the ellipse fitting error into an error on the observed line-of-sight vector, which is directly related to the navigation accuracy and very suitable to be used to determine the performance of the three ellipse fitting algorithms. Assume the focal length of the camera $f = 200$ mm, the angle of view field $\theta = 3.5°$, scale factor $K_x = K_y = 83.8$ *pixel/mm*, then the observation errors of the line-of-sight vector from the camera to centre of the planet can be found in Table 4.

Table 1. Fitting error of Least Squares-based ellipse fitting algorithm.

Fitting parameter errors	Simulated image I (flat surface)	Simulated image II (crater texture)	Simulated image III (atmosphere)
Centre coordinate of x_0 (pixel)	− 0·0038	0·0144	0·0133
Centre coordinate of y_0 (pixel)	− 2·4686	− 2·7940	− 2·6894
Semi-major axis a (pixel)	− 0·4945	− 0·5499	− 0·5370
Semi-minor axis b (pixel)	− 2·6271	**− 2·9920**	− 2·8405
Inclined angle ϕ (rad)	0	0	0

Table 2. Fitting error of improved ellipse fitting with direct least-squares estimation.

Fitting parameter errors	Simulated image I (flat surface)	Simulated image II (crater texture)	Simulated image III (atmosphere)
Centre coordinate of x_0 (pixel)	− 0·0080	− 0·0431	0·0506
Centre coordinate of y_0 (pixel)	− 2·2533	− 3·0431	− 2·8717
Semi-major axis a (pixel)	− 0·4358	− 0·6462	− 0·6073
Semi-minor axis b (pixel)	− 2·3039	**− 3·2245**	− 3·0436
Inclined angle ϕ (rad)	− 0·1020	0·1382	0·1721

Table 3. Fitting error of Levenberg-Marquardt based ellipse fitting algorithm.

Fitting parameter errors	Simulated image I (flat surface)	Simulated image II (crater texture)	Simulated image III (atmosphere)
Centre coordinate of x_0 (pixel)	− 0·0257	0·0126	0·0085
Centre coordinate of y_0 (pixel)	− 2·7108	− 2·6403	− 2·5615
Semi-major axis a (pixel)	− 0·5687	− 0·5494	− 0·5269
Semi-minor axis b (pixel)	**− 2·9047**	− 2·7969	− 2·7141
Inclined angle ϕ (rad)	1·4833	0·9103	− 0·7252

Table 4. Error of the line-of-sight vector.

Simulated image	Error of the line-of-sight vector (rad)		
	Least Squares based ellipse fitting algorithm	Improved ellipse fitting with direct Least-Squares estimation	Levenberg Marquardt based ellipse fitting algorithm
Simulated image I (flat surface)	$1·4729 \times 10^{-4}$	$\mathbf{1·3445 \times 10^{-4}}$	$1·6175 \times 10^{-4}$
Simulated image II (crater texture)	$1·6671 \times 10^{-4}$	$1·8159 \times 10^{-4}$	$\mathbf{1·5754 \times 10^{-4}}$
Simulated image III (atmosphere)	$1·6047 \times 10^{-4}$	$1·7137 \times 10^{-4}$	$\mathbf{1·5283 \times 10^{-4}}$

It can be seen from Tables 1 to 4 that, when compared with the reference algorithm presented in Christian and Lightsey (2012), that the two methods proposed in this paper lead to a better fitting accuracy. The fitting errors of ellipse centre, semi-major and semi-minor axes are less than three pixels, and the error of the line-of-sight vector

to the object centre is less than $1 \cdot 67 \times 10^{-4}$ rad, which can meet the requirements of deep-space autonomous navigation. Given the illustrated performance of these two algorithms in the different simulation scenarios, we can conclude that the Least Squares-based ellipse fitting algorithm works well in the case of good illumination conditions and low background noise, and the Levenberg-Marquardt-based ellipse fitting algorithm will deliver a better accuracy of profile fitting when visibility is poor with much noise. An interesting result here is that the improved ellipse fitting with the direct Least-Squares estimation algorithm works better than our proposed algorithms only with the simulated images covered by a flat texture. When simulated images covered by a crater texture or atmosphere are taken into account, both the Least Squares-based ellipse fitting and the Levenberg-Marquardt-based ellipse fitting algorithms outperform the Improved Ellipse Fitting with Direct Least-Squares Estimation algorithm, and the Levenberg-Marquardt-based ellipse fitting algorithm is superior to the other two. It is noteworthy that a simulated image covered by a flat surface texture presents almost no imaging noise, while simulated images with a crater texture or atmosphere usually include a large number of imaging noises. So, a conclusion can be drawn that the Least Squares-based ellipse fitting and the Improved Ellipse Fitting with direct Least-Squares estimation algorithms work well in a low background noise environment; in contrast, the Levenberg-Marquardt-based ellipse fitting algorithm performs better, and is more reliable, in a high noise environment. The main reason why both the algorithms developed in this paper outperform the reference algorithm in Christian and Lightsey (2012) in high noise cases is the fact that the latter depends heavily on the location distribution of the edge data points and thus is also more sensitive to the presence of outliers. It can be seen from Table 4 that improved ellipse fitting with direct least-squares estimation performs best among all the tests in the case of simulated image covered by a flat surface. In this situation, there are few outliers and the quality of edge characteristics is fairly good. In fact, various factors and limitations in real flight missions, such as imaging noise and resolution, result in the real flight image being covered by complicated textures and noise, which inevitably gives rise to a lot of outliers.

7. CONCLUSION. The capability of autonomous optical navigation (OPNAV) is crucial for future robotics and manned deep-space exploration missions. New image processing algorithms, in particular Least Squares-based ellipse fitting and Levenberg-Marquardt-based ellipse fitting algorithms, are developed in this paper to meet the requirements of deep-space autonomous OPNAV. Both simulated images and real flight images from the *MESSENGER* mission are utilized to test the performance of the image processing algorithm developed in this paper. Simulation results show that the two methods proposed can accurately extract the navigation observables from the raw images. The maximum ellipse fitting errors are less than three pixels, and the error of the line-of-sight vector to the object centre is less than $1 \cdot 67 \times 10^{-4}$ rad, which can meet the requirements of deep-space autonomous navigation.

Future work will focus on an end-to-end assessment of the performance of a complete autonomous OPNAV algorithm, which will systematically incorporate the image processing algorithm developed in this paper with a representative dynamics model and navigation filter algorithm.

ACKNOWLEDGEMENTS

REFERENCES

Brannan, D., Esplen, M. and Gray, J. *Geometry*. (1999). Cambridge University Press, Cambridge, UK.

Bhaskaran, S., et al. (2011). Small body landings using autonomous onboard optical navigation. *Journal of the Astronautical Sciences*, **58**(3), 409–427.

Canny, J. (1986). A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **8**(6), 679–698.

Christian, J. A. and Lightsey, E. G. (2012). Onboard Image-Processing Algorithm for a Spacecraft Optical Navigation Sensor System. *Journal of Spacecraft and Rockets*, **49**(2), 337–352.

Crassidis, J. L., and Junkins, J. L. (2011). *Optimal Estimation of Dynamic Systems*. Chapman & Hall/CRC Press, 2nd revised edition.

Giannatrapani, A., et al. (2011). Comparison of EKF and UKF for Spacecraft Localization via Angle Measurements. *IEEE Transactions on Aerospace and Electronic Systems*. **47**(1), 75–84.

Hash, C. (2008). *MESSENGER MDIS EXPERIMENT (EDR) DATA E/V/H V1.0. NASA Planetary Data System*.

Kominato, T., et al. (2006). *Optical hybrid navigation and station keeping around Itokawa*. AIAA/AAS Astrodynamics Specialist Conference, 2, 1352–1366.

Levenberg, K. (1944). A Method for the Solution of Certain Nonlinear Problems in Least Squares. *Quarterly of Applied Mathematics*, **2**, 164–168.

Li, S. and Lu, R. K. (2013). Generating method of synthetic planet images for optical navigation validation. *Advances in Space Research*, submitted for publication.

Li, S. and Cui, P. Y. (2008). Landmark tracking based autonomous navigation schemes for landing spacecraft on asteroids, *Acta Astronautica*, **62**(6–7), 391–403.

Li, S., Cui, P. Y. and Cui, H. T. (2006). Autonomous navigation and guidance for landing on asteroids. *Aerospace Science and Technology*, **10**(3), 239–247.

Li, S., Cui, H. T. and Cui, P. Y. (2005). Autonomous optical navigation for landing on asteroids. *Aircraft Engineering and Aerospace Technology*, **77**(4), 317–323.

Ma, L. C., Liu, Z. Z. and Meng, X. Y. (2010). Autonomous optical navigation based on adaptive SR-UKF for deep space probes. *Proceedings of the 29th Chinese Control Conference*, 321–325.

Marquardt, D. (1963). An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *SIAM Journal on Applied Mathematics*, **11** (2), 431–441.

Mastrodemos, N., et al. (2006). Autonomous Navigation for the Deep Impact. *Advances in the Astronautical Sciences*, **124**(2), 1251–1271.

Owen Jr, William, M. (2011). Methods of Optical Navigation. *Advances in the Astronautical Sciences*, **140**, 1635–1653.

Owen, W. M. Jr., et al. (2008). A brief history of optical navigation at JPL. *Advances in the Astronautical Sciences*, **131**, 329–348.

Owen, W. M. Jr., et al. (2006). Optical navigation for deep impact. *Advances in the Astronautical Sciences*, **124**(2), 1231–1250.

Owen, W. M. Jr., et al. (2001). *NEAR Optical Navigation at Eros*. AAS/AIAA Astrodynamics Specialist Conference, Quebec, Canada.

Paluszek, M. A., Mueller, J. B. and Littman, M. G. (2010). *Optical Navigation System*. AIAA Infotech Meeting at Aerospace.

Rush, B., Bhaskaran, S. and Synnott, S. P. (2002). Improving Mars approach navigation using optical data. *Advances in the Astronautical Sciences*, **109**(2), 1651–1660.

Rafael, C. G. and Richard, E. W. (2003). *Digital Image Processing*. Second Edition, Prentice Hall.

Riedel, J. (Ed.), et al. (2008). *Configuring the deep impact AutoNav system for lunar, comet and Mars landing*. AIAA/AAS Astrodynamics Specialist Conference and Exhibit.

Riedel, J. E., Bhaskaran, S. and Synnott, S. P., et al. (2000). Navigation for the New Millennium: Autonomous Navigation for Deep Space 1, *Deep Space 1 Technology Validation Report—Autonomous Optical Navigation (AutoNav). JPL Technology Report*, 48–65.

Riedel, J. E., et al. (1997). Navigation for the new millennium: Autonomous navigation for Deep Space 1. European Space Agency (Special Publication) ESA SP, SP-403, 303–320.

Sui, S. L., Yao, W. L. and Zhao, X. W. (2008). Spacecraft autonomous optical navigation based on the wavelet-UPF. *Systems Engineering and Electronics*, **30**(8), 1519–1522.

Sui, S. L., et al. (2007). Improved UKF and SIFT based on information fusion method in autonomous optical navigation. *Proceedings of the 26th Chinese Control Conference*, 103–107.

Thompson., D. R., et al. (2012). Image Processing On-Board Spacecraft for Autonomous Plume Detection. *Planetary and Space Science*, **62**(1), 153–159.

Theodoridis, S. and Koutroumbas, K. (2009). *Pattern Recognition*. Elsevier Science.

Wang, D. Y., Huang, X. Y. (2009). Survey of Autonomous Navigation and Control for Deep Space Exploration. *Space Technology and Application*, **3**, 6–13.

Yi, M., Jana, K. and Stefano, S. (2004). *An Invitation to 3-D Vision from Images to Geometric Models*. Springer, New York, NY.

Zhang, Z. (1997). Parameter Estimation Techniques: A Tutorial with Application to Conic Fitting. *Image and Vision Computing*, **15**(1), 59–76.