



**Member :** Yuanxiu Wang

参赛队员姓名: 王元秀

**School :** Dalian Maple Leaf International School

中学: 大连枫叶国际学校

**Province :** Liaoning

省份: 辽宁

**Country/Region :** China

国家/地区: 中国

**Supervisors :** Brahim El Fadil and Zhisong Lian

指导教师姓名: Brahim El Fadil and Zhisong Lian

**Topic: Investigating the Variation of the Sun's Visual Shape,  
Atmospheric Refraction and the Einstein's Special  
Relativity Considered**

论文题目: 太阳可视形状变化的试验测量和原因探究——考  
虑大气折射和爱因斯坦狭义相对论效应



## Statement of Academic Integrity

### Statement of Academic Integrity

本参赛团队声明所提交的论文是在指导老师指导下进行的研究工作和取得的研究成果。尽本团队所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果。若有不实之处，本人愿意承担一切相关责任。

**My team claims that the paper I upload is the work I get under the advice of my supervisors. As I know, except for citations and listed contents in my acknowledgements, the paper does not contain work others have published or written. I will take all the consequences of any false statements.**

参赛队员 (Members): Yuanxiu Wang

指导老师 (Supervisors): Zhisong Lian / Brahim El Fadi?

1st September, 2019



# Investigating the Variation of the Sun's Visual Shape, Atmospheric Refraction and the Einstein's Special Relativity Considered

Yuanxiu Wang

Dalian Maple Leaf International School

**Abstract:** By experimental measurements and theoretical analyses, this paper investigates the variation of the Sun's visual shape and figures out the reasons for the variation of its shape. First, the method of image processing, the method of moments and the least-square method are combined to perform experimental measurements and calculations, and the features of the Sun's visual shape are extracted from the photos of the Sun. Second, theoretical analyses are conducted based on atmospheric refraction and the Einstein's special relativity theory. The relationship model is established between the zenith and azimuth angles of the Sun, the velocity of the Sun relative to the Earth, and the observation time and position; the refraction index of the atmosphere is expressed as a function of altitude and wavelength of light; an iterative algorithm is constructed to trace rays of light in the atmosphere; a set of formulas is derived to determine the contraction ratio and contraction direction of the Sun's visual shape. Finally, the theoretical and experimental results are compared; their relative errors are less than 0.8%, which verifies the theoretical analyses. Both theoretically and experimentally, this research proves that the Sun's visual shape is an ellipse; its shape variation mainly results from atmospheric refraction effects, and the length contraction effect of the Einstein's special relativity also contributes a little, except at the time of sunrise and sunset.

**Keywords:** image processing; data fitting; atmospheric refraction; the Einstein's special relativity; zenith and azimuth angles of the Sun; ray tracing



# 太阳可视形状变化的试验测量和原因探究 ——考虑大气折射和爱因斯坦狭义相对论效应

王元秀

大连枫叶国际学校

**摘要:** 通过试验测量和理论分析, 本文探究了太阳可视形状的变化规律及其原因。首先, 应用高分辨率数码相机对太阳进行试验测量, 并结合数字图像处理方法、矩量法和最小二乘法对数据进行处理, 从太阳图像中抽取描述其形状的特征参数, 并归纳太阳形状的变化规律。然后, 通过仿真分析探究太阳形状变化的原因, 深入研究大气折射和狭义相对论效应对太阳可视形状的影响。包括: 建立了太阳天顶角和方位角、太阳和地球的相对运动速度等与观察时间和地点之间的关系模型; 将折射率表示为大气海拔高度和光波波长的函数, 构造了太阳光线追踪的迭代算法, 给出了大气折射效应对太阳形状比影响的计算方法; 推导出在综合考虑大气折射和狭义相对论效应情况下, 计算太阳可视形状参数的计算公式等。最后, 比较了试验测量数据和仿真计算结果, 其相对误差小于 0.8%, 验证了太阳形状变化原因理论分析的正确性。结果表明: 太阳的可视形状可用椭圆来描述, 误差极小可忽略, 太阳可视形状变化的主要原因是大气折射效应, 但除了日出和日落时间以外, 狭义相对论的尺缩效应对其也有一定的影响。

**关键词:** 图像处理; 数据拟合; 大气折射; 爱因斯坦狭义相对论; 太阳的天顶角和方位角; 光线追踪



## Contents

<b>Statement of Academic Integrity</b> .....	<b>1</b>
<b>Abstract</b> .....	<b>2</b>
<b>Chapter 1: Introduction</b> .....	<b>5</b>
<b>Chapter 2: Experimental Measurements and Calculations</b> .....	<b>8</b>
2.1 Processing the Sun's image taken by a camera .....	9
2.2 Extracting features from the Sun's binary image by the method of moments.....	10
2.3 Fitting an ellipse to the Sun's periphery by the least-square method .....	12
2.4 Experimental results.....	16
<b>Chapter 3: Theoretical Analysis and Simulation</b> .....	<b>23</b>
3.1 Atmospheric refraction effects.....	23
3.1.1 Model of the atmospheric altitude and the refraction index .....	23
3.1.2 Model of the theoretical zenith angle of the Sun, the observation time and position.....	27
3.1.3 Iterative algorithm for tracing rays of light in the atmosphere .....	33
3.2 The length contraction effects in the Einstein's special relativity theory.....	38
<b>Chapter 4: Numerical Simulation and Experimental Verification</b> .....	<b>42</b>
4.1 Numerical examples of the atmospheric refraction effects.....	42
4.2 Comparison and verification.....	46
<b>Chapter 5: Conclusions</b> .....	<b>53</b>
<b>References</b> .....	<b>55</b>
<b>Experiment Time and Locations</b> .....	<b>57</b>
<b>Acknowledgements</b> .....	<b>58</b>
<b>Resume of Yuanxiu Wang</b> .....	<b>60</b>
<b>Paper Detection Report</b> .....	<b>62</b>
<b>Appendix: My MATLAB Source Code for Experimental Measurements of the Sun's Visual Shape</b> .....	<b>64</b>



## Chapter 1: Introduction

When you enjoy sunrise and sunset, do you pay attention to the variation of the Sun's visual shape? Comparatively speaking, the Sun's image at sunrise or sunset is flatter than that at noon; however, although the Sun's image at noon is the roundest on a day, it still appears to be an ellipse and not a perfect circle. Very few research papers systemically explain the theoretical reasons for the variation of the Sun's visual shape, and only a few very short articles on Wikipedia and in popular science magazines simply outline the natural phenomenon based on atmospheric refraction.

The effects of atmospheric refraction have been studied widely in the fields of optical communications and weather forecast. For the free-space optical communication systems, by applying astronomical refraction formulas and considering meteorological conditions on ground, Karin and Florian (2004) have simulated the refraction angles of solar rays in the atmosphere, and calculated the vertical deviation of solar beams at various wavelengths. For the earth-to-satellite laser communications, Xiang (2008) analyzed the effect of atmospheric chromatic dispersion on the pointing error of the uplink laser beam. For the sake of improving accuracy of satellite laser ranging, Yuan et al. (2011) used ray tracing to compute the various optical paths caused by atmosphere refraction, and gave a regional distribution of the optical path difference in China. Jiang et al. (2013) investigated the relationship between the optical paths and the satellite zenith angles, and proposed an atmospheric refraction compensation scheme for the different satellite zenith angles. In order to find out about the effect of atmospheric refraction on the propagation of weather radar beams, Wang et al. (2018) explored the vertical variation of the refraction index in the first kilometer of the atmosphere with regional climate and topography. Balal et al. (2019) evaluated the effect of atmospheric refraction and absorption on the propagation of millimeter and sub-millimeter wavelengths from land to satellite. Chaim et al. (2000) made use of ray tracing to determine atmospheric refraction, and combined the digital terrain model to calculate the visual sunrise and



sunset time at some cities in Israel. Kambezidis and Papanikolaou (1990) researched the relationship between solar position and atmospheric refraction. Kambezidis and Tsangrassoulis (1993) proposed a new correction of right ascension according to solar position. Kambezidis (1997) established a set of appropriate spherical trigonometric formulas to estimate sunrise and sunset hours by considering flat and complex terrains and atmospheric refraction. Instead of spherical trigonometry, Sproul (2007) used vector analysis to research the position relationship between the Sun and the Earth.

This paper investigates the variation of the Sun's visual shape on a day, and figures out the reasons for the shape variation by massive experimental measurements and systemic theoretical analyses. First, hundreds of photos of the Sun were taken from sunrise to sunset; an image processing was performed to extract features of the Sun's visual shape from the photos. The method of moments and the least-square method were combined to fit the periphery of the Sun in the processed images; and a set of formulas was derived to calculate the feature parameters of the elliptic Sun. Error analyses showed that the relative measurements accuracy was about 0.023%, the average residuals of the fitting curve of the ellipse was only 4 pixels. The experimental results proved that the Sun's visual shape can be accurately approximated by an ellipse. Second, in order to investigate the reasons for the variation of the Sun's visual shape, the atmospheric refraction effects were first researched to simulate the influence on the Sun's visual shape at different observation time and positions. Based on Kepler's laws, the relative position and velocity between the Sun and the Earth were analyzed, and a set of formulas was derived to calculate the zenith and azimuth angles of the Sun for different observation positions and time; according to the differential equation of light propagation in an inhomogeneous medium, an iterative algorithm was developed to trace the rays of light in the atmosphere. Meanwhile, because the refraction index of the atmosphere mainly depends on air density, it is expressed as a function of altitude and wavelength of light. Under this expression, it is proved theoretically that the trajectory of sunlight is a planar curve. Nevertheless, as a result of atmospheric refraction, the Sun's visual shape contracts only in the zenith direction, resulting in an elliptic Sun in the



observer's eyes. In addition, the length contraction phenomenon in the Einstein's special relativity theory due to the relative movement between the Sun and the observer was also investigated to simulate its effects on the Sun's visual shape at different observation time and positions. Because the relative velocity between the Sun and the observer varies with time, especially the direction of the motion, the effect of the length contraction phenomenon on the Sun's visual shape also varies with time; therefore, the analysis included the contraction ratio and the contraction direction of the Sun's visual shape, which are concepts describing the ratio of the minor to the major axis and the direction of semi-minor axis of the Sun's visual shape. A relationship between the length contraction effect, observation positions and time was established. Although the effect of the Einstein's special relativity theory also makes the Sun's visual shape contract in one direction and results in an elliptic Sun, this differs from the atmospheric refraction effect, because the contraction direction caused by the Einstein's special relativity theory varies with time. Comprehensively considering the effects of atmospheric refraction and the Einstein's special relativity theory on the Sun's visual shape, although different, we can prove that the Sun's visual shape is still an ellipse. Therefore, a formula calculating the shape parameter of the ellipse was derived. Finally, the observation position was chosen to be Dalian, and the observation time included before winter solstice in 2018 (mid-December), after spring equinox in 2019 (late March) and around summer solstice in 2019 (late June). Then, the variation of the Sun's visual shape was simulated, including two cases: one with atmospheric refraction effects only, and another with both atmospheric refraction and the Einstein's special relativity theory. By comparison, it can be found that the relative errors of the simulation results and the experimental data are less than 0.8%. These results show that the reason for the variation of the Sun's visual shape is mainly atmospheric refraction; and at time except of sunrise and sunset, the length contraction effect of the Einstein's special relativity theory also contributes, but little.



## Chapter 2: Experimental Measurements and Calculations

In order to investigate the variation of the Sun's visual shape, a high-resolution digital camera was used to take photos of the Sun from sunrise to sunset. The time chosen included before winter solstice in 2018 (mid-December), after spring equinox in 2019 (late March) and around summer solstice in 2019 (late June); the position was Dalian, a city in northeast China. In Fig. 1, three photos are given, the projection plane is perpendicular to the direction from the camera to the Sun, and the horizontal direction of the photo is about parallel to the sea level. Here, the Sun's visual shape is assumed an ellipse. In the later Sections, the assumption will be verified by experimental results and theoretical analyses. A shape parameter describing the ellipse is defined as  $k = b/a$ , where  $a$  stands for the dimension of its semi-major axis, and  $b$  stands for the dimension of its semi-minor axis. When the sun rises or sets, the vertical direction of the photo or the direction of  $b$  is about perpendicular to sea level. A bigger  $k$  means that the Sun's visual shape is rounder. Seeing from Fig. 1, at sunrise, noon and sunset, the major and minor axes of the Sun are basically along the horizontal and vertical directions of the images.

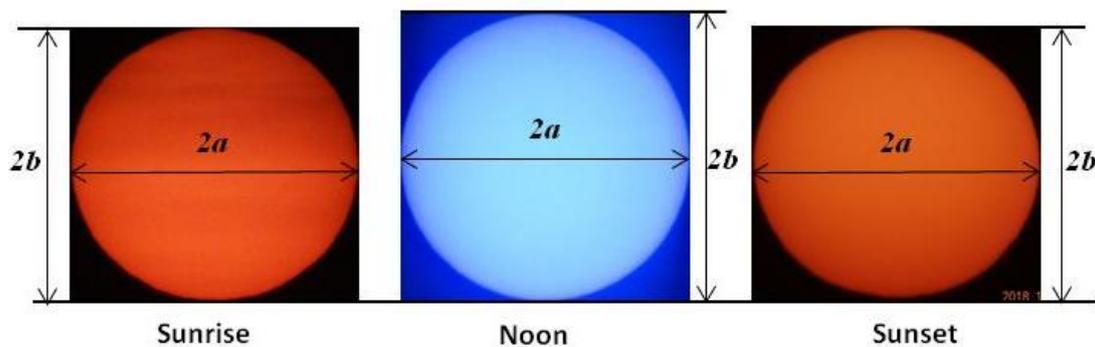


Fig. 1 Images of the Sun

For extracting the shape parameter  $k$  from the Sun's photos, image processing, and the methods of moments and least-squares were employed. First, the Sun's image expressed in RGB was converted to a binary image expressed by two luminance values of 0 and 1; then the periphery of the Sun was extracted from the binary image by removing the interior pixels; next, under the assumption that the Sun's binary image is an ellipse, the method of moments was used to evaluate its feature



parameters, such as the semi-major axis, the semi-minor axis, the coordinates of the center, and the angle between the major axis of the ellipse and the horizontal direction of the image, which were defined as the initial values of the iterative algorithm for fitting an ellipse to the periphery of the Sun. Finally, the least-squares method or the iterative algorithm was applied to fit an ellipse to the periphery of the Sun and compute the feature parameters of the Sun's elliptical image.

## 2.1 Processing the Sun's image taken by a camera

A photo of the Sun taken by a camera is a true color image expressed in RGB. In order to extract the shape parameter  $k$  from the Sun's photo, the Sun's image should be processed.

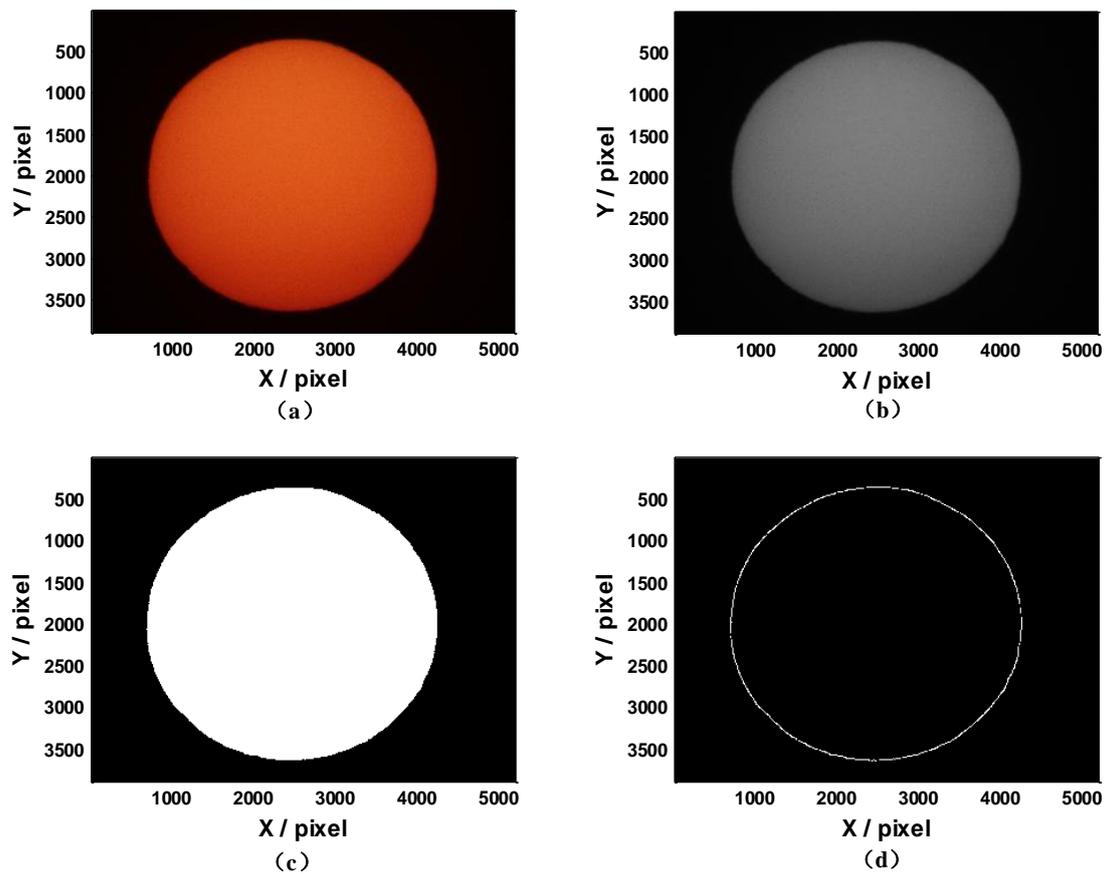


Fig. 2 Image processing procedure

First, the true color image was converted to a grayscale image by removing the hue and saturation information, while retaining luminance. The mapping relationship between luminance and RGB can be described as



$$Y = 0.299R + 0.587G + 0.114B \quad (1)$$

where,  $Y$  denotes the luminance value at every pixel, ranging from 0 to 1;  $Y=1$  denotes white;  $Y=0$  stands for black. Then, in order to reduce noise in the grayscale image, a 3-by-3 neighborhood median filtering was performed. Fig. 2(a) is a true color image taken by a camera, and Fig. 2(b) is a processed image after median filtering. Next, according to the contrast of Fig. 2(b), a luminance threshold  $\tilde{Y}$  was estimated, and the luminance values of all pixels with  $Y \geq \tilde{Y}$  were replaced with the value 1, while the luminance values of the other pixels were set to 0. In this way, Fig. 2(b) is converted to a binary image expressed by two luminance values of 0 and 1. Besides, in order to eliminate isolated points and burrs on the periphery, the morphological operations were applied to the binary image, including image erosion and dilation. During the image erosions and dilations, a 3-by-3 matrix  $\mathbf{SE}$  was used as a structuring element object

$$\mathbf{SE} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (2)$$

After performing 3 times erosions, 3 times dilations were implemented. Fig. 2(c) is the binary image after performing erosions and dilations. The last step was to remove the interior pixels and retain the periphery of the Sun. For a pixel in Fig. 2(c), if the luminance values of all its 4-connected neighbors were 1, its luminance value would be set to 0, otherwise, the pixel would be retained as a point on the periphery of the Sun, and its luminance value is 1. Fig. 2(d) illustrates the periphery of the Sun.

## 2.2 Extracting features from the Sun's binary image by the method of moments

Fig. 2(c) is a binary image, where the Sun can be regarded as an ellipse. The feature parameters of the ellipse include the dimension  $a$  of the semi-major axis, the dimension  $b$  of the semi-minor axis, the center coordinate  $C(x_c, y_c)$ , and the angle  $\gamma$  between the major axis of the ellipse and the horizontal direction of the image, as shown in Fig. 3.

The binary image is expressed by an  $M \times N$  matrix consisting of 0 and 1; 1 and 0 denote white and black, respectively. It is assumed that the matrix element is



$IM_{i,j}, i = 1, 2, \dots, M; j = 1, 2, \dots, N$ , using the method of moments, the Sun's area  $A$  and the center coordinate  $C(x_c, y_c)$  can be calculated by

$$\begin{aligned} A &= \sum_{i=1}^M \sum_{j=1}^N IM_{i,j} \\ x_c &= \frac{1}{A} \sum_{i=1}^M \sum_{j=1}^N (i \times IM_{i,j}) \\ y_c &= \frac{1}{A} \sum_{i=1}^M \sum_{j=1}^N (j \times IM_{i,j}) \end{aligned} \quad (3)$$

The angle  $\gamma$  between the major axis of the ellipse and the horizontal direction of the image can also be obtained by

$$\tan(2\gamma) = \frac{2I_{xy}}{I_{xx} - I_{yy}} \quad (4)$$

where,  $I_{xx}$ ,  $I_{xy}$  and  $I_{yy}$  are the secondary moments relative to  $C(x_c, y_c)$ , which can be computed by the following equations

$$\begin{aligned} I_{xx} &= \sum_{i=1}^M \sum_{j=1}^N (i - x_c)^2 \times IM_{i,j} \\ I_{xy} &= \sum_{i=1}^M \sum_{j=1}^N (i - x_c) \times (j - y_c) \times IM_{i,j} \\ I_{yy} &= \sum_{i=1}^M \sum_{j=1}^N (j - y_c)^2 \times IM_{i,j} \end{aligned} \quad (5)$$

Applying Eqs. (4) and (5), the principal moments of inertia,  $I_1$  and  $I_2$ , are obtained

$$\begin{aligned} I_1 &= \cos^2(\gamma)I_{xx} + \sin^2(\gamma)I_{yy} + \sin(2\gamma)I_{xy} \\ I_2 &= \cos^2(\gamma)I_{yy} + \sin^2(\gamma)I_{xx} - \sin(2\gamma)I_{xy} \end{aligned} \quad (6)$$

If the Sun in the binary image is considered as an ellipse with two semi-axes  $a$  and  $b$ , its area  $A$  and the principal moments of inertia  $I_1$  and  $I_2$  can be expressed as

$$A = \pi ab \quad I_1 = \frac{\pi}{4} ab^3 \quad I_2 = \frac{\pi}{4} a^3 b \quad (7)$$

Therefore,  $a$  and  $k$  can be solved by

$$a = 2\sqrt{I_2/A} \quad k = \sqrt{I_1/I_2} \quad (8)$$

Applying the method of moments, the feature parameters of the Sun's binary image are finally extracted, including  $a$ ,  $b$ ,  $C(x_c, y_c)$ ,  $\gamma$  and  $k$ .

Taking the Sun's image in Fig. 2(c) for example, the feature parameters are extracted. The result data are given in Table 1, where the length unit is pixel. Comparing the binary image of the Sun with the ellipse obtained by Eq. (7), the area of non-overlapped domain is calculated, equal to 37418, and its error relative to the total area is about 0.38%.



Table 1 Results calculated by the method of moments (Fig. 2(c))

$x_c$	$y_c$	A	$I_{xx}$	$I_{xy}$	$I_{yy}$
2540.6	1968.8	9788446	$8.0768 \times 10^{12}$	$1.3993 \times 10^{10}$	$7.1981 \times 10^{12}$
a	b	k	$I_1$	$I_2$	$\gamma$
1816.7	1715.0	0.9440	$8.0771 \times 10^{12}$	$7.1978 \times 10^{12}$	$-0.9123^\circ$

### 2.3 Fitting an ellipse to the Sun's periphery by the least-square method

Fig. 2(d) illustrates the periphery of the Sun's visual shape. It is assumed that  $P_i(x_i, y_i)$ ,  $i = 1, 2, \dots, q$  is a point on the periphery. In order to fit an ellipse to the periphery points, two coordinate systems are established, one is the global coordinate system  $O - XY$ , the other is the local coordinate system  $C - X_e Y_e$ , as shown in Fig. 3. In  $O - XY$ ,  $X$  and  $Y$  are along the horizontal and vertical directions of the image, respectively; in  $C - X_e Y_e$ , the original point is set as the center of the ellipse;  $X_e$  and  $Y_e$  are along the semi-major axis and semi-minor axis of the ellipse, respectively.

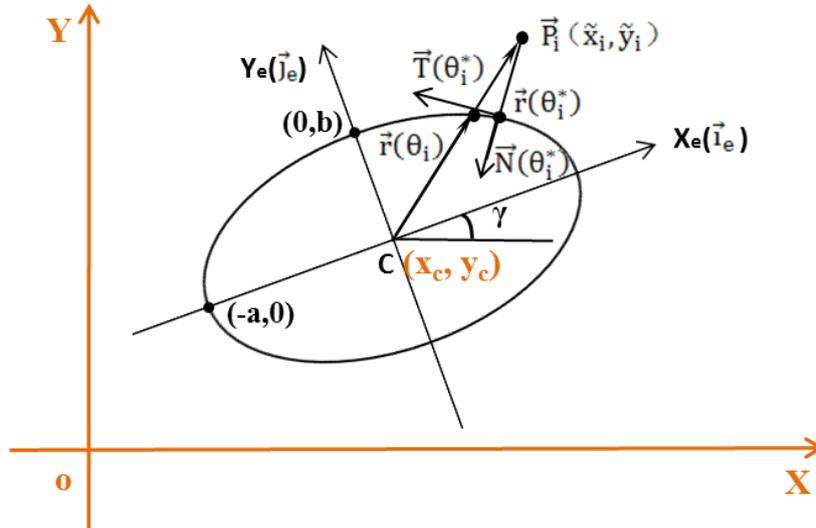


Fig. 3 Relationship between two coordinate systems

In the local coordinate system  $C - X_e Y_e$ ,  $P_i(x_i, y_i)$  on the Sun's periphery can be expressed in the form of a vector  $\vec{P}_i$ , satisfying

$$\begin{aligned}
 \vec{P}_i &= \tilde{x}_i \vec{1}_e + \tilde{y}_i \vec{1}_e & i = 1, 2, \dots, q \\
 \tilde{x}_i &= (x_i - x_c) \cdot \cos(\gamma) + (y_i - y_c) \cdot \sin(\gamma) \\
 \tilde{y}_i &= -(x_i - x_c) \cdot \sin(\gamma) + (y_i - y_c) \cdot \cos(\gamma)
 \end{aligned} \tag{9}$$



The equation of fitting a reference ellipse can be written as

$$\begin{aligned}\vec{r}(\theta) &= x_e \vec{i}_e + y_e \vec{j}_e \quad \theta \in [0, 2\pi] \\ x_e &= a \cdot \cos(\theta) \\ y_e &= b \cdot \sin(\theta)\end{aligned}\quad (10)$$

where,  $\vec{r}(\theta)$  is the polar radius of an edge point on the fitting reference ellipse;  $\theta$  is the corresponding polar angle;  $\vec{i}_e$  and  $\vec{j}_e$  stand for the unit vectors along  $X_e$  and  $Y_e$ ;  $(x_e, y_e)$  is the coordinate of the point in C- $X_e Y_e$ . Its unit tangent vector  $\vec{T}(\theta)$  and unit normal vector  $\vec{N}(\theta)$  can be expressed as

$$\begin{aligned}\vec{T}(\theta) &= T_x \vec{i}_e + T_y \vec{j}_e \\ \vec{N}(\theta) &= N_x \vec{i}_e + N_y \vec{j}_e\end{aligned}\quad (11)$$

Setting  $D = \sqrt{a^2 \cdot \sin^2(\theta) + b^2 \cdot \cos^2(\theta)}$ ,  $T_x, T_y, N_x$  and  $N_y$  satisfy

$$\begin{cases} T_x = -a \cdot \sin(\theta)/D \\ T_y = +b \cdot \cos(\theta)/D \end{cases} \quad \begin{cases} N_x = -T_y \\ N_y = +T_x \end{cases}\quad (12)$$

A fitting error  $\varepsilon_i^*$  is defined as the shortest distance from the Sun's periphery point  $\vec{P}_i$  to the fitting reference ellipse

$$\varepsilon_i^* = (\vec{P}_i - \vec{r}(\theta_i^*)) \cdot \vec{N}(\theta_i^*)\quad (13)$$

where, the sign “ $\cdot$ ” means the scalar product of vectors;  $\vec{r}(\theta_i^*)$  is the projection of point  $\vec{P}_i$  on the fitting reference ellipse along the normal direction.  $\theta_i^*$  satisfies

$$(\vec{P}_i - \vec{r}(\theta_i^*)) \cdot \vec{T}(\theta_i^*) = 0\quad (14)$$

In order to calculate  $\theta_i^*$ , first connect  $\vec{P}_i$  to the center C of the fitting reference ellipse by a straight line, then determine the intersection  $\vec{r}(\theta_i)$  of the line and the ellipse, and finally take  $\theta_i$  as an initial value of  $\theta_i^*$ , where  $\theta_i$  satisfies

$$\tan(\theta_i) = (a \cdot \tilde{y}_i)/(b \cdot \tilde{x}_i)\quad (15)$$

Based on Eq. (14), an iterative algorithm is constructed to calculate  $\theta_i^*$

$$\theta_i^{n+1} = \theta_i^n + (\vec{P}_i - \vec{r}(\theta_i^n)) \cdot \vec{T}(\theta_i^n) / \sqrt{a^2 \cdot \sin^2(\theta_i^n) + b^2 \cdot \cos^2(\theta_i^n)}\quad (16)$$

where,  $n$  stands for the number of iterations. During fitting the ellipse, the number of iterations is not more than 2, that is  $n \leq 2$ , because  $\vec{P}_i$  is near the ellipse when taking  $\theta_i$  as an initial value of  $\theta_i^*$ .



During the iterative calculation, the fitting error  $\varepsilon_i^*, i = 1, 2, \dots, q$  can be obtained by Eq. (13). For the sake of reducing the error and finding the optimal fitting to the ellipse, the feature parameters of the ellipse should be modified iteratively, including  $a$ ,  $b$ ,  $C(x_c, y_c)$ , and  $\gamma$ . Assuming that the increments of the five parameters are  $\Delta x_c$ ,  $\Delta y_c$ ,  $\Delta \gamma$ ,  $\Delta a$  and  $\Delta b$ , applying Eq. (13), the relationship between the increments and the fitting errors can be established

$$\begin{aligned} \varepsilon_i = \varepsilon_i^* - \Delta \gamma [x_e(\theta_i^*)N_y(\theta_i^*) - y_e(\theta_i^*)N_x(\theta_i^*)] - \Delta x_c N_x(\theta_i^*) \\ - \Delta y_c N_y(\theta_i^*) - \Delta a \cdot \cos(\theta_i^*)N_x(\theta_i^*) - \Delta b \cdot \sin(\theta_i^*)N_y(\theta_i^*) \end{aligned} \quad (17)$$

where  $\varepsilon_i$  is the shortest distance from  $\vec{P}_i$  to the fitting reference ellipse with modified feature parameters. For convenience, Eq. (17) is rewritten as

$$[\varepsilon]_{q \times 1} = [\varepsilon^*]_{q \times 1} - [X]_{q \times 5} [\Delta]_{5 \times 1} \quad (18)$$

The parameter adjustment results in the change of the normal vector of the fitting reference ellipse. In Eq. (18),  $[\varepsilon]_{q \times 1}$  is a column vector consisting of the normal errors calculated after adjusting the parameters, and its  $i$ th element is  $\varepsilon_i$ ;  $[\varepsilon^*]_{q \times 1}$  is a column vector consisting of the normal errors calculated before adjusting the parameters, and its  $i$ th element is  $\varepsilon_i^*$ ;  $[X]_{q \times 5}$  is a matrix, and its  $i$ th row  $[X]_{i \times 5}$  is  $[x_e(\theta_i^*)N_y(\theta_i^*) - y_e(\theta_i^*)N_x(\theta_i^*), N_x(\theta_i^*), N_y(\theta_i^*), \cos(\theta_i^*)N_x(\theta_i^*), \sin(\theta_i^*)N_y(\theta_i^*)]$ ;  $[\Delta]_{5 \times 1}$  is a column vector consisting of the adjusted parameters,  $[\Delta]_{5 \times 1} = [\Delta \gamma, \Delta x_c, \Delta y_c, \Delta a, \Delta b]^T$ , and the sign “T” means the transpose of a vector or matrix.

The least-square method is applied to adjust the five feature parameters. The objective is to make  $\sum_{i=1}^q \varepsilon_i^2$  minimum. Based on Eq. (18), the normal equation of the least-square method can be described as

$$([X]_{q \times 5}^T [X]_{q \times 5}) [\Delta]_{5 \times 1} = [X]_{q \times 5}^T [\varepsilon^*]_{q \times 1} \quad (19)$$

By Solving Eq. (19), the parameter increments  $\Delta x_c, \Delta y_c, \Delta \gamma, \Delta a$  and  $\Delta b$  can be figured out.

According to the principle of statistics, by applying Eq. (18), the variance  $SE^2$  of the residuals of the fitting ellipse can be calculated



$$SE^2 = \frac{1}{q-5} \sum_{i=1}^q \varepsilon_i^2 \quad (20)$$

Subsequently, the variances  $S_a^2$  and  $S_b^2$  are obtained

$$S_a^2 = SE^2 \cdot V_{4,4} \quad S_b^2 = SE^2 \cdot V_{5,5}$$

where,  $S_a^2$  is the variance of  $a$ ;  $S_b^2$  is the variance of  $b$ ;  $V_{4,4}$  and  $V_{5,5}$  stand for the 4<sup>th</sup> and 5<sup>th</sup> diagonal elements of  $([X]_{q \times 5}^T [X]_{q \times 5})^{-1}$ .

By employing the least-square method (LSM) to fit an ellipse to the periphery of the Sun's image, the optimal result parameters are obtained, including the semi-major axis  $a^{\text{new}}$ , the semi-minor axis  $b^{\text{new}}$ , the angle  $\gamma^{\text{new}}$

$$a^{\text{new}} = a + \Delta a \quad b^{\text{new}} = b + \Delta b \quad \gamma^{\text{new}} = \gamma + \Delta \gamma \quad (21)$$

and the center coordinate  $(x_c^{\text{new}}, y_c^{\text{new}})$  of the fitting ellipse in  $O - XY$

$$\begin{aligned} x_c^{\text{new}} &= \Delta x_c \cdot \cos(\gamma^{\text{new}}) - \Delta y_c \cdot \sin(\gamma^{\text{new}}) + x_c \\ y_c^{\text{new}} &= \Delta x_c \cdot \sin(\gamma^{\text{new}}) + \Delta y_c \cdot \cos(\gamma^{\text{new}}) + y_c \end{aligned} \quad (22)$$

Taking the Sun's periphery in Fig. 2 (d) for example, the least-square method is used to fit an ellipse to the periphery. The initial values are the data listed in Table 1, which are the feature parameters calculated by the method of moments. The final results are shown in Fig. 4 and Table 2.

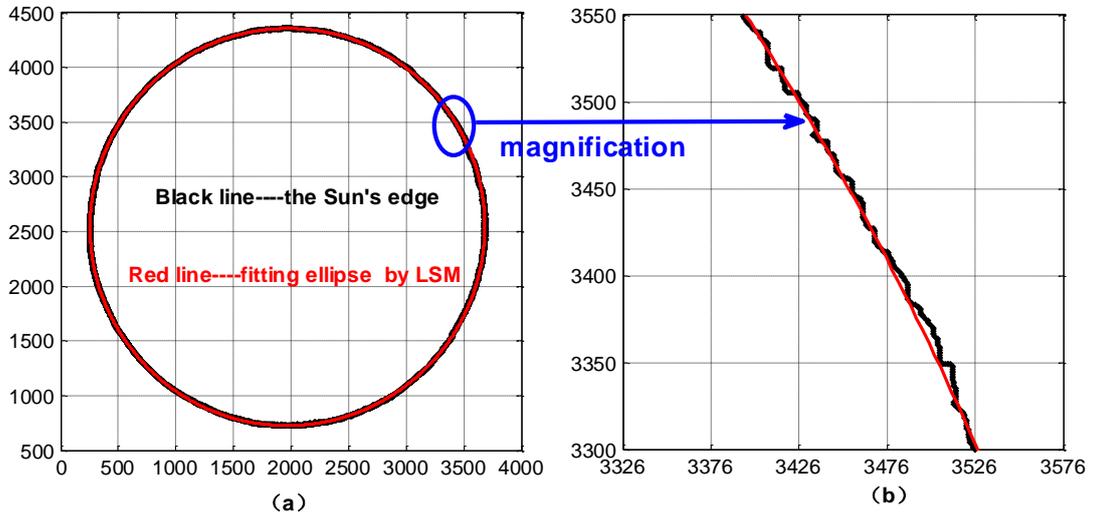


Fig. 4 The Sun's periphery and the fitting ellipse

Seeing from Table 2, the variances  $S_a^2$  and  $S_b^2$  are very little. If the confidence level is 99.73%, for the semi-major axis  $a$ , its measurement accuracy is  $\pm 3S_a = \pm 0.1971$ , and the relative accuracy is 0.011%; for the semi-minor axis  $b$ , its measurement accuracy is  $\pm 3S_b = \pm 0.2016$ , and the relative accuracy is 0.012%. For



the shape parameter  $k$  in Table 2, its relative measurement accuracy can be estimated as 0.023%, the sum of the relative accuracies of  $a$  and  $b$ . The average residuals of the fitted ellipse to the periphery of the Sun is only about 4 pixels, while the average diameter of the Sun is about 1765 pixels. The results show that the Sun's image can be described as an ellipse.

Table 2 Results calculated by the least-square method (Fig. 2(d))

$\Delta x_c$	$\Delta y_c$	$\Delta \gamma$	$\Delta a$	$\Delta b$
0.0693	0.2682	$-0.0248^\circ$	-1.2094	-0.7831
$x_c^{new}$	$y_c^{new}$	$\gamma$	$a$	$b$
1968.9	2540.9	$-0.9371^\circ$	1815.5	1714.2
q	SE	$S_a$	$S_b$	k
11504	4.1252	0.0657	0.0672	0.9442

## 2.4 Experimental results

The method of moments and the least-square method are combined to calculate the shape parameter  $k$  of the Sun's images. The images have been taken from sunrise to sunset in mid-December 2018, late March 2019 and late June 2019, and the location is Dalian. In the Sun's binary image, the Sun is considered as an ellipse. The method of moments is used to evaluate the feature parameters of the ellipse, and the feature parameters are set as the initial values of the iterative algorithm for fitting an ellipse to the periphery of the Sun; then, the least-square method or the iterative algorithm is performed to fit an ellipse to the periphery of the Sun and compute the feature parameters of the ellipse.

The resulting data are listed in Tables 3, 4 and 5, including the shape parameter  $k$ , the semi-major axis  $a$ , the semi-minor axis  $b$ , and the corresponding errors, where the maximum and minimum values of  $k$ , and their corresponding moments are marked in red. Obviously, the Sun's visual shape always appears to be an ellipse instead of a perfect circle, and the Sun's image is the roundest at noon on one day. According to the results in Tables 3, 4 and 5, at noon on a day, the shape parameters are 0.9995 in mid-December 2018, 0.9999 in late March 2019 and 0.9998 in late June



2019, respectively; while at sunrise or sunset on a day, the Sun's images are flat ellipses with the minimum shape parameters of 88.13%, 86.29% and 93.83%. It should be noted that the sunrise and sunset photos were taken at Xinghai Bridge and Jinshitan Beach, and the earliest and the latest time in the Tables were not the exact moments of sunrise and sunset because mountains obstructed some view.

Table 3 Results calculated based on the Sun's images (a day in mid-December 2018)

time	$a \pm 3S_a$	$b \pm 3S_b$	k	k errors: $\pm 3(S_a/a + S_b/b)$
<b>07:11:00</b>	2051.60 $\pm$ 0.422	1808.13 $\pm$ 0.377	<b>0.8813</b>	$\pm$ 0.041%
07:15:00	1918.37 $\pm$ 0.572	1740.53 $\pm$ 0.540	0.9073	$\pm$ 0.061%
07:16:00	1652.57 $\pm$ 0.397	1509.73 $\pm$ 0.367	0.9136	$\pm$ 0.048%
07:16:30	1653.31 $\pm$ 0.604	1515.34 $\pm$ 0.550	0.9166	$\pm$ 0.073%
07:17:00	1657.26 $\pm$ 0.350	1520.83 $\pm$ 0.331	0.9177	$\pm$ 0.043%
07:18:00	1814.23 $\pm$ 0.587	1673.19 $\pm$ 0.555	0.9223	$\pm$ 0.066%
07:22:00	1659.54 $\pm$ 0.414	1562.01 $\pm$ 0.400	0.9412	$\pm$ 0.051%
07:25:00	1653.11 $\pm$ 0.552	1570.95 $\pm$ 0.497	0.9503	$\pm$ 0.065%
07:32:00	1651.89 $\pm$ 0.296	1594.07 $\pm$ 0.291	0.9650	$\pm$ 0.036%
07:36:00	1672.73 $\pm$ 0.252	1624.79 $\pm$ 0.252	0.9713	$\pm$ 0.031%
07:37:00	1669.71 $\pm$ 0.196	1624.07 $\pm$ 0.192	0.9727	$\pm$ 0.024%
07:49:00	1590.75 $\pm$ 0.181	1564.15 $\pm$ 0.179	0.9833	$\pm$ 0.023%
08:07:00	1866.79 $\pm$ 0.136	1849.31 $\pm$ 0.136	0.9906	$\pm$ 0.015%
08:11:00	1713.01 $\pm$ 0.134	1698.65 $\pm$ 0.132	0.9916	$\pm$ 0.016%
08:21:00	1692.02 $\pm$ 0.103	1680.89 $\pm$ 0.102	0.9934	$\pm$ 0.012%
09:18:00	1777.32 $\pm$ 0.098	1772.44 $\pm$ 0.097	0.9973	$\pm$ 0.011%
09:47:00	1817.90 $\pm$ 0.086	1814.91 $\pm$ 0.085	0.9984	$\pm$ 0.009%
10:09:00	1741.56 $\pm$ 0.094	1739.42 $\pm$ 0.094	0.9988	$\pm$ 0.011%
<b>10:25:00</b>	1663.48 $\pm$ 0.105	1662.66 $\pm$ 0.105	<b>0.9995</b>	$\pm$ 0.013%
<b>11:38:00</b>	1636.65 $\pm$ 0.081	1635.88 $\pm$ 0.081	<b>0.9995</b>	$\pm$ 0.010%
<b>12:07:00</b>	1792.39 $\pm$ 0.184	1791.52 $\pm$ 0.181	<b>0.9995</b>	$\pm$ 0.020%
13:09:00	1856.95 $\pm$ 0.155	1855.72 $\pm$ 0.158	0.9993	$\pm$ 0.017%



14:10:00	1699.52 ±0.421	1695.62 ±0.424	0.9977	±0.050%
14:50:00	1726.93 ±0.211	1719.97 ±0.204	0.9960	±0.024%
15:04:00	1568.53 ±0.216	1560.47 ±0.227	0.9949	±0.028%
15:28:00	1680.15 ±0.119	1664.06 ±0.118	0.9904	±0.014%
15:44:00	1610.27 ±0.122	1584.30 ±0.121	0.9839	±0.015%
16:10:00	1788.21 ±0.268	1696.28 ±0.259	0.9486	±0.030%
16:10:30	1859.19 ±0.214	1760.80 ±0.206	0.9471	±0.023%
16:12:00	1893.49 ±0.242	1781.53 ±0.233	0.9409	±0.026%
16:12:30	1895.24 ±0.213	1780.28 ±0.207	0.9393	±0.023%
16:13:00	1918.92 ±0.249	1797.80 ±0.242	0.9369	±0.026%
16:15:00	1832.14 ±0.372	1704.38 ±0.351	0.9303	±0.041%
16:15:30	1847.69 ±0.380	1715.54 ±0.360	0.9285	±0.042%
16:19:00	1996.37 ±0.484	1807.51 ±0.450	0.9054	±0.049%
16:20:00	1776.78 ±0.390	1607.40 ±0.383	0.9047	±0.046%
16:20:30	1773.00 ±0.382	1600.67 ±0.371	0.9028	±0.045%

Table 4 Results calculated based on the Sun's images (a day in late March 2019)

time	$a \pm 3S_a$	$b \pm 3S_b$	k	k errors: $\pm 3(S_a/a + S_b/b)$
<b>05:56:52</b>	2083.53 ±0.014	1797.81 ±0.011	<b>0.8629</b>	±0.001%
06:01:02	1625.48 ±0.007	1463.32 ±0.006	0.9002	±0.001%
06:02:00	1688.49 ±0.005	1530.90 ±0.004	0.9067	±0.001%
06:06:43	1770.73 ±0.003	1665.52 ±0.003	0.9406	±0.000%
06:09:12	1802.30 ±0.003	1700.42 ±0.003	0.9435	±0.000%
06:11:29	1934.18 ±0.003	1845.17 ±0.003	0.9540	±0.000%
06:16:14	1857.30 ±0.003	1787.54 ±0.002	0.9624	±0.000%
06:20:21	1869.46 ±0.003	1818.70 ±0.003	0.9728	±0.000%
06:25:38	1803.71 ±0.003	1767.03 ±0.003	0.9797	±0.000%
06:38:57	1803.18 ±0.002	1780.98 ±0.002	0.9877	±0.000%
07:04:50	1774.83 ±0.002	1762.89 ±0.002	0.9933	±0.000%
07:15:06	1771.28 ±0.002	1761.82 ±0.002	0.9947	±0.000%



07:34:20	1709.54 ±0.002	1704.28 ±0.002	0.9969	±0.000%
07:50:52	1823.61 ±0.002	1818.79 ±0.002	0.9974	±0.000%
08:00:20	1831.29 ±0.002	1827.64 ±0.002	0.9980	±0.000%
08:34:17	1841.25 ±0.002	1839.37 ±0.002	0.9990	±0.000%
09:04:19	1653.51 ±0.002	1651.85 ±0.003	0.9990	±0.000%
10:17:00	1778.43 ±0.002	1777.03 ±0.002	0.9992	±0.000%
10:22:21	1715.10 ±0.002	1714.20 ±0.002	0.9995	±0.000%
10:35:47	1700.60 ±0.002	1699.88 ±0.002	0.9996	±0.000%
10:40:47	1753.08 ±0.002	1751.91 ±0.002	0.9993	±0.000%
11:02:27	1679.31 ±0.002	1678.70 ±0.002	0.9996	±0.000%
<b>12:25:23</b>	1643.72 ±0.002	1643.59 ±0.002	<b>0.9999</b>	±0.000%
12:50:24	1671.63 ±0.002	1671.02 ±0.002	0.9996	±0.000%
13:00:03	1753.23 ±0.001	1750.89 ±0.001	0.9987	±0.000%
13:10:04	1731.99 ±0.002	1731.08 ±0.002	0.9995	±0.000%
16:29:18	1766.96 ±0.002	1762.56 ±0.002	0.9975	±0.000%
16:50:53	1834.97 ±0.002	1826.93 ±0.002	0.9956	±0.000%
17:02:20	1730.45 ±0.002	1721.65 ±0.002	0.9949	±0.000%
17:16:25	1803.96 ±0.001	1787.82 ±0.001	0.9911	±0.000%
17:35:25	1885.78 ±0.002	1836.25 ±0.002	0.9737	±0.000%
17:43:44	1682.79 ±0.005	1607.22 ±0.004	0.9551	±0.001%
17:51:12	1725.45 ±0.005	1608.18 ±0.005	0.9320	±0.001%

Table 5 Results calculated based on the Sun's images (a day in late June 2019)

time	$a \pm 3S_a$	$b \pm 3S_b$	k	k errors: $\pm 3(S_a/a + S_b/b)$
04:45:49	1821.16 ±0.013	1720.81 ±0.009	0.9449	±0.001%
04:46:01	1822.78 ±0.013	1729.41 ±0.008	0.9488	±0.001%
04:47:01	1821.92 ±0.009	1734.26 ±0.007	0.9519	±0.001%
04:50:03	1825.11 ±0.007	1746.73 ±0.005	0.9571	±0.001%
04:52:09	1801.83 ±0.003	1727.70 ±0.003	0.9589	±0.000%



---

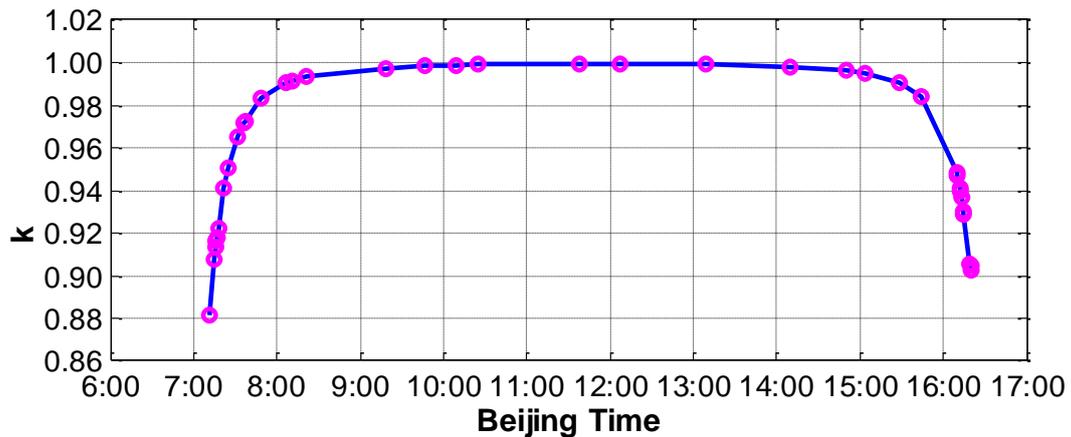
04:58:19	1828.80±0.002	1772.69±0.002	0.9693	±0.000%
04:58:45	1835.35±0.002	1781.18±0.002	0.9705	±0.000%
05:00:46	1701.34±0.003	1655.29±0.003	0.9729	±0.000%
05:02:31	1936.48±0.002	1889.75±0.002	0.9759	±0.000%
05:04:46	1890.95±0.003	1847.66±0.003	0.9771	±0.000%
05:13:04	1897.30±0.002	1874.96±0.002	0.9882	±0.000%
05:19:32	1820.33±0.002	1801.49±0.002	0.9896	±0.000%
05:21:10	1786.44±0.002	1770.50±0.002	0.9911	±0.000%
05:31:32	1858.52±0.002	1845.54±0.002	0.9930	±0.000%
05:37:12	1778.22±0.002	1767.18±0.002	0.9938	±0.000%
06:14:40	1821.13±0.002	1816.44±0.002	0.9974	±0.000%
06:15:34	1832.46±0.002	1828.51±0.002	0.9978	±0.000%
06:32:30	1829.15±0.002	1822.93±0.002	0.9966	±0.000%
07:04:26	1790.26±0.002	1787.92±0.002	0.9987	±0.000%
07:24:18	1757.60±0.002	1755.73±0.002	0.9989	±0.000%
07:58:56	1648.63±0.003	1646.38±0.003	0.9986	±0.000%
11:13:37	1553.44±0.002	1552.92±0.002	0.9997	±0.000%
11:32:59	1539.83±0.002	1539.18±0.002	0.9996	±0.000%
<b>11:52:07</b>	1690.82±0.002	1690.52±0.002	<b>0.9998</b>	±0.000%
12:20:45	1666.14±0.002	1665.31±0.002	0.9995	±0.000%
12:46:47	1714.09±0.001	1713.26±0.001	0.9995	±0.000%
13:07:05	1717.40±0.002	1714.57±0.002	0.9984	±0.000%
13:35:05	1796.36±0.001	1794.49±0.001	0.9990	±0.000%
14:37:57	1796.12±0.001	1794.42±0.001	0.9991	±0.000%
15:18:53	1754.75±0.002	1753.22±0.002	0.9991	±0.000%
16:00:53	1614.01±0.002	1611.78±0.002	0.9986	±0.000%
18:11:28	1780.27±0.003	1775.33±0.003	0.9972	±0.000%
18:16:04	1768.28±0.003	1761.67±0.003	0.9963	±0.000%
18:24:36	1727.34±0.003	1711.89±0.003	0.9911	±0.000%

---

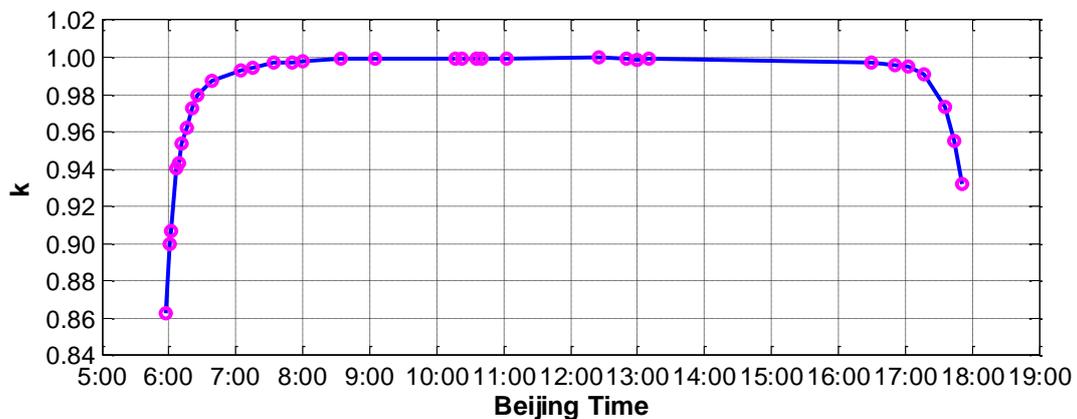


18:27:08	1721.17±0.003	1704.63±0.003	0.9904	±0.000%
18:41:38	1680.77±0.003	1653.06±0.003	0.9835	±0.000%
18:42:04	1678.20±0.003	1649.58±0.003	0.9829	±0.000%
18:48:50	1817.99±0.004	1775.55±0.004	0.9767	±0.000%
18:49:32	1818.28±0.005	1773.34±0.004	0.9753	±0.000%
18:49:56	1816.03±0.005	1772.43±0.004	0.9760	±0.000%
18:55:16	1825.18±0.003	1762.56±0.003	0.9657	±0.000%
<b>18:59:10</b>	1819.62±0.011	1707.32±0.008	<b>0.9383</b>	±0.001%

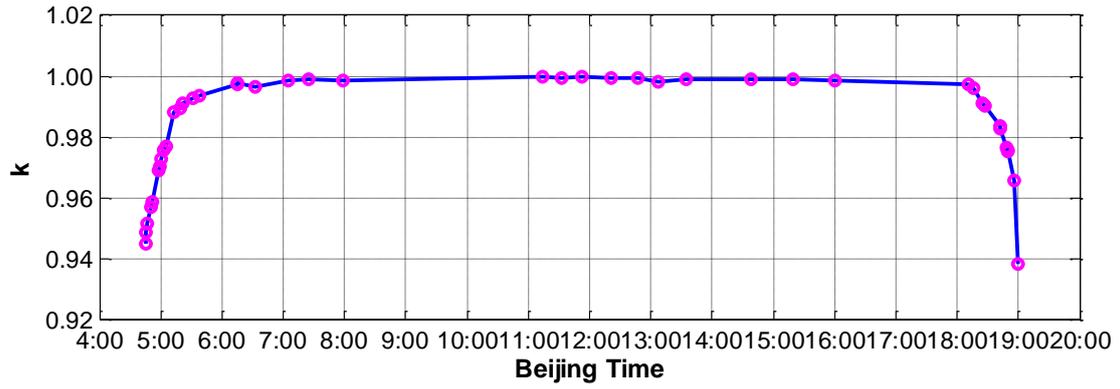
To easily observe the variation of the shape parameter, three variation curves are plotted based on the data of Tables 3, 4 and 5, as shown in Fig. 5, where the horizontal axis is Beijing time, and the vertical axis is  $k$ . Seeing in Fig. 5, it is observed that  $k$  varies fast and abruptly around sunrise and sunset, whereas it is almost flat at time except of sunrise and sunset.



(a) One day before winter solstice in 2018 (in mid-December)



(b) One day after spring equinox in 2019 (in late March)



(c) The summer solstice in 2019 (in late June)

Fig. 5 Shape parameter extracted from the Sun's images



## Chapter 3: Theoretical Analysis and Simulation

When the Sun is just rising in the morning, the Sun's image is flattened in the vertical direction, and its shape parameter is minimal; as time goes on, the Sun's visual shape becomes rounder and rounder, until noon, when the shape parameter reaches its maximum; however, the roundest image of the Sun is still an ellipse, not a circle. In this Section, based on atmospheric refraction effects and the length contraction phenomenon in the Einstein's special relativity theory, the theoretical analysis and simulation are conducted to investigate the reasons for the variation of the Sun's visual shape.

### 3.1 Atmospheric refraction effects

By considering the atmospheric refraction effects only, the Sun's visual shape is affected by the refractive index of the atmosphere, the Sun's position in the sky and observation time; therefore, a model to estimate the refraction index at an altitude was established, and a set of formulas was derived to determine the zenith and azimuth angles of the Sun for different observation time and positions. Finally, an iterative algorithm was constructed to trace rays of light in the atmosphere.

#### 3.1.1 Model of the atmospheric altitude and the refraction index

The Sun's diameter is  $1.392 \times 10^6$  km, and the average Sun-Earth distance is about  $1.496 \times 10^8$  km. It is assumed that the rays from the Sun travel in a straight line. When an observer on Earth sees the Sun, his angle of view can be calculated by the Sun's diameter over the average Sun-Earth distance, equal to  $32.004'$  or  $0.5334^\circ$ . However, the Earth is surrounded by a thick atmosphere, and the rays of light must pass through the atmosphere before reaching the Earth. As the altitude increases, the air density decreases, resulting in a decrease in the refractive index of the atmosphere. As a result, the rays of light are bent in the atmosphere, which means the angle of view varies. The variation in the angle of view mainly occurs in the vertical direction, while in the horizontal direction, the angle of view almost remains unaffected. Therefore, the Sun in our eyes is like an ellipse, not a circle. The shape of the ellipse varies with observation time and position. The altitude influences the



refractive index due to the varying temperature and pressure. When the altitude is above 50 km, the refractive index approaches 1, so the paper mainly focuses on the propagation of light rays in the atmosphere below this altitude.

The relationship between air temperature  $T$  (in  $^{\circ}\text{K}$ ) and altitude  $H$  (in km) can be formulated as (Karin et al. 2004; Jiang et al.2013; Wang et al.2018)

$$\mathbf{T}(H) = \begin{cases} T_0 - 6.5H & H \leq 11 \text{ km} \\ T_0 - 71.5 & 11 \text{ km} < H \leq 22 \text{ km} \\ T_0 + H - 93.5 & 22 \text{ km} < H \leq 32 \text{ km} \\ T_0 + 3H - 157.5 & 32 \text{ km} < H \leq 50 \text{ km} \end{cases} \quad (23)$$

where,  $T_0$  is the air temperature at ground level.

Setting  $T_0 = 269.15$   $^{\circ}\text{K}$ , i.e.  $-4$   $^{\circ}\text{C}$ , based on Eq. (23), a temperature variation curve  $\mathbf{T}(H)$  is given in Fig. 6, where the horizontal axis is the altitude  $H$ , the vertical axis is the temperature  $\mathbf{T}$ .

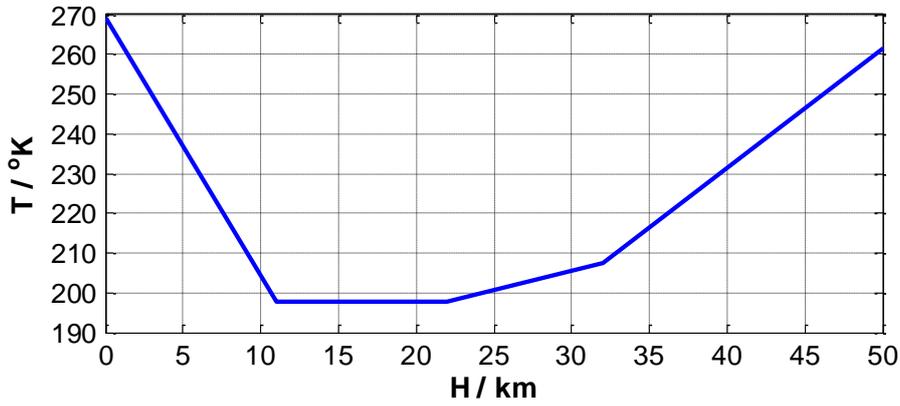


Fig. 6 Temperature variation as a function of altitude

According to hydrostatics, the air pressure  $P$  and the altitude  $H$  satisfy the condition

$$\frac{dP}{dH} = -\rho g \quad (24)$$

where,  $\rho$  is the air density in  $\text{g}/\text{m}^3$ ;  $g$  is the gravitational acceleration, equal to  $9.8 \text{ m}/\text{s}^2$ . In the atmosphere, the air satisfies the equation of state

$$P = \frac{\rho}{\mu} RT \quad (25)$$

where,  $R = 8.3144 \text{ J} \cdot \text{Mol}^{-1} \cdot \text{K}^{-1}$  is the gas constant;  $\mu = 28.966 \text{ g} \cdot \text{Mol}^{-1}$  is the molar mass of air. Substituting Eq. (25) into Eq. (24), the relationship between the air pressure  $P$  and altitude  $H$  is obtained

$$\frac{dP}{dH} = -\frac{\mu g P}{R T} \quad (26)$$



The relationship between air density  $\rho$  and altitude  $H$  can be expressed as

$$\frac{d\rho}{dH} = -\frac{\rho}{T} \left( \frac{\mu g}{R} + \frac{dT}{dH} \right) \quad (27)$$

Substituting Eq. (23) into Eqs. (26) and (27), the following formulas are obtained by calculating the integral

$$P(H) = \begin{cases} P_0 \cdot \left(1 - \frac{6.5}{T_0} H\right)^{5.2526} & H \leq 11 \text{ km} \\ P(11) \cdot e^{\frac{34.1416}{T(11)}(H-11)} & 11 \text{ km} < H \leq 22 \text{ km} \\ P(22) \cdot \left[1 + \frac{1}{T(22)} \cdot (H - 22)\right]^{-34.1416} & 22 \text{ km} < H \leq 32 \text{ km} \\ P(32) \cdot \left[1 + \frac{3}{T(32)} \cdot (H - 32)\right]^{-11.3805} & 32 \text{ km} < H \leq 50 \text{ km} \end{cases} \quad (28)$$

and

$$\rho(H) = \begin{cases} \rho_0 \cdot \left(1 - \frac{6.5}{T_0} H\right)^{4.2526} & H \leq 11 \text{ km} \\ \rho(11) \cdot e^{\frac{34.1416}{T(11)}(H-11)} & 11 \text{ km} < H \leq 22 \text{ km} \\ \rho(22) \cdot \left[1 + \frac{1}{T(22)} \cdot (H - 22)\right]^{-35.1416} & 22 \text{ km} < H \leq 32 \text{ km} \\ \rho(32) \cdot \left[1 + \frac{3}{T(32)} \cdot (H - 32)\right]^{-12.3805} & 32 \text{ km} < H \leq 50 \text{ km} \end{cases} \quad (29)$$

where,  $P_0$  and  $\rho_0$  stand for the air pressure and air density at ground level.  $P_0$ ,  $\rho_0$  and  $T_0$  satisfy Eq.(25);  $P$ ,  $T$  and  $\rho$  are functions of  $H$ .

Setting  $T_0 = 269.15$  °K,  $P_0 = 1.01325 \times 10^5$  Pa,  $\rho_0$  can be calculated from Eq. (25), that is  $\rho_0 = \frac{P_0 \mu}{RT_0} = 1.3115$  kg/m<sup>3</sup>. Based on Eqs. (23), (28) and (29), the variations of atmospheric pressure and air density as a function of altitude are obtained, as shown in Fig. 7.

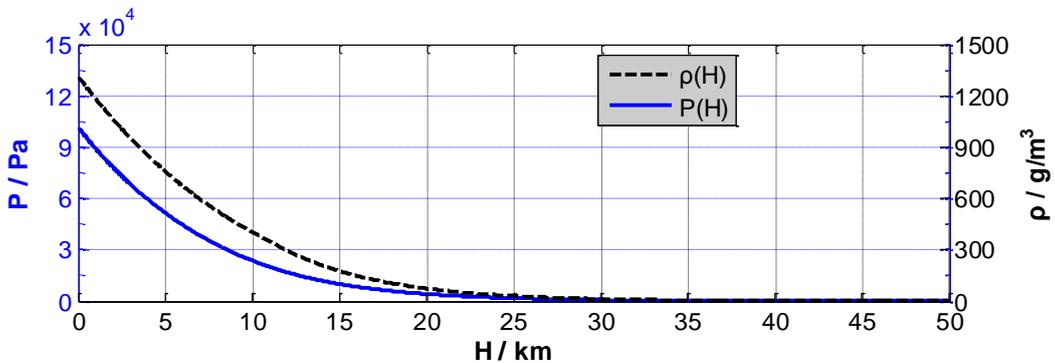


Fig. 7 Atmospheric pressure and air density variations as a function of altitude  
Neglecting influences of vapor in the air on the refractive index of the



atmosphere, based on Eq. (25), the refractive index  $n$  can be calculated by (Karin and Florian 2004; Jiang et al. 2013; Xiang 2008)

$$\begin{aligned} n(H) &= 1 + 7.76 \times 10^{-8} \times (1 + 7.52 \times 10^{-3} \lambda^{-2}) \cdot \frac{P(H)}{T(H)} \\ &= 1 + 2.23 \times 10^{-8} \times (1 + 7.52 \times 10^{-3} \lambda^{-2}) \cdot \rho(H) \end{aligned} \quad (30)$$

Where,  $\lambda$  (in  $\mu\text{m}$ ) is the wavelength of light.

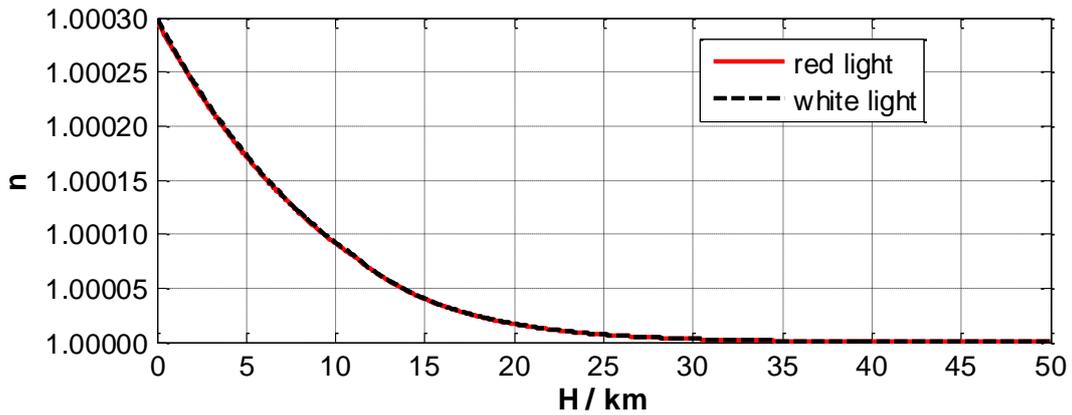


Fig. 8 Refractive index variation as a function of altitude

Based on Eq. (30), the refractive index  $n(H)$  is plotted in Fig. 8, where the red line stands for the red light from the Sun at sunrise, at the wavelength of  $\lambda = 0.685 \mu\text{m}$ ; the black line stands for the white light from the Sun at noon, at the wavelength of  $\lambda = 0.550 \mu\text{m}$ . Seeing Fig. 8, it can be found that with an increase in the altitude, the refractive index  $n$  decreases. When  $H > 30 \text{ km}$ ,  $n$  approaches 1, i.e. the refractive index in vacuum.

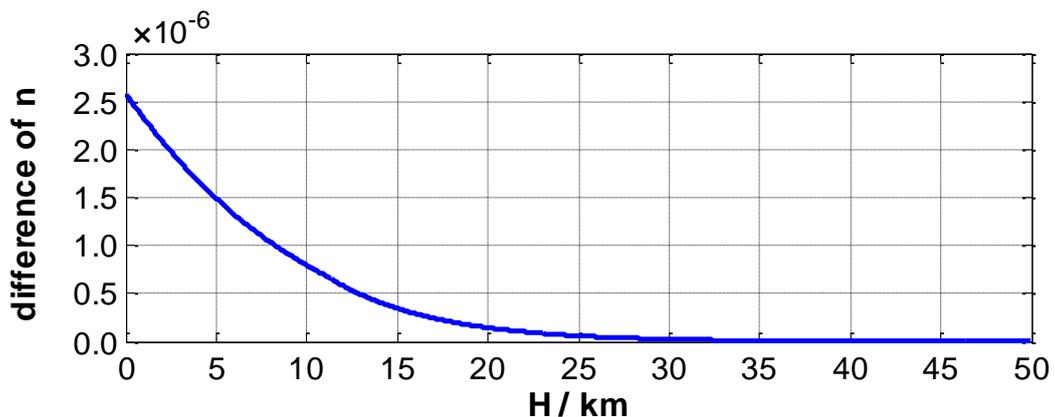


Fig. 9 Difference of  $n$  between red and white light

In order to investigate the influence of wavelength of light on the refractive index  $n$ , the difference of  $n$  between the red and white light is calculated; the results



are shown in Fig. 9. Obviously, the influence of wavelength on  $n$  mainly occurs in the atmosphere below 30 km, but the influence is very little and can be neglected.

### 3.1.2 Model of the theoretical zenith angle of the Sun, the observation time and position

The Sun's photos were taken in mid-December 2018, late March 2019 and late June 2019 at Dalian. Besides the refractive index of the atmosphere, the observation time and position also affect the Sun's visual shape. Therefore, a model of the Sun's zenith angle, the observation time and position should be established. Here, neglecting atmospheric refraction effects, a set of formulas was derived to calculate the theoretical zenith angle of the Sun according to the observation position and time.

Fig. 10 is a sectional view of the Earth surrounded by a thick atmosphere. Dalian is located at  $121.62^\circ\text{E}$  longitude and  $38.92^\circ\text{N}$  latitude. It is assumed that a line connects Dalian (point A) and the Earth's center (point O), another one connects the Sun and the Earth's center (point O). The angle  $\varphi_{DL}$  between the two lines is the theoretical zenith angle of the Sun. The Earth rotates around its axis, and simultaneously around the Sun on the ecliptic plane. The Earth's axis is not perpendicular to the ecliptic plane since there is a declination angle between the equatorial plane and the ecliptic plane. Therefore, the theoretical zenith angle  $\varphi_{DL}$  of the Sun varies with the Earth's rotation and its position on the ecliptic plane.

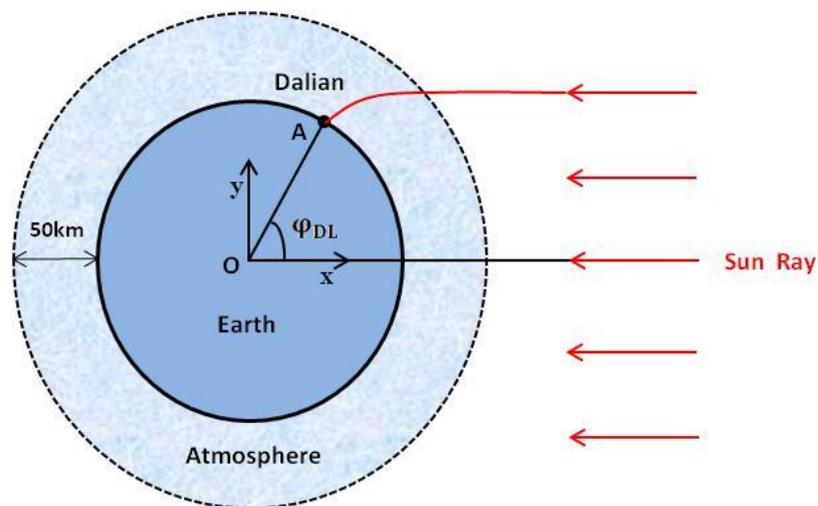


Fig. 10 Sectional view of the Earth surrounded by a thick atmosphere



For convenience, it is assumed that the Sun moves around the Earth in an elliptical orbit, and the Earth is located at a focal point of the elliptical orbit. The parameters of the elliptical orbit are: the semi-major axis  $\hat{a}_S = 1.4960 \times 10^8$  km, the semi-minor axis  $\hat{b}_S = 1.4958 \times 10^8$  km, the focal length  $\hat{c}_S = 25 \times 10^5$  km, the eccentricity  $\hat{e}_S = 0.0167$ .

The Sun coordinate system  $O-X_S Y_S$  is shown in Fig. 11, where the original point is set as the focus where the Earth is located;  $X_S$  points to the perihelion;  $Y_S$  is perpendicular to  $X_S$ . In terms of  $X_S$  the elliptical orbit can be described as

$$r(\theta) = \frac{\hat{e}_S \hat{P}_S}{1 + \hat{e}_S \cos(\theta)} \quad (31)$$

where,  $r(\theta)$  is the polar radius;  $\theta$  is the corresponding polar angle;  $\hat{P}_S = (\hat{b}_S)^2 / \hat{c}_S$ .

According to Kepler's second law,  $r(\theta)$  satisfies

$$r^2(\theta) \cdot \frac{d\theta}{dt} = C_0 \quad (32)$$

where,  $C_0$  is an undetermined constant;  $t$  stands for the time counting from the perihelion.

Calculating the integral of Eq. (32),  $t(\theta)$  can be expressed as

$$t(\theta) = \frac{1}{C_0} \int_0^\theta r^2(\theta) d\theta \quad (33)$$

The area of the elliptical orbit is  $\pi \cdot \hat{a}_S \cdot \hat{b}_S$ , which is defined as

$$\frac{1}{2} \int_0^{2\pi} r^2(\theta) d\theta = \pi \cdot \hat{a}_S \cdot \hat{b}_S \quad (34)$$

Because the period of revolution of the Earth is  $T_0 = 365.2422$  days, and  $T_0 = t(2\pi)$ , substituting Eq. (34) into Eq. (33),  $C_0$  can be obtained

$$C_0 = \frac{2\pi \cdot \hat{a}_S \cdot \hat{b}_S}{T_0} \quad (35)$$

Substituting Eqs. (31) and (35) into Eq. (33),  $t(\theta)$  can be rewritten as

$$t(\theta) = \frac{T_0}{2\pi \cdot \hat{a}_S \cdot \hat{b}_S} \int_0^\theta \left( \frac{\hat{e}_S \hat{P}_S}{1 + \hat{e}_S \cos(\theta)} \right)^2 d\theta \quad (36)$$

Letting

$$\tan\left(\frac{\varphi}{2}\right) = \sqrt{\frac{1 - \hat{e}_S}{1 + \hat{e}_S}} \tan\left(\frac{\theta}{2}\right) \quad (37)$$

replacing the variable  $\theta$  in Eq. (36) yields



$$t(\hat{\varphi}) = \frac{T_0}{2\pi} (\hat{\varphi} - \hat{e}_S \sin(\hat{\varphi})) \quad (38)$$

For a given time  $t_0$ , in order to determine the Sun's polar angle  $\theta$ ,  $\hat{\varphi}$  should be first solved by using Eq. (38); then  $\theta$  is obtained by substituting  $\hat{\varphi}$  into Eq. (37).

Eq. (38) is a transcendental equation. In order to solve for  $\hat{\varphi}$ , an iterative algorithm is constructed

$$\hat{\varphi}^{n+1} = \frac{2\pi \cdot t_0}{T_0} + \hat{e}_S \sin(\hat{\varphi}^n) \quad (39)$$

where,  $\hat{\varphi}^n$  is the iteration variable in the  $n^{\text{th}}$  iteration, with an initial iteration value of  $\hat{\varphi}^0 = \frac{2\pi \cdot t_0}{T_0}$ . The error formula of the iterative algorithm can be expressed as

$$|\hat{\varphi}^n - \hat{\varphi}^*| \leq \frac{L^n}{1-L} |\hat{\varphi}^0 - \hat{\varphi}^1| \quad (40)$$

where,  $\hat{\varphi}^*$  is the solution to Eq. (38);  $L$  is Lipschitz constant, the maximum value of the derivative of  $\hat{\varphi}^n$ , satisfying

$$L = \left| \frac{d\left(\frac{2\pi \cdot t_0}{T_0} + \hat{e}_S \sin(\hat{\varphi})\right)}{d\hat{\varphi}} \right| \leq \hat{e}_S = 0.0167 \quad (41)$$

From Eqs. (40) and (41), it is seen that the iterative algorithm converges very fast. During the iteration, when setting the accuracy as  $10^{-6}$ ( $^{\circ}$ ), the number of iterations is not more than 3, that is  $n \leq 3$ .

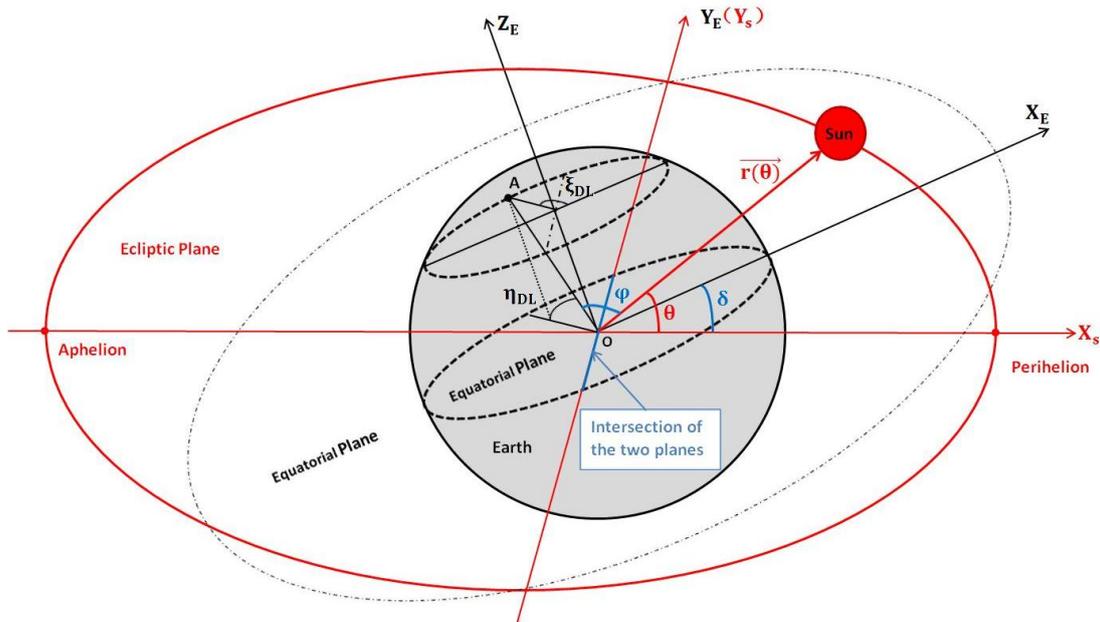


Fig. 11 Two coordinate systems  $O-X_sY_sZ_s$  and  $O-X_EY_EZ_E$

In Fig. 11, the Earth coordinate system  $O-X_EY_EZ_E$  is also established, where



the original point is the Earth's center, which coincides with the original point of the Sun coordinate system  $O-X_S Y_S$ ;  $Z_E$  is the Earth's axis;  $Y_E$  is along the line of the intersection of the equatorial plane and the ecliptic plane, and coincides with  $Y_S$ ;  $X_E$  is in the equatorial plane, and the angle  $\delta$  is the declination angle equal to  $23.433^\circ$ .

According to the Gregorian calendar, when the Sun is located at an intersection of the equatorial plane and the ecliptic plane, or when the Sun is on the  $Y_E$ , the time in 2018 is 00:15:24 on March 21 (Beijing time); in this paper, this time is taken as reference time. Beijing time is defined as the time at  $120^\circ\text{E}$  longitude. At 00:15:24 on March 21,  $\xi_{\text{BJ}}^0$  is used to denote the angle between the projection of the  $120^\circ\text{E}$  longitude on the plane  $O-X_E Y_E$  and the  $X_E$  axis. At the reference time, the Sun's polar angle  $\theta = 90^\circ$ , and there are 11.7433 hours from 00:15:24 to 12:00:00 Beijing time; then,  $\xi_{\text{BJ}}^0$  can be calculated by

$$\xi_{\text{BJ}}^0 = 90^\circ - 11.7433(\bar{\omega}_E - \bar{\omega}_S) = -86.1495^\circ \quad (42)$$

where,  $\bar{\omega}_E = 15.0411^\circ/\text{hour}$ , is the average angular velocity of the Earth around its axis;  $\bar{\omega}_S = 0.0411^\circ/\text{hour}$ , is the average angular velocity of the Earth around the Sun.

Dalian is located at  $121.62^\circ\text{E}$  longitude, with a longitude difference between Dalian and Beijing of  $1.62^\circ$ .  $\xi_{\text{DL}}$  is used to denote the angle between the  $X_E$  axis and the projection of the  $121.62^\circ\text{E}$  longitude on the  $O-X_E Y_E$ , as shown in Fig. 11. At the reference time, using  $\xi_{\text{DL}}^0$  to describe  $\xi_{\text{DL}}$ , and  $\xi_{\text{DL}}^0$  can be calculated by

$$\xi_{\text{DL}}^0 = \xi_{\text{BJ}}^0 + 1.62^\circ = -84.5295^\circ \quad (43)$$

Taking 00:15:24 on March 21, 2018 (Beijing time) as reference time, after  $\Delta t$  hours,  $\xi_{\text{DL}}$  can be obtained

$$\xi_{\text{DL}} = \xi_{\text{DL}}^0 + \Delta t(\bar{\omega}_E - \bar{\omega}_S) \quad (44)$$

In the Earth coordinate system  $O-X_E Y_E Z_E$ ,  $\eta_{\text{DL}}$  is used to stand for the angle between the radius direction from the original point (point O) to Dalian (point A) and the coordinate plane  $O-X_E Y_E$ , as shown in Fig. 11. Because Dalian is located at



38.92 °N latitude,  $\eta_{DL} = 38.92^\circ$ .  $\overrightarrow{OA}$  is the unit vector along the radius direction OA, and can be expressed by

$$\overrightarrow{OA} = \cos(\eta_{DL}) \cos(\xi_{DL}) \vec{i}_E + \cos(\eta_{DL}) \sin(\xi_{DL}) \vec{j}_E + \sin(\eta_{DL}) \vec{k}_E \quad (45)$$

where,  $\vec{i}_E$ ,  $\vec{j}_E$  and  $\vec{k}_E$  are the unit vectors along  $X_E$ ,  $Y_E$  and  $Z_E$ , respectively.

Based on Eqs. (37) and (38), setting  $\theta = 90^\circ$ , the moving time of the Sun from the perihelion to the reference time can be obtained, that is  $\Delta t_0 = 89.367789$  days. In the Sun coordinate system  $O - X_s Y_s$ , applying the iterative algorithm of Eq. (39), setting  $t_0 = \Delta t_0 + \Delta t$ , the polar angle  $\theta$  of the Sun after moving  $\Delta t$  hours can be obtained from Eqs. (37) and (38). At the moment  $t_0$ , it is assumed that  $\overrightarrow{OS}$  is the unit vector along the radius vector  $r(\theta)$  of the Sun. In the Earth coordinate system  $O - X_E Y_E Z_E$ ,  $\overrightarrow{OS}$  can be expressed as

$$\overrightarrow{OS} = \cos(\theta) \cos(\delta) \vec{i}_E + \sin(\theta) \vec{j}_E - \cos(\theta) \sin(\delta) \vec{k}_E \quad (46)$$

The angle between  $\overrightarrow{OS}$  and  $\overrightarrow{OA}$  is the theoretical zenith angle  $\varphi_{DL}$  of the Sun relative to Dalian, as shown in Fig. 10. Based on Eqs. (45) and (46),  $\varphi_{DL}$  can be figured out by

$$\begin{aligned} \cos \varphi_{DL} &= \overrightarrow{OS} \cdot \overrightarrow{OA} \\ &= \cos(\eta_{DL}) \cos(\xi_{DL}) \cos(\theta) \cos(\delta) \\ &\quad + \cos(\eta_{DL}) \sin(\xi_{DL}) \sin(\theta) - \sin(\eta_{DL}) \cos(\theta) \sin(\delta) \end{aligned} \quad (47)$$

Projecting  $\overrightarrow{OS}$  on the equatorial plane  $O - X_E Y_E$ ,  $\xi_{Sun}$  is used to denote the angle between the projection vector of  $\overrightarrow{OS}$  and the  $X_E$ , which describes the direction of the Sun, satisfying

$$\tan(\xi_{Sun}) = \frac{\sin(\theta)}{\cos(\theta) \cos(\delta)} \quad (48)$$

12:00:00 Beijing time is not the moment when the Sun shines over Dalian. The time difference  $\Delta t$  can be calculated by

$$\Delta t = \frac{\xi_{Sun} - \xi_{DL}}{\bar{\omega}_E - \bar{\omega}_S} \quad (49)$$

In Fig. 11, the Dalian coordinate system  $A - X_{DL} Y_{DL} Z_{DL}$  is established too,



where the original point is Dalian (point A);  $\overrightarrow{AN}_{DL}$  is the unit vector of the  $X_{DL}$  axis, and it is tangent to the longitude of Dalian, pointing to the North Pole;  $\overrightarrow{AW}_{DL}$  is the unit vector of the  $Y_{DL}$  axis, and it is tangent to the latitude of Dalian, pointing to west;  $Z_{DL}$  is the Zenith direction, and its unit vector is  $\overrightarrow{OA}$  expressed by Eq. (45).

$\overrightarrow{AN}_{DL}$  and  $\overrightarrow{AW}_{DL}$  can be given by

$$\begin{aligned}\overrightarrow{AN}_{DL} &= -\sin(\eta_{DL})\cos(\xi_{DL})\vec{i}_E - \sin(\eta_{DL})\sin(\xi_{DL})\vec{j}_E + \cos(\eta_{DL})\vec{k}_E \\ \overrightarrow{AW}_{DL} &= \sin(\xi_{DL})\vec{i}_E - \cos(\xi_{DL})\vec{j}_E\end{aligned}\quad (50)$$

In the Dalian coordinate system  $A - X_{DL}Y_{DL}Z_{DL}$ , the components of  $\overrightarrow{OS}$  can be written as

$$\begin{aligned}X_{DL}^S &= \overrightarrow{AN}_{DL} \cdot \overrightarrow{OS} \\ &= -\cos(\theta)\cos(\delta)\sin(\eta_{DL})\cos(\xi_{DL}) \\ &\quad -\sin(\theta)\sin(\eta_{DL})\sin(\xi_{DL}) - \cos(\theta)\sin(\delta)\cos(\eta_{DL})\end{aligned}\quad (51)$$

$$Y_{DL}^S = \overrightarrow{AW}_{DL} \cdot \overrightarrow{OS} = \cos(\theta)\cos(\delta)\sin(\xi_{DL}) - \cos(\xi_{DL})\sin(\theta)\quad (52)$$

Thus, relative to Dalian, the Sun's azimuth angle  $\psi_{DL}$  can be figured out by

$$\tan(\psi_{DL}) = -\frac{Y_{DL}^S}{X_{DL}^S}\quad (53)$$

Defining clockwise rotation as positive,  $\psi_{DL}$  describes the angle between the Sun's direction  $\overrightarrow{OS}$  and due north.

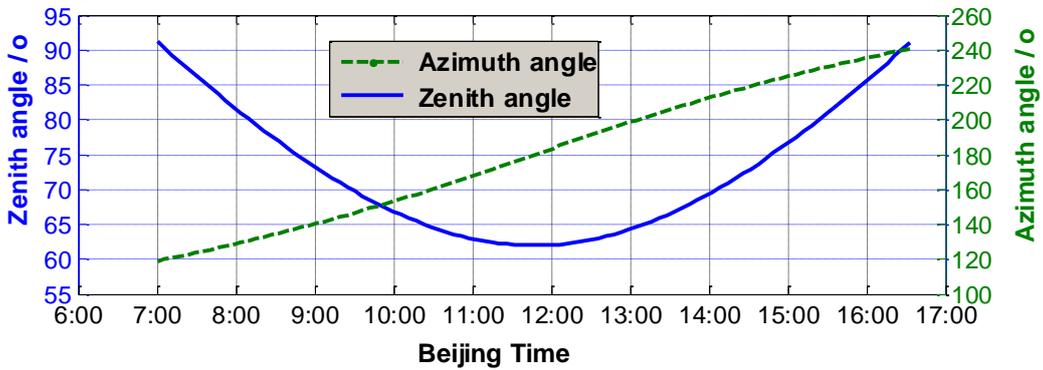


Fig. 12 Variations of the theoretical zenith and azimuth angles of the Sun

It is assumed that atmospheric refraction effects are neglected, the observation position is Dalian, and the observation time is Beijing time one day in mid-December



2018. In Fig. 12, the variations of the theoretical zenith angle  $\varphi_{DL}$  and the azimuth angle  $\psi_{DL}$  are given, which are calculated from Eqs. (47) and (53).

Besides, the time when the Sun shines over Dalian is calculated by Eq. (49). The simulation results are given in Fig. 13, where the horizontal axis is the date in the whole year of 2018, and the vertical axis is the time when the Sun shines over Dalian. From Fig. 13, it is seen that the time ranges from 11:36 to 12:08.

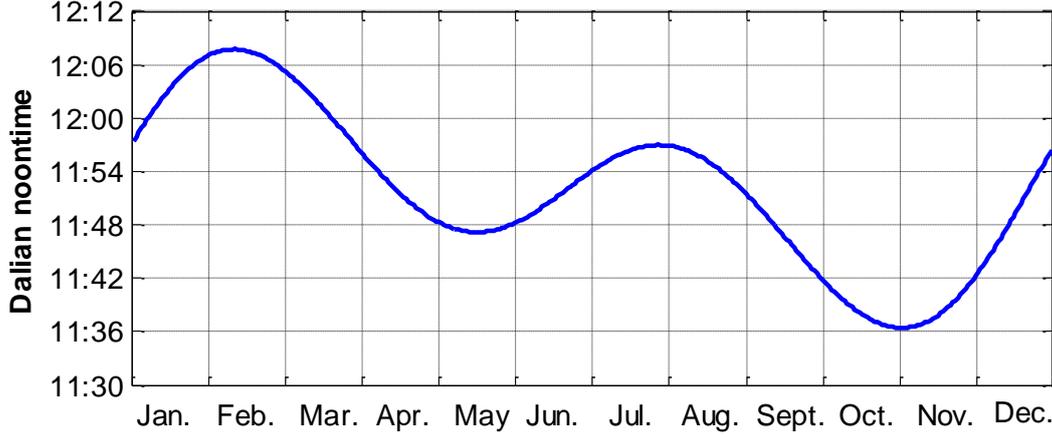


Fig. 13 Variation of the time that the Sun shines over Dalian in 2018

### 3.1.3 Iterative algorithm for tracing rays of light in the atmosphere

Because the refractive index of the atmosphere varies with altitude, the rays of light are bent in the atmosphere surrounding the Earth. In the inhomogeneous medium, the propagation of light satisfies (Born and Wolf 2007)

$$\frac{d}{ds} \left( n \frac{d\vec{R}(s)}{ds} \right) = \vec{\nabla} n \quad (54)$$

where,  $\vec{R}(s)$  stands for the trace of light;  $s$  is the length of arc;  $n$  is the refractive index of the atmosphere;  $\vec{\nabla} n$  is the gradient of the refractive index.

According to Eq. (30), the refractive index of the atmosphere can be expressed as  $n(R)$ , where  $R$  is the distance of a point in the atmosphere to the Earth's center. Because the contour surface of  $n(R)$  is spherical, its gradient direction is parallel to the radial direction of the Earth, that is  $\vec{R} \times \vec{\nabla} n = 0$ . Thus, Eq. (54) can be simplified as

$$\vec{R}(s) \times \frac{d}{ds} \left( n \frac{d\vec{R}(s)}{ds} \right) = 0 \quad (55)$$



or

$$\frac{d}{ds} \left( \vec{R}(s) \times n \frac{d\vec{R}(s)}{ds} \right) = 0 \quad (56)$$

So

$$\vec{R}(s) \times n \frac{d\vec{R}(s)}{ds} = \overrightarrow{\text{Const}} \quad (57)$$

where,  $\overrightarrow{\text{Const}}$  denotes a constant vector;  $\frac{d\vec{R}(s)}{ds}$  describes the propagation direction of light. It can be seen from Eq. (57) that all position vectors  $\vec{R}(s)$  of a ray are perpendicular to the constant vector  $\overrightarrow{\text{Const}}$ . Meanwhile, the position vectors  $\vec{R}(s)$  pass through the original point, so all position vectors  $\vec{R}(s)$  are in the same plane, and the constant vector  $\overrightarrow{\text{Const}}$  is normal to the plane. Therefore, Eq. (57) proves that the trace of light in the atmosphere is a plane curve, and the plane is determined by the Earth's center, the incidence point and incidence direction of light, as shown in Fig. 14. Using  $\hat{\theta}$  to denote the angle between the propagation direction of light and the radial direction of the Earth, taking the modulus of the two sides of Eq. (57) derives

$$R(s) \cdot n \cdot \sin(\hat{\theta}) = \text{Const} \quad (58)$$

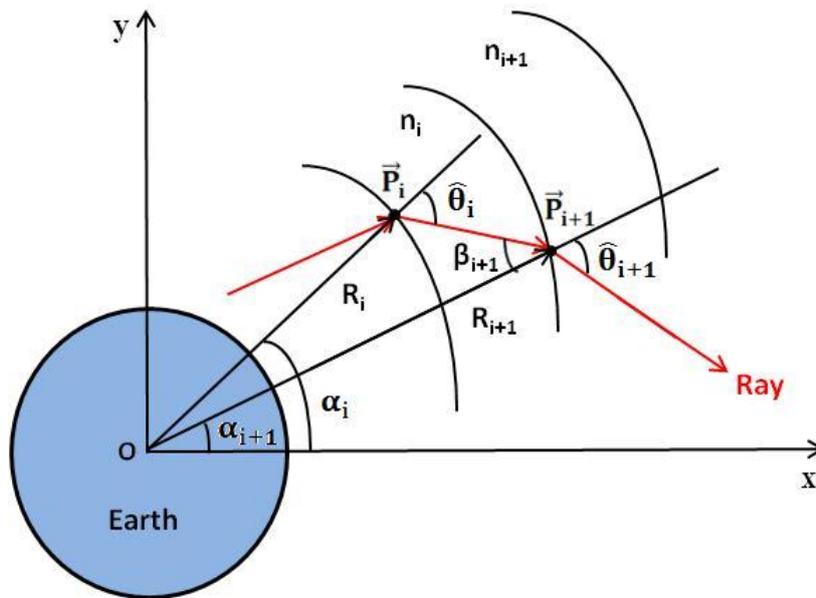


Fig. 14 Propagation direction of rays in the atmosphere

In order to trace the rays of light in the atmosphere, the atmosphere below the



altitude of 50 km is equally divided into 500 layers. It is assumed that the refractive index of each layer is constant. For convenience, the subscript "i" is used to express the  $i$ th layer of the atmosphere,  $i = 1, 2 \dots 500$ .  $R_i$  stands for the distance of the  $i$ th layer of the atmosphere from the Earth's center;  $n_i$  is the refractive index of the  $i$ th layer;  $\theta_i$  is the angle between the radial direction of the Earth and the propagation direction of light in the  $i$ th layer, as shown in Fig. 14.

Eq. (58) is rewritten as

$$R_{i+1} \cdot n_{i+1} \cdot \sin(\hat{\theta}_{i+1}) = R_i \cdot n_i \cdot \sin(\hat{\theta}_i) \quad (59)$$

According to Eq. (59), an iterative algorithm for tracing the rays of light can be constructed. In Fig. 14, a coordinate system  $O - XY$  was established, where the original point was the Earth's center; the horizontal axis  $X$  pointed to the Sun; the  $Y$  axis was perpendicular to the  $X$  axis in the propagation plane of the rays. Based on the reversibility of the optical path, it is assumed that the rays of light travel from the Earth to the Sun.  $\vec{P}_i(x_i, y_i)$  is an incidence point of light in the  $i$ th layer in the atmosphere, and  $|\vec{OP}_i| = R_i$ ;  $\alpha_i$  is the angle between  $\vec{OP}_i$  and the  $X$  axis;  $\vec{P}_{i+1}(x_{i+1}, y_{i+1})$  is the incidence point of light in the  $(i + 1)$ th layer, and  $|\vec{OP}_{i+1}| = R_{i+1}$ .  $\vec{P}_{i+1}(x_{i+1}, y_{i+1})$  can be traced and calculated by

$$\begin{aligned} \sin\hat{\theta}_{i+1} &= \frac{R_i n_i}{R_{i+1} n_{i+1}} \sin\hat{\theta}_i \\ \sin\beta_{i+1} &= \frac{R_i}{R_{i+1}} \sin\hat{\theta}_i \\ \alpha_{i+1} &= \alpha_i + \beta_{i+1} - \hat{\theta}_i \\ x_{i+1} &= R_{i+1} \cos(\alpha_{i+1}) \\ y_{i+1} &= R_{i+1} \sin(\alpha_{i+1}) \end{aligned} \quad (60)$$

where,  $\beta_{i+1}$  is the incidence angle of light from the  $i$ th to the  $(i + 1)$ th layer. Eq. (60) is a one-step explicit scheme with first-order accuracy, and can be used to simulate the propagation path of light in the atmosphere. During the simulation, the observation position coordinate  $\vec{P}_0(x_0, y_0)$  and the angle  $\hat{\theta}_0$  between the direction of light from the observation position and the radial direction of the Earth should be determined first. When simulating the rays of light from the center of the Sun,  $\hat{\theta}_0$  is



the observed zenith angle  $\tilde{\varphi}_{DL}$ .

Taking Dalian as the observation position, the rays of light propagate in the plane determined by Dalian, the Earth's center and the Sun, as shown in Fig. 10. The coordinates of  $\vec{P}_0$  can be obtained by  $[OA\cos(\varphi_{DL}), OA\sin(\varphi_{DL})]$ , where  $\varphi_{DL}$  is the theoretical zenith angle calculated by Eq. (47). Because the observation position is at the seaside in Dalian,  $OA$  is set as the Earth's radius, that is  $OA = 6371$  km. In order to solve for  $\hat{\theta}_0$ , the method of bisection is performed. An initial guess of  $\hat{\theta}_0$  is set to  $[\varphi_{DL} - 2^\circ, \varphi_{DL} + 2^\circ]$ . After the light from the Earth reaches the altitude of 50 km, its propagation direction remains unchanged, so the angle  $\tau_{Sun}$  between the ray and the x axis at that altitude is constrained as a given value.

The observation time is in mid-December 2018. Fig. 15 shows the propagation paths of rays from the center of the Sun in the atmosphere. In Fig. 15, Dalian is set as the origin; the horizontal and vertical directions are the same as the  $X$  and  $Y$  axes in Fig. 10; and the time is Beijing time. During the simulation  $\tau_{Sun} = 0$ , which means when the light reaches the atmosphere at the altitude of 50 km, its propagation direction is parallel to the  $X$  axis.

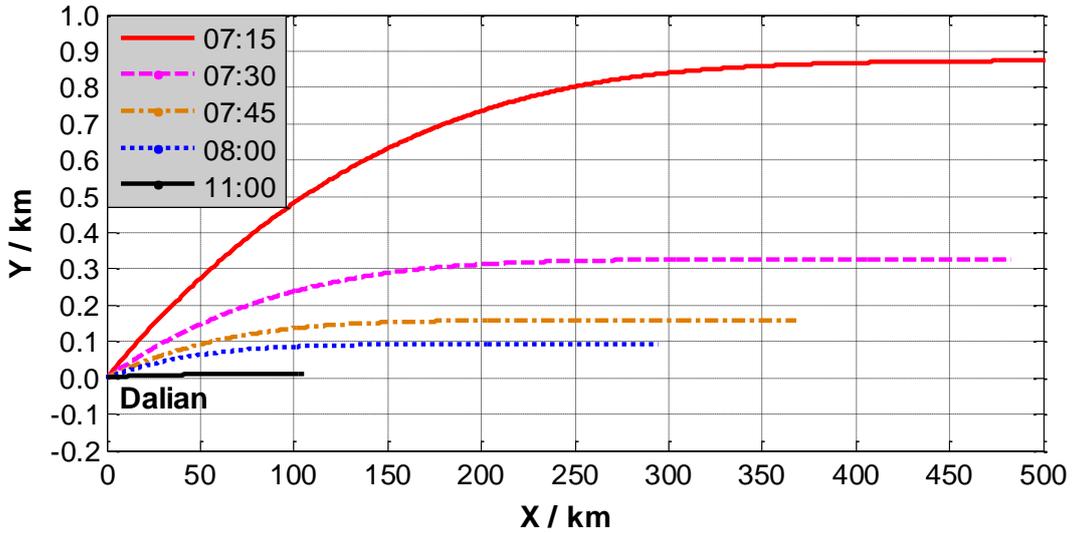


Fig. 15 Propagation paths of light rays in the atmosphere at different time in the morning at Dalian

Defining  $\delta\varphi = \tilde{\varphi}_{DL} - \varphi_{DL}$ ,  $\varphi_{DL}$  is the theoretical zenith angle of the Sun, and



$\tilde{\varphi}_{DL}$  is the observed zenith angle of the Sun. In Fig. 16, the difference between  $\varphi_{DL}$  and  $\tilde{\varphi}_{DL}$  is given, where the horizontal axis is Beijing time of a day, the vertical axis is  $\delta\varphi$ . From Fig. 16 it could be found that  $\delta\varphi$  is bigger at sunrise and sunset than at noon, a result caused by atmospheric refraction.

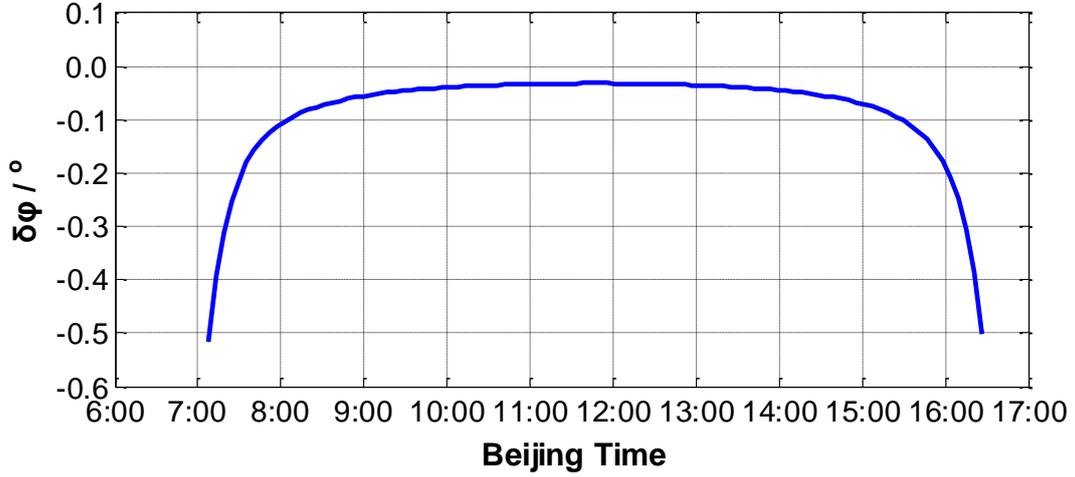


Fig. 16 Variation of the difference between  $\varphi_{DL}$  and  $\tilde{\varphi}_{DL}$  at Dalian

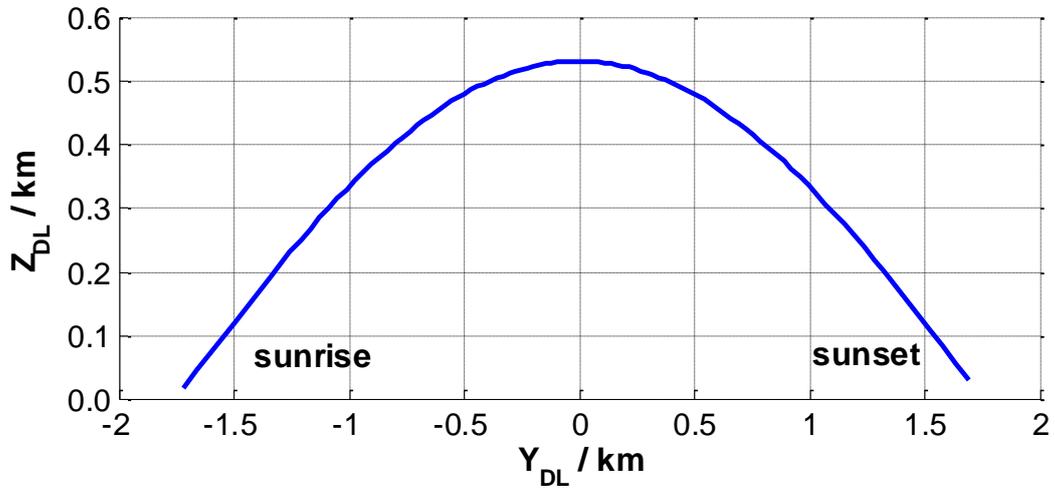


Fig. 17 Projection of the Sun's path on the  $Y_{DL}Z_{DL}$  plane at  $X_{DL} = -1$  km at Dalian

In the Dalian coordinate system  $A - X_{DL}Y_{DL}Z_{DL}$ , the direction of the Sun can be described by the observed zenith angle  $\tilde{\varphi}_{DL}$  and the azimuth angle  $\psi_{DL}$

$$\begin{aligned} X_{DL}^S &= \sin(\tilde{\varphi}_{DL})\cos(\psi_{DL}) \\ Y_{DL}^S &= -\sin(\tilde{\varphi}_{DL})\sin(\psi_{DL}) \\ Z_{DL}^S &= \cos(\tilde{\varphi}_{DL}) \end{aligned} \quad (61)$$



Based on Eq. (61), the direction of the Sun in  $A - X_{DL}Y_{DL}Z_{DL}$  can be calculated. Fig. 17 is the projection of the Sun's path on the  $Y_{DL}Z_{DL}$  plane at  $X_{DL} = -1\text{km}$ .

### 3.2 The length contraction effects in the Einstein's special relativity theory

Here, atmospheric refraction effects are neglected, and the length contraction phenomenon in the Einstein's special relativity theory is considered to investigate its effect on the Sun's visual shape. The length contraction phenomenon in the Sun's image can be described by the contraction ratio and the contraction direction, which are determined by the relative velocity between the Sun and the Earth.

Because the Sun is far away from the Earth, and the Earth rotates around its axis and simultaneously around the Sun, the relative velocity between the Earth and the Sun is very big. According to the length contraction phenomenon in the Einstein's theory of special relativity, when an observer on the Earth sees the Sun, the size of the Sun's visual shape in its moving direction will become shorter, while the size perpendicular to the moving direction remains unchanged. Assuming that the Sun is a sphere, the Sun in the eyes of the observer will be an ellipsoid, and its ratio of the minor to the major axis can be expressed as

$$k_{SRT} = \sqrt{1 - v^2/c^2} \quad (62)$$

where,  $c = 3 \times 10^8$  m/s, the velocity of light in vacuum;  $v$  is the relative velocity between the Sun and the observer.

Taking into account that the average angular velocity of the Earth around its axis is  $\bar{\omega}_E = 15.0411$  °/hour, the average angular velocity of the Earth around the Sun is  $\bar{\omega}_S = 0.0411$  °/hour, the average distance from the Sun to the Earth is  $1.496 \times 10^8$  km, the relative velocity between the Sun and the observer can approximate to

$$v \approx 1.496 \times 10^8 (\bar{\omega}_E - \bar{\omega}_S) \approx 1.088 \times 10^7 \text{ m/s} \quad (63)$$

Substituting Eq. (63) into Eq. (62) yields

$$k_{SRT} \approx 99.93 \% \quad (64)$$

$k_{SRT}$  is the shape parameter of the Sun's visual shape, describing the effect of the Einstein's special relativity theory on the visual shape of the Sun. Eq. (64) proves that even if neglecting atmospheric refraction effects, the Sun's shape in our eyes is an ellipse but not a perfect circle, and the ratio of the length contraction or the ratio of the dimensional deviation between the minor and major axes to the major axis is about



0.07%. However, seeing from Fig. 17, the Sun's direction relative to the observer varies with time; as a result, the direction of the major axis in the ellipse image of the Sun also varies.

Applying Eqs. (31), (32) and (35), the transient angular velocity  $\omega_S(\theta)$  of the Sun around the Earth can be expressed as

$$\omega_S(\theta) = \bar{\omega}_S \frac{(1 + \hat{e}_S \cos(\theta))^2}{(1 - \hat{e}_S^2)^{3/2}} \quad (65)$$

The Sun rotates around the Earth in the elliptical plane, as shown in Fig. 11; its angular velocity vector  $\vec{\omega}_S(\theta)$  can be written as

$$\vec{\omega}_S(\theta) = \omega_S(\theta) [\sin(\delta) \vec{i}_E + \cos(\delta) \vec{k}_E] \quad (66)$$

The Earth rotates around its axis  $\vec{k}_E$ ; its angular velocity vector  $\vec{\omega}_E$  can be written as

$$\vec{\omega}_E = \bar{\omega}_E \cdot \vec{k}_E \quad (67)$$

Meanwhile, applying Eqs. (31) and (46), the relative radial velocity  $\vec{V}_r(\theta)$  is obtained

$$\vec{V}_r(\theta) = \frac{\hat{e}_S \sin(\theta)}{1 + \hat{e}_S \cos(\theta)} \cdot r(\theta) \cdot \omega_S(\theta) \cdot \vec{OS} \quad (68)$$

where,  $\vec{V}_r(\theta)$  is caused by the variable distance between the Sun and the Earth.

Based on Eqs. (66), (67) and (68), the transient relative velocity  $\vec{V}_{SE}(\theta)$  between the Sun and the Earth can be calculated by

$$\vec{V}_{SE}(\theta) = \vec{V}_r(\theta) + r(\theta) [\vec{\omega}_S(\theta) - \vec{\omega}_E(\theta)] \times \vec{OS} \quad (69)$$

Substituting Eq. (69) into Eq. (62) yields

$$k_{SRT} = \sqrt{1 - |\vec{V}_{SE}(\theta)|^2 / c^2} \quad (70)$$

Eq. (70) describes the way that the Einstein's special relativity theory affects the visual shape of the Sun when the relative position varies between the Sun and the Earth.

When a high-resolution digital camera is used to take photos of the Sun, the camera remains horizontal and points to the Sun, which means the Sun's image is taken in the direction  $\vec{OS}$ . In the Dalian coordinate system  $A - X_{DL} Y_{DL} Z_{DL}$ , the horizontal direction of the camera can be described by the Sun's azimuth angle  $\psi_{DL}$ , the unit vectors  $\vec{AN}_{DL}$  and  $\vec{AW}_{DL}$  along two coordinate axes, that is

$$\vec{H}_{camera} = -\sin(\psi_{DL}) \vec{AN}_{DL} - \cos(\psi_{DL}) \vec{AW}_{DL} \quad (71)$$

where,  $\vec{H}_{camera}$  is the horizontal direction of the camera or image. The vertical



direction  $\vec{V}_{\text{camera}}$  of the camera or image can be determined by

$$\vec{V}_{\text{camera}} = \vec{H}_{\text{camera}} \times \vec{OS} \quad (72)$$

In the Sun's image, the direction of the semi-minor axis caused by the Einstein's special relativity theory is along the projection of the Sun's velocity  $\vec{V}_{\text{SE}}(\theta)$  relative to the Earth on the image. Defining  $\psi_{\text{bH}}$  as the angle between the horizontal direction of the camera and the semi-minor axis of the Sun's image, it describes the direction of the length contraction or semi-minor axis of the Sun's image, and can be calculated by

$$\tan(\psi_{\text{bH}}) = \frac{\vec{V}_{\text{camera}} \cdot \vec{V}_{\text{SE}}(\theta)}{H_{\text{camera}} \cdot \vec{V}_{\text{SE}}(\theta)} \quad (73)$$

Upon neglecting atmospheric refraction effects or only considering the Einstein's special relativity theory, and regarding the Sun's visual shape as an ellipse, its shape parameter  $k_{\text{SRT}}$  and the angle  $\psi_{\text{bH}}$  can be evaluated by Eqs. (70) and (73), respectively.

In Fig. 18, the variation of the shape parameter  $k_{\text{SRT}}$  is given, where the horizontal axis is the date in the whole year of 2018, and the vertical axis is  $k_{\text{SRT}}$ . During the simulation, the observation time is set to 12:00:00 Beijing time every day, and the observation position is Dalian. Fig. 18 shows that the Sun's visual shape is the roundest at winter solstice, and the flattest at spring and autumnal equinoxes.

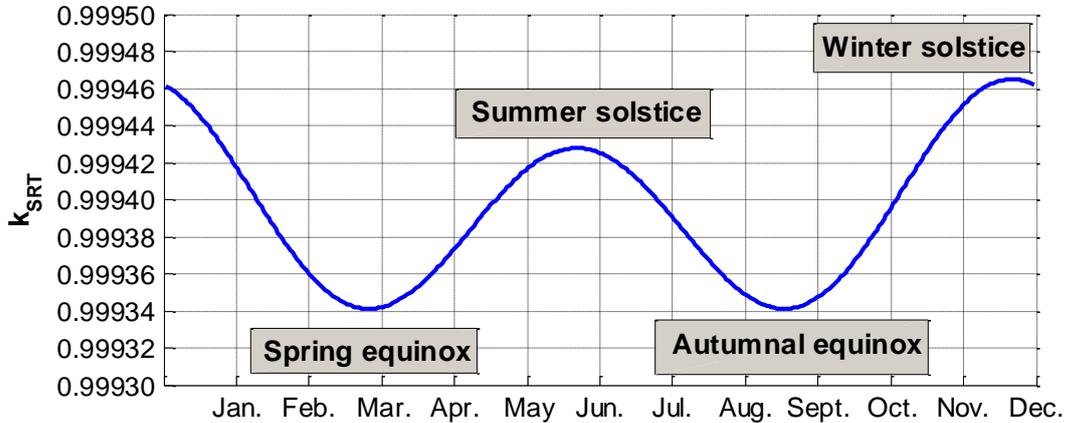


Fig. 18 Variation of the shape parameter  $k_{\text{SRT}}$  in 2018 at Dalian

In Fig. 19, the variation of the angle  $\psi_{\text{bH}}$  is given, where the horizontal axis is the Beijing time of a day in mid-December 2018, the vertical axis is the angle between the semi-minor axis of the Sun's ellipse image and the horizontal direction of the camera, which describes the direction of the length contraction in the Sun's image. Fig. 19 shows that the angle ranges from  $47^\circ$  to  $-47^\circ$ , and the extremes occur at sunrise and sunset. The two directions of the length contraction occurring at sunrise



and sunset basically satisfy the mirror-image relation.

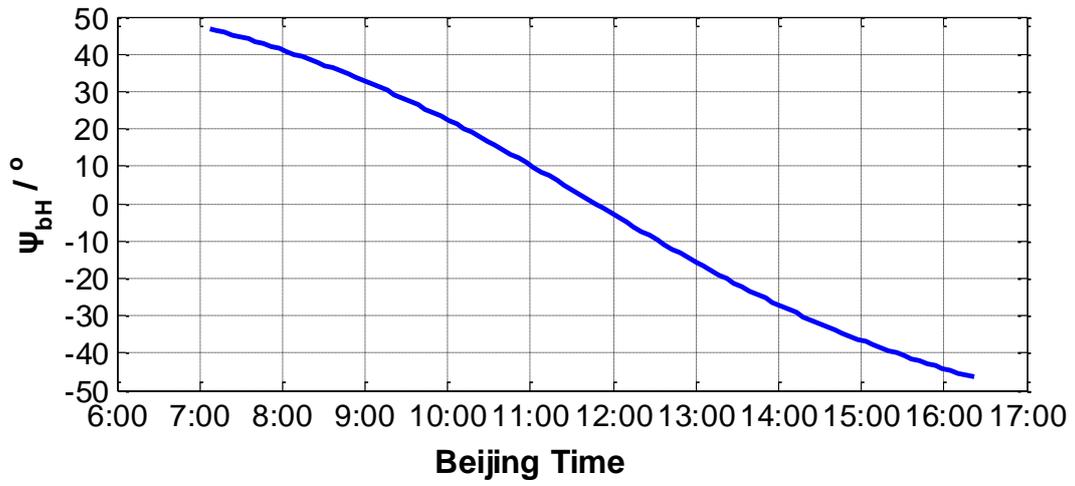


Fig. 19 Variation of the angle  $\psi_{bH}$  on a day of mid-December 2018 at Dalian



## Chapter 4: Numerical Simulation and Experimental Verification

### 4.1 Numerical examples of the atmospheric refraction effects

Eq. (57) shows that the light in the atmosphere propagates in a plane, and the plane is determined by the Earth's center, the incidence point and incidence direction of light, as shown in Fig. 10 or Fig. 14. In the following numerical examples, the propagation path is simulated and analyzed by applying Eqs. (47) and (60).

When an observer on Earth sees the Sun, the rays from the Sun pass through the thick atmosphere, and then focus on the observer's eyes. The Sun's visual shape is determined by the rays from the upper and lower edges of the Sun; therefore, during the analysis, only the rays from the edges of the Sun are simulated and analyzed. Meanwhile, due to the reversibility of optical path, it is assumed that the rays travel from Dalian to the Sun. In addition, when the altitude in the atmosphere is above 50 km, its refractive index approaches 1; therefore, the atmosphere below the altitude of 50 km is only considered. If the Sun's rays propagate in a straight line, the observer on Earth has an angle of view of  $0.5334^\circ$ . As a result, when simulating the ray of the upper edge of the Sun, the angle constraint  $\tau_{\text{Sun}}$  is set to  $+0.2667^\circ$ , and when simulating the ray of the lower edge of the Sun,  $\tau_{\text{Sun}}$  is set to  $-0.2667^\circ$ . This means that when the rays reach the atmosphere at the altitude of 50 km, the angles between the rays and the x axis are restricted to be  $\pm 0.2667^\circ$ .

Dalian is taken as the observation position. The coordinates of Dalian are  $\vec{P}_0 [O\text{Acos}(\varphi_{\text{DL}}), O\text{Asin}(\varphi_{\text{DL}})]$ ,  $OA = 6371$  km, and the observation moments are in the morning, at noon and in the afternoon on one day in mid-December 2018. In order to determine the initial angle  $\hat{\theta}_0$  between the ray from Dalian and the radial direction of the Earth, the method of bisection is performed by setting the initial range of  $\hat{\theta}_0$  to  $[\varphi_{\text{DL}} - 2^\circ, \varphi_{\text{DL}} + 2^\circ]$  and constraining the propagation direction of the ray in the atmosphere at the altitude of 50 km as  $\tau_{\text{Sun}} = \pm 0.2667^\circ$ .  $\hat{\theta}_0^{\text{Up}}$  denotes the incident angle of the ray from the upper edge of the Sun at Dalian, which is the angle between the ray and the radial direction of the Earth;  $\hat{\theta}_0^{\text{Lo}}$  expresses the incident angle of the ray from the lower edge of the Sun at Dalian, which is the angle between the

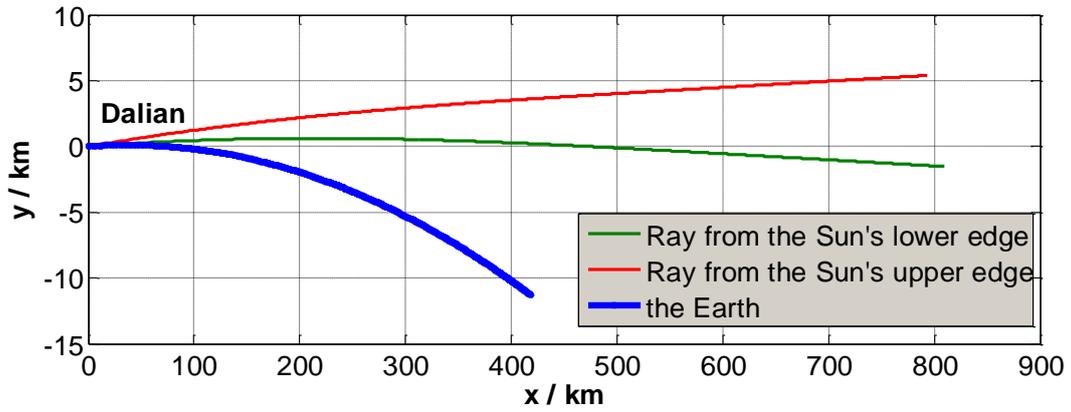


ray and the radial direction of the Earth. After performing the method of bisection,  $\hat{\theta}_0^{\text{Up}}$  and  $\hat{\theta}_0^{\text{Lo}}$  can be obtained by iteration; then the shape parameter of the Sun's visual shape is evaluated by

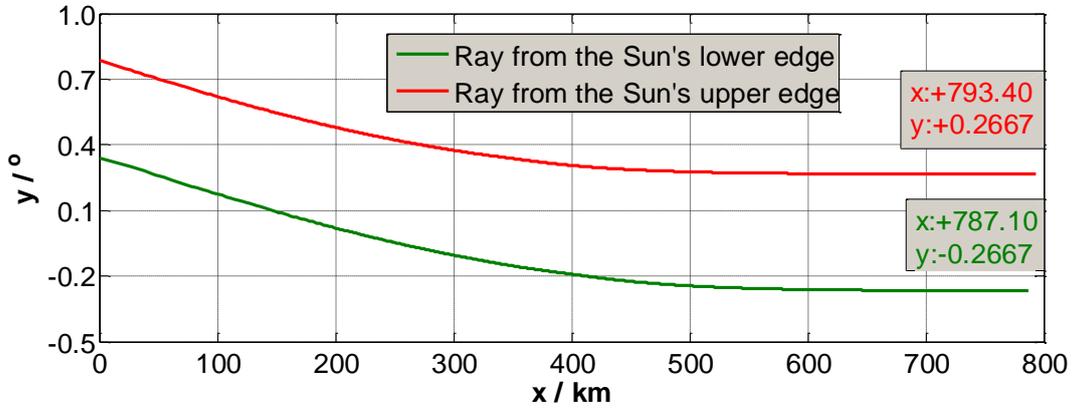
$$k_{\text{Ref}} = \frac{\hat{\theta}_0^{\text{Up}} - \hat{\theta}_0^{\text{Lo}}}{0.5334^\circ} \quad (74)$$

where,  $k_{\text{Ref}}$  describes the effect of atmospheric refraction on the Sun's visual shape.

Fig. 20 shows the propagation paths of the rays from the upper and lower edges of the Sun at 7:06 Beijing time on a day in mid-December 2018 at Dalian.



(a)



(b)

Fig. 20 Propagation paths of the rays at 07:06 on a day in mid-December at Dalian

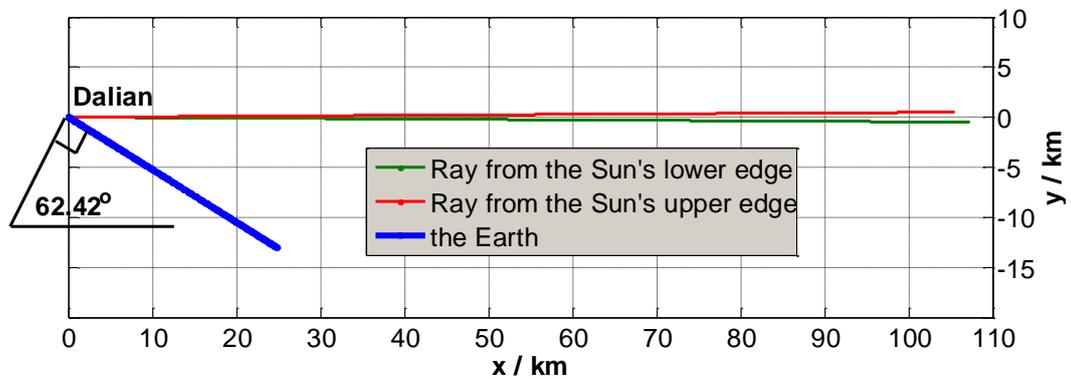
In Fig. 20(a), Dalian is set as the origin, the horizontal and vertical directions are the same as the X and Y axes in Fig. 10. From the figure, it can be found that the horizon is basically tangent to the ray from the lower edge of the Sun in the morning.

Fig. 20(b) illustrates bending deformations of the rays in the atmosphere, where the horizontal axis is the horizontal distance between the observer and the Sun, and the vertical axis is the angle between the propagation direction of the ray and the line

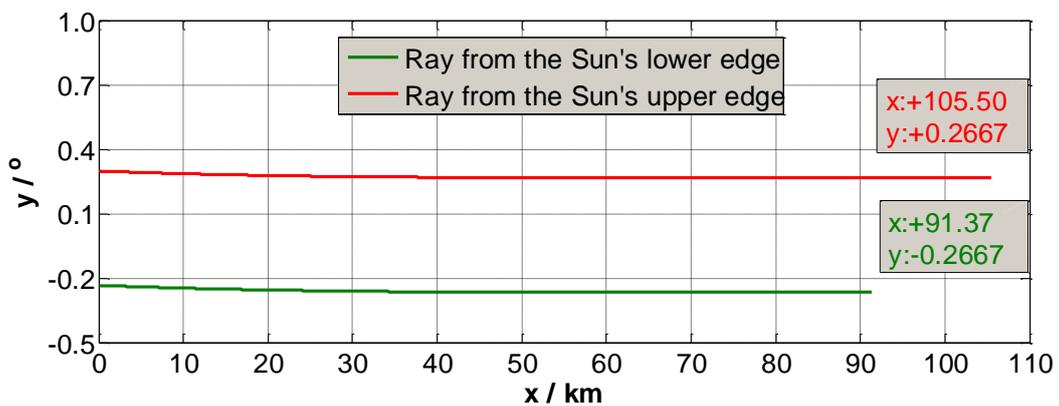


connecting the Sun's and the Earth's centers.  $(+793.40, +0.2667)$  and  $(+787.10, -0.2667)$  are the coordinates of the right end points of the two curves, i.e. the coordinates of the rays of the upper and lower edges of the Sun at the altitude of 50 km. When the rays reach an observer at Dalian, his angle of view becomes  $0.4457^\circ$ , that is  $\hat{\theta}_0^{\text{Up}} - \hat{\theta}_0^{\text{Lo}} = 0.4457^\circ$ . According to Eq. (74), the shape parameter  $k_{\text{Ref}} = 0.8357$ , which means that the Sun in the observer's eyes is flattened in the vertical direction.

Fig. 21 shows the propagation paths of the rays from the upper and lower edges of the Sun at 11:40 Beijing time on a day in mid-December 2018 at Dalian.



(a)



(b)

Fig. 21 Propagation paths of the rays at 11:40 on a day in mid-December at Dalian

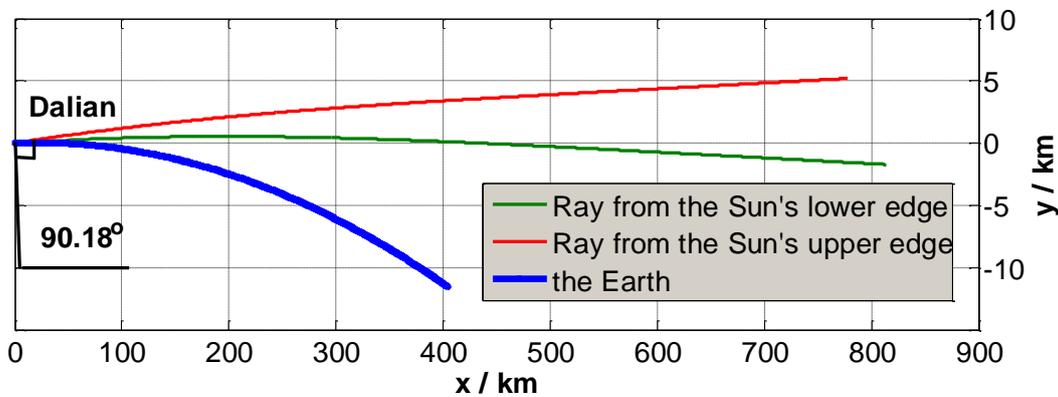
At noon, the Sun shines overhead in Dalian. The angle  $\varphi_{\text{DL}}$  between the line connecting Dalian and the Earth's center and the x axis is  $62.42^\circ$ , as shown in Fig. 21(a). In Fig. 21(b),  $(+105.50, +0.2667)$  and  $(+91.37, -0.2667)$  are the coordinates of the right end points of the two curves, i.e. the coordinates of the rays of



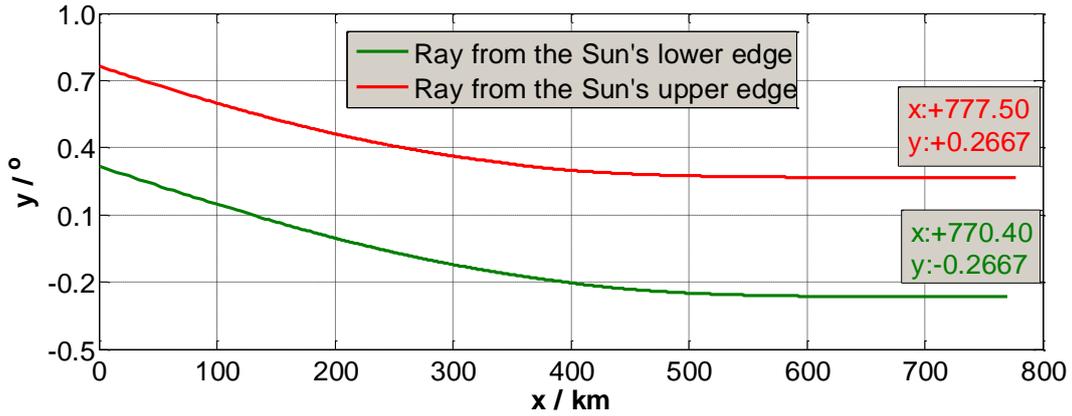
the upper and lower edges of the Sun at the altitude of 50 km.

Compared with the morning, the propagation distance of the rays from the edge of the Sun becomes short in the 50-km-thick atmosphere. Meanwhile, the angles between the rays from the Sun and the Earth's radial direction passing through Dalian (point A) also become small. As a result, bending deformations of the rays in the atmosphere are weakened, which can easily be observed in Fig. 21(b). When the rays reach Dalian at noon, an observer's angle of view becomes  $0.5327^\circ$ , bigger than  $0.4457^\circ$  (in the morning) and a little less than  $0.5334^\circ$  (the angular diameter of the Sun). According to Eq. (74), the shape parameter  $k_{\text{Ref}} = 0.9986$ , which means that the Sun's image is rounder than that in the morning.

Fig. 22 shows the propagation paths of the rays from the upper and lower edges of the Sun at 16:28 Beijing time on a day in mid-December 2018 at Dalian. Similarly to the situation in the morning, the angle between the line connecting Dalian and the Earth's center and the x axis is  $\varphi_{\text{DL}} = 90.18^\circ$ , as shown in Fig. 22(a). In Fig. 22(b),  $(+777.50, +0.2667)$  and  $(+770.40, -0.2667)$  are the coordinates of the right end points of the two curves, i.e. the coordinates of the rays of the upper and lower edges of the Sun at the altitude of 50 km. At this moment, an observer on the Earth can see the whole image of the Sun; however, the fact is that the whole Sun have almost gone under the horizon. The observer's angle of view is  $0.4471^\circ$ , approximate to  $0.4457^\circ$  (in the morning). According to Eq. (74), the shape parameter  $k_{\text{Ref}} = 0.8382$ , which means that the Sun's image becomes flattened again.



(a)



(b)

Fig. 22 Propagation paths of the rays at 16:28 on a day in mid-December at Dalian

In order to further investigate the effects of atmospheric refraction on the Sun's visual shape in the whole year, a variation of the shape parameter is given in Fig. 26, where the horizontal axis is the date in the whole year of 2018, and the vertical axis is the shape parameter  $k_{\text{Ref}}$ , which is calculated by Eq. (74). During the simulation, the observation time is set as 12:00:00 Beijing time every day, the observation position is Dalian. Fig. 23 shows that the Sun's visual shape is the roundest at summer solstice, and the flattest at winter solstice.

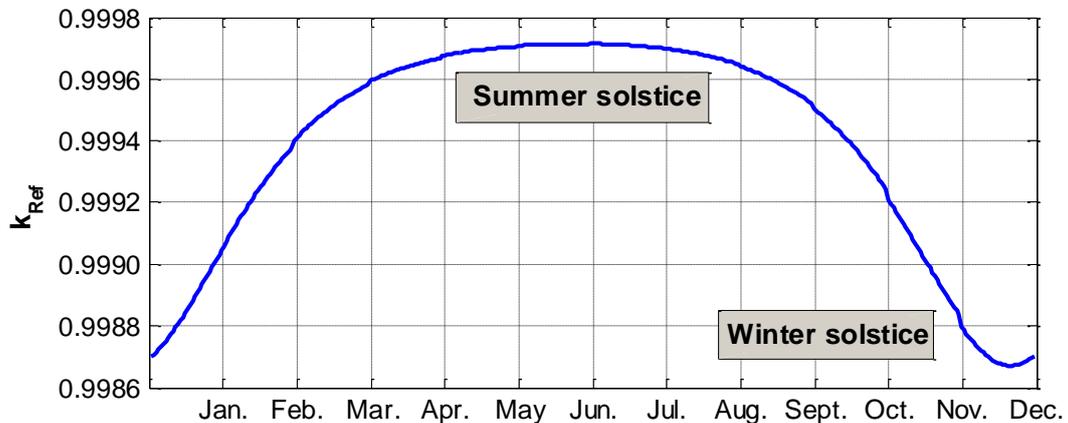


Fig. 23 Variation of the shape parameter  $k_{\text{Ref}}$  in 2018 at Dalian

#### 4.2 Comparison and verification

When researching the effect of the Einstein's theory of special relativity on the Sun's visual shape, it is proved that the contraction ratio occurring in the Sun's image varies slowly and little with time. For example, the shape parameter  $k_{\text{SRT}}$  ranged



from 0.99934 to 0.99948 in the whole year of 2018, as shown in Fig. 18. However, the contraction direction or the semi-minor axis direction in the Sun's ellipse image varies rapidly and greatly with time. For example, the angle  $\psi_{bH}$  between the horizontal direction of the image and the semi-minor axis of the Sun's ellipse image ranged from  $47^\circ$  to  $-47^\circ$  on one day of mid-December 2018, as shown in Fig. 19.

When researching the effect of atmospheric refraction on the Sun's visual shape, it was shown that the shape parameter  $k_{Ref}$  of the Sun's ellipse image changes greatly. For example,  $k_{Ref}$  ranged from 0.8357 to 0.9986 on one day of mid-December 2018, as shown in Figs. 20, 21 and 22. However, the contraction direction of the Sun's ellipse image caused by the atmospheric refraction remains unchanged. Because Eq. (57) proves that the propagation path of light in the atmosphere is a planar curve, the semi-minor axis of the Sun's ellipse image is always perpendicular to the horizontal direction of the image.

Assuming that the Sun is a circle, by applying the elementary transformation, it can be shown that after the Einstein's theory of special relativity and atmospheric refraction effects are applied to the circle, the circle will become an ellipse, and the shape parameter  $k$  of the ellipse can be calculated by

$$k = \begin{cases} \frac{\Theta_1 \cos(2\alpha) - \Theta_2}{\Theta_1 \cos(2\alpha) + \Theta_2} & \alpha \neq \frac{\pi}{2} \\ \frac{\Theta_1 \sin(2\alpha) - \Theta_3}{\Theta_1 \sin(2\alpha) + \Theta_3} & \alpha \neq 0 \end{cases} \quad (75)$$

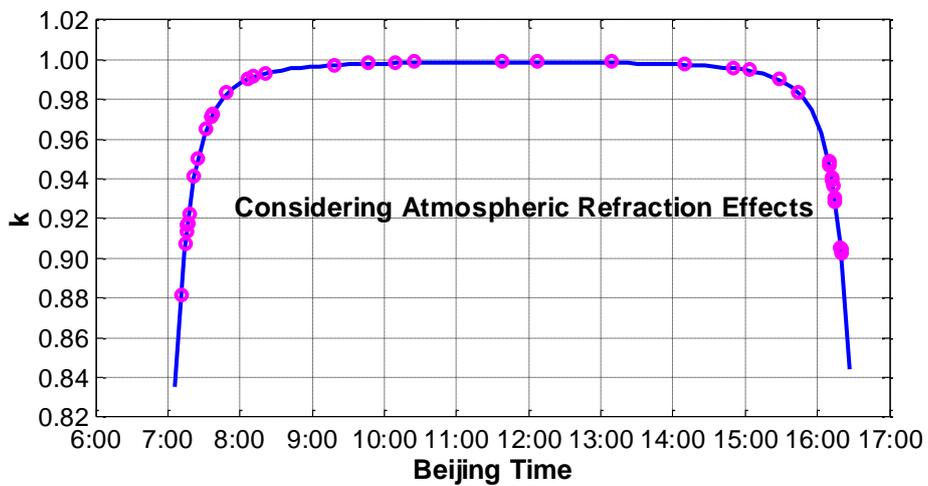
where

$$\begin{cases} \Theta_1 = 1 + p \cdot q \cos(2\psi_{bH}) \\ \Theta_2 = q + p \cdot \cos(2\psi_{bH}) \\ \Theta_3 = p \cdot \sqrt{1 - q^2} \cdot \sin(2\psi_{bH}) \end{cases} \quad \begin{cases} p = \frac{1 - k_{SRT}^2}{1 + k_{SRT}^2} \\ q = \frac{k_{Ref}^2 - 1}{1 + k_{Ref}^2} \end{cases} \quad \tan(2\alpha) = \frac{\Theta_3}{\Theta_2} \quad (76)$$

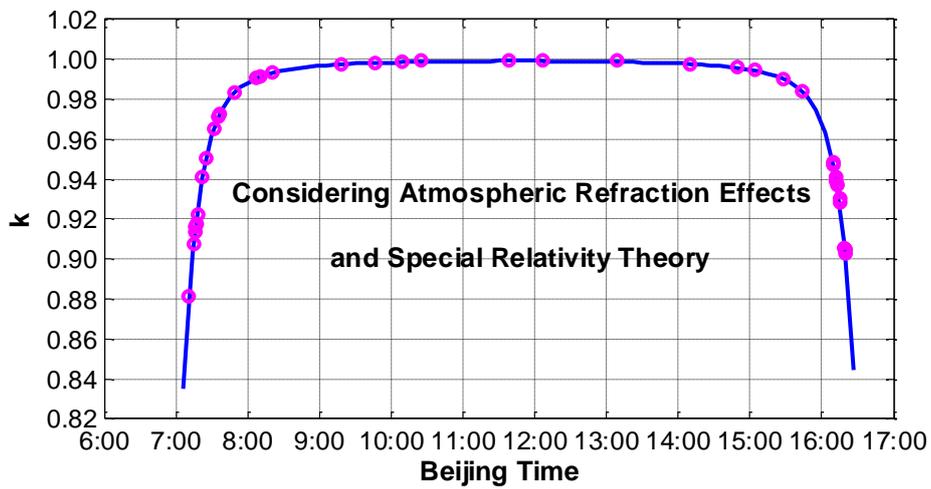
Based on Eq. (70), the shape parameter  $k_{SRT}$  can be figured out, which describes the effect of the Einstein's special relativity on the Sun's visual shape; based on Eq. (74), the shape parameter  $k_{Ref}$  can be evaluated, which describes the effects of atmospheric refraction on the Sun's visual shape; combining Eqs. (70), (73), (74) and (75), the shape parameter  $k$  can be calculated, which describes the effects of atmospheric refraction and the Einstein's special relativity on the Sun's visual shape.



In order to show the reasons for the variation of the Sun's visual shape, the theoretical simulation results and the experimental results were compared, as shown in Figs. 24, 25 and 26. The horizontal axis is Beijing time from sunrise to sunset; the vertical axis is the shape parameter, and the pink circles are the experimental results of Tables 3, 4 and 5. In Figs. 24(a), 25(a) and 26(a), the blue line stands for the simulation results by only considering atmospheric refraction; in Figs. 24(b), 25(b) and 26(b), the blue line stands for the simulation results by considering the combination of atmospheric refraction and the Einstein's special relativity.

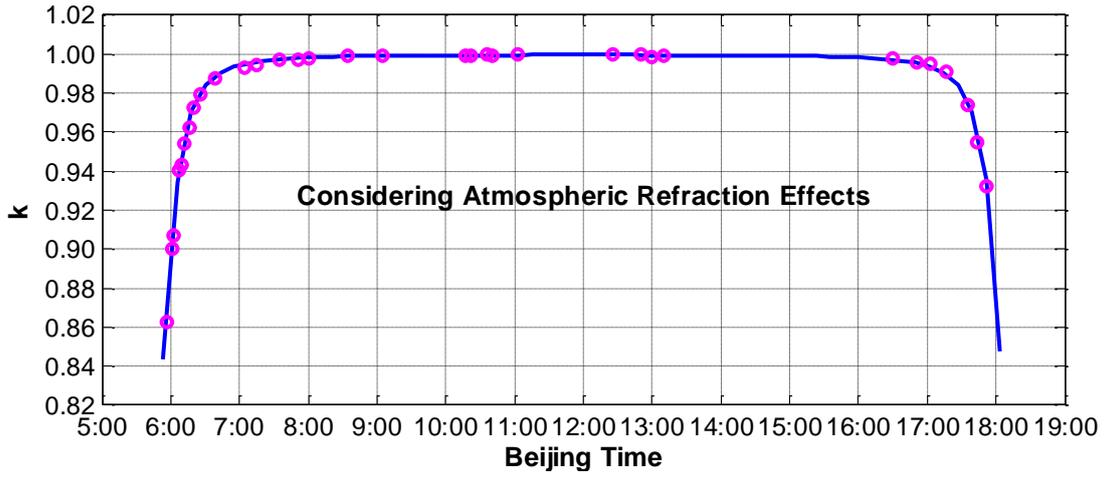


(a)

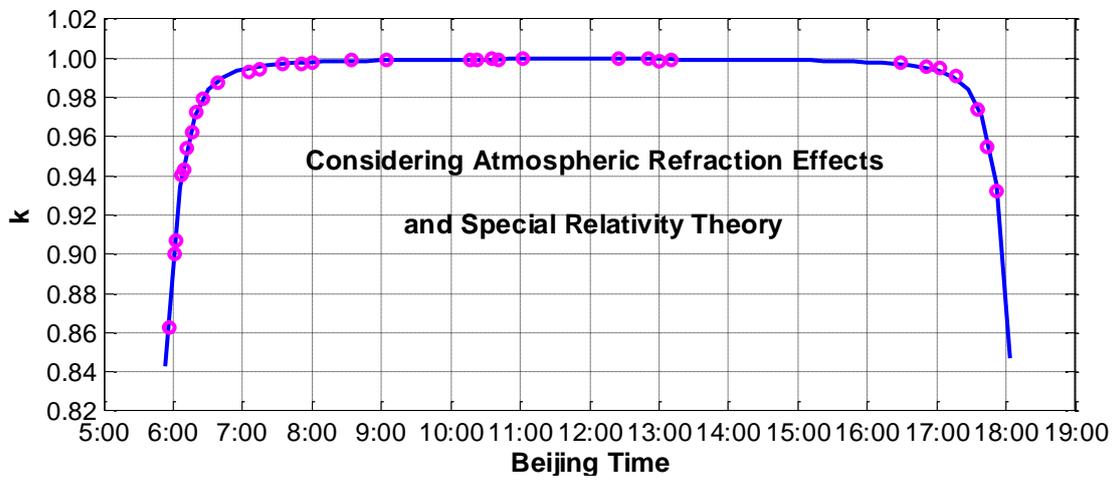


(b)

Fig. 24 Comparison between simulation and experimental results on a day in mid-December 2018 at Dalian

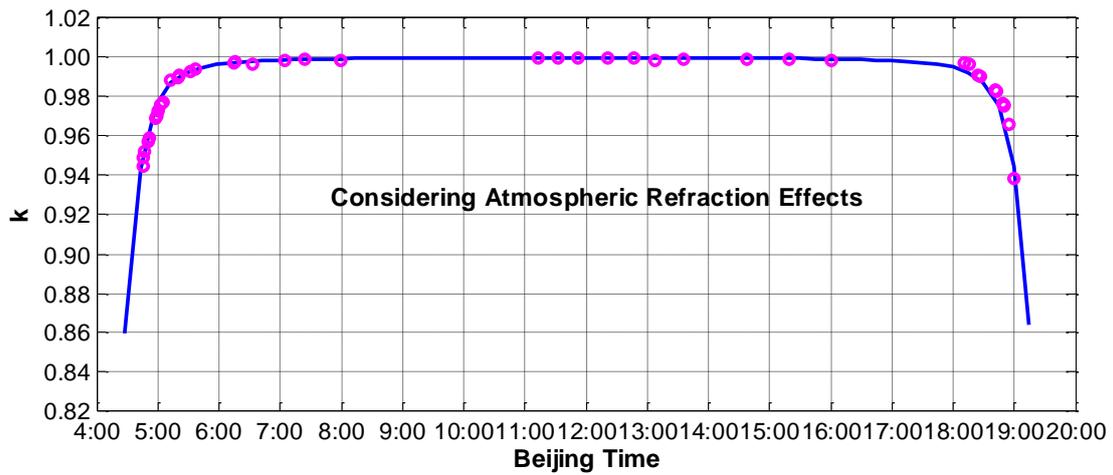


(a)

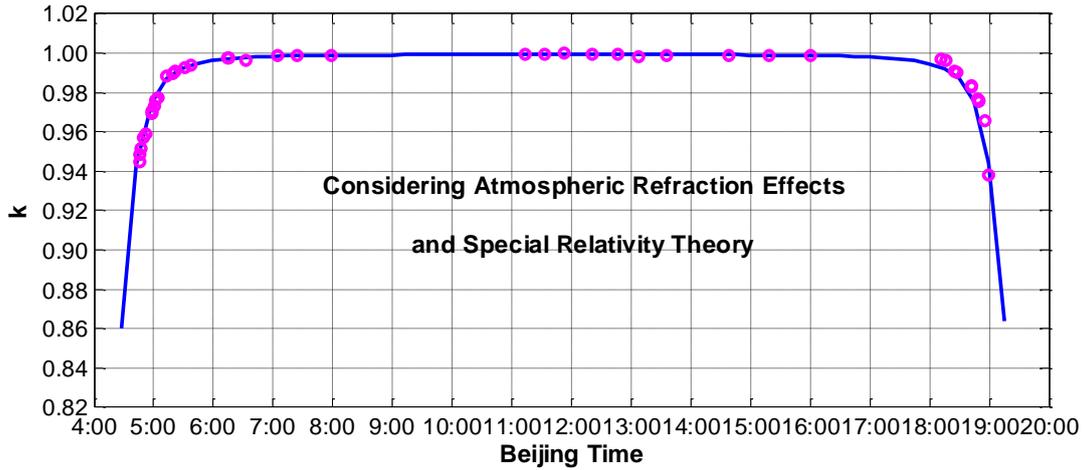


(b)

Fig. 25 Comparison between simulation and experimental results on a day in late March 2019 at Dalian



(a)

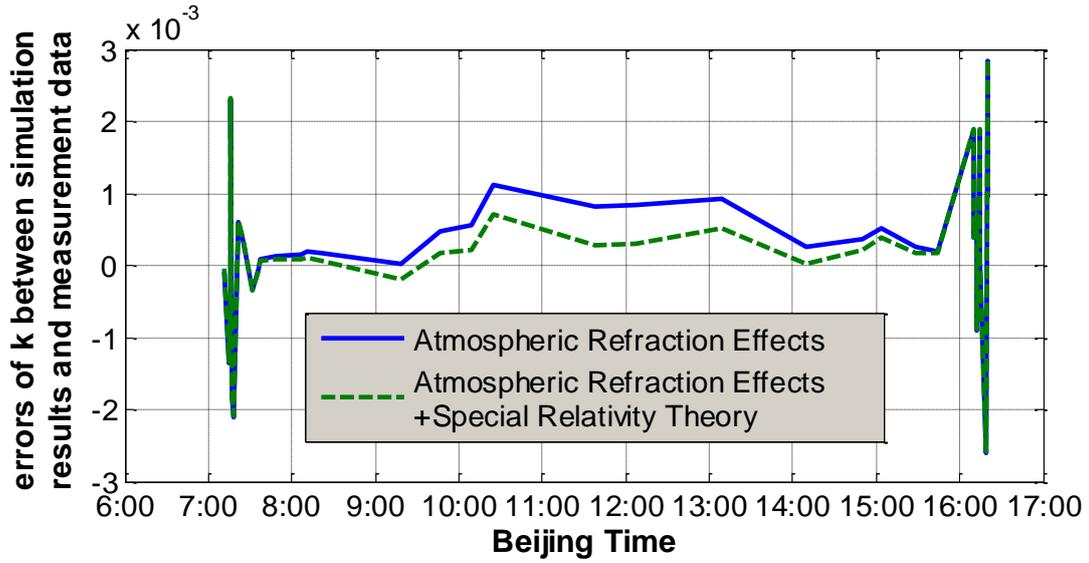


(b)

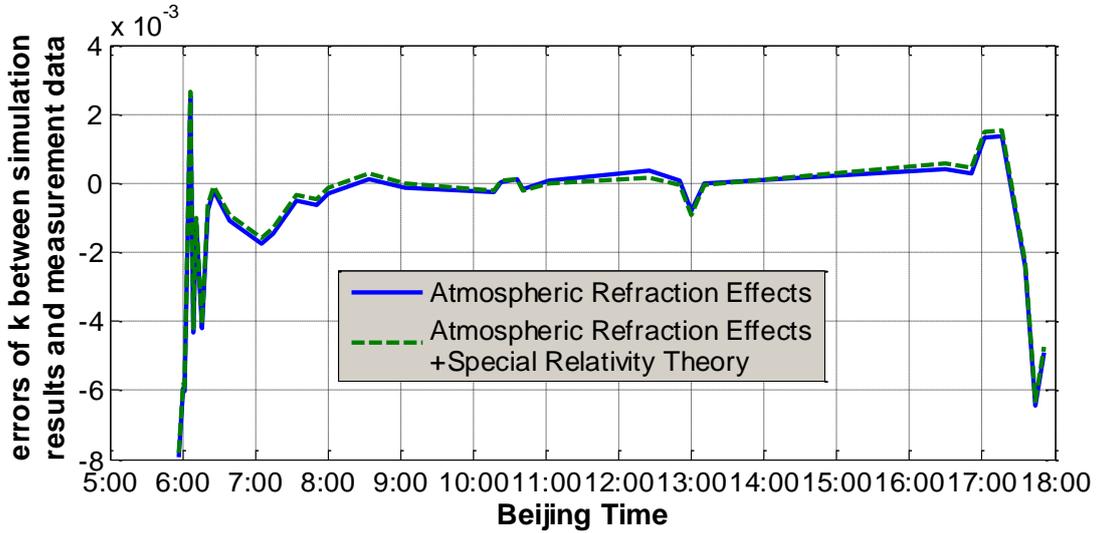
Fig. 26 Comparison between simulation and experimental results on a day in late June 2019 at Dalian

The relative error is defined as the difference between simulations and experimental results over the experimental results. Fig. 27 shows the relative errors of the simulated results, where the blue line is the errors calculated when only considering atmospheric refraction; the green dotted line is the errors calculated when considering the combination of atmospheric refraction and the Einstein's theory of special relativity.

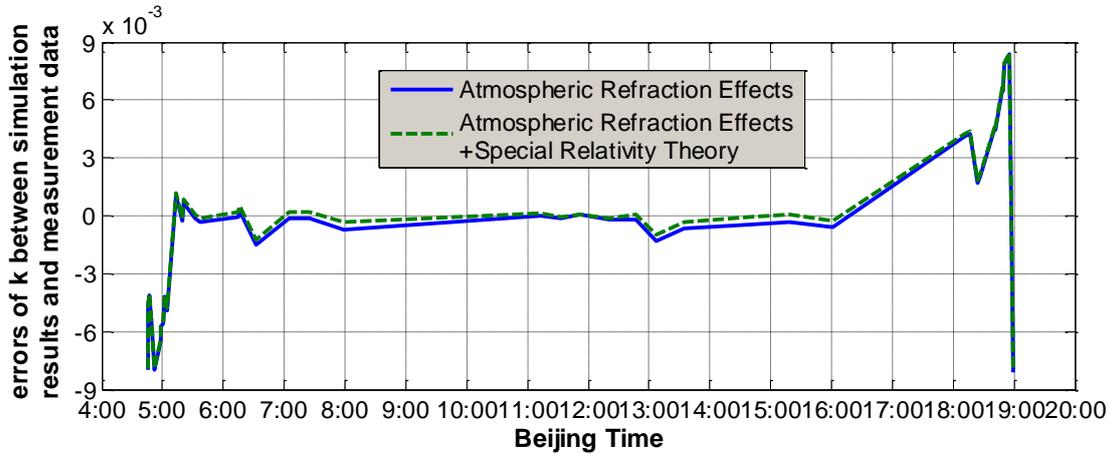
Fig. 27 shows that the Einstein's special relativity affects the Sun's visual shape much more at noon than at sunrise and sunset. Taking Fig. 27(a) for example, at 11:38 Beijing time on one day in mid-December 2018, the shape parameter extracted from the Sun's photo is  $k_{\text{Meas}} = 0.9995$ ; the shape parameter caused by atmospheric refraction is  $k_{\text{Ref}} = 0.9987$  with a relative error of  $0.8 \times 10^{-3}$ ; the shape parameter calculated by considering the combination of atmospheric refraction and the Einstein's special relativity is  $k = 0.9993$  with a relative error of  $0.2 \times 10^{-3}$ . Although the relative error of  $k_{\text{Ref}}$  is about four times than that of  $k$ , the error is still very small and can be neglected. Fig. 27 illustrates that the simulated results are coincident with the experimental ones of Tables 3, 4 and 5.



(a) On a day in mid-December 2018 at Dalian



(b) On a day in late March 2019 at Dalian



(c) On a day in late June 2019 at Dalian

Fig.27 Relative Errors of the simulation results



The comparison study showed that the theoretical analyses in the paper are correct; the reason for the variation of the Sun's visual shape is mainly due to atmospheric refraction, while the length contraction effect of the Einstein's special relativity also contributes a little except at sunrise and sunset. Therefore, the Einstein's special relativity can explain why the Sun's visual shape always appears to be an ellipse instead of a perfect circle.

In order to further analyze the contributions of atmospheric refraction and the Einstein's special relativity to the variation of the Sun's visual shape, three variations of the shape parameter were simulated in Fig. 28, where the horizontal axis is the date in the whole year of 2018, and the vertical axis is the shape parameter. The blue line stands for  $k_{\text{Ref}}$  caused by atmospheric refraction, similar to the curve in Fig. 23; the pink dotted line stands for  $k_{\text{SRT}}$  caused by the length contraction effect in the Einstein's special relativity, which is the same with the curve in Fig. 18; the black line is the shape parameter calculated when considering the combination of atmospheric refraction and the Einstein's special relativity. During the simulation, the observation time was set at 12:00:00 Beijing time every day, and the observation position was Dalian.

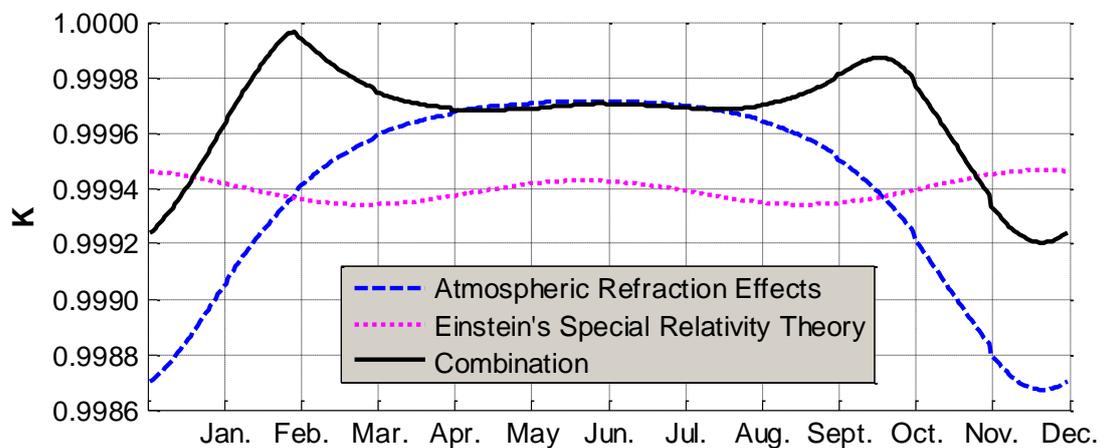


Fig.28 Variation of the shape parameter in 2018 at Dalian

Fig. 28 shows that in the whole year, the Sun's visual shape at Dalian is the roundest at the end of February, and is the flattest at winter solstice. The conclusion differs from that drawn from only considering atmospheric refraction or the Einstein's special relativity, because both of them contribute to the variation of the Sun's visual shape.



## Chapter 5: Conclusions

From sunrise to sunset, the Sun's visual shape changes continuously from a flatter ellipse to almost a circle and then back to a flatter ellipse. In the paper, the experimental measurements and theoretical analyses were performed to investigate the reasons for the variation of the Sun's visual shape. Some meaningful conclusions were drawn:

The method of image processing, the method of moments and the least-square method were combined to perform experimental measurements and calculations. The statistical error analyses showed that the relative measurements accuracy was about 0.023%, and the average residuals of the fitting curve of ellipse was only 4 pixels. The experimental results showed that the Sun's visual shape can be approximated by an ellipse accurately.

Atmospheric refraction effects make the Sun's visual shape become an ellipse, and have a great influence on its shape during a day. The refraction index of the atmosphere was expressed as a function of altitude and wavelength of light. Because the trajectory of sunlight is a planar curve, when considering the atmospheric refraction effects only, the Sun's visual shape contracts only in the zenith direction, resulting in an elliptic Sun. In the Sun's photo, the contraction direction remains unchanged, but the contraction ratio varies rapidly and greatly with time, especially at sunrise and sunset.

The length contraction effect in the Einstein's special relativity also transforms the Sun's visual shape to an ellipse, and contributes a little to its shape variation on a day, especially at noon. Due to the relative movement of the Sun to the observer on the Earth, the Sun's visual shape contracts in a direction, resulting in an elliptic Sun. However, unlike atmospheric refraction effects, the contraction ratio varies slowly and little with time, and it is about 0.06%, while its contraction direction varies rapidly and greatly with time.

Comprehensively considering effects of atmospheric refraction and the Einstein's special relativity on the Sun's visual shape, although their contraction directions are



different, it was shown that the Sun's visual shape is still an ellipse.

In a day, the Sun's visual shape is the roundest at noon and the flattest at sunrise and sunset; in the whole year, the Sun's visual shape for an observer at Dalian is the roundest at the end of February, and the flattest at winter solstice. For observers at different geographic positions, if they see the Sun at the same moment, the Sun looks different. In theory, an observer at the equator could see the roundest Sun.

Comparing the theoretical simulations and experimental measurements, the relative errors were less than 0.8%. The results, thus, verified the theoretical analyses in the paper, including a set of formulas describing the relationship between the zenith angle of the Sun, the observation time and positions, the model of altitude and refraction index, the iterative algorithm for tracing rays of light in the atmosphere, and the calculation of feature parameters of the Sun's visual shape.



## References

- Balal Yael, Pinhasi Yosef (2019) Atmospheric Effects on Millimeter and Sub-millimeter (THz) Satellite Communication Paths. *Journal of Infrared Millimeter and Terahertz Waves* 40(2):219-230
- Born, M., Wolf, E (2007) Principles of optics: Electromagnetic theory of propagation, interference and diffraction of light. Seventh (Expanded) edition. Publishing house of electronics industry
- Chaim Keller, John K. Hall (2000) Using a digital terrain model to calculate visual sunrise and sunset times. *Computers & Geosciences* 26(9-10): 991-1000
- Jiang Dagang, Deng Ke, Huang Jian, Zhu Bin, Qin Kaiyu (2013) Influence of atmospheric refraction on leo-ground laser communication and ranging link. *Manned Spaceflight* 19(5): 85-89
- Kambezidis Harry D (1997) Estimation of sunrise and sunset hours for locations on flat and complex terrain: Review and advancement. *Renewable Energy* 11(4): 485-494
- Kambezidis Harry D, Papanikolaou N S (1990) Solar position and atmospheric refraction. *Solar Energy* 44(3): 143-144
- Kambezidis Harry D, Tsangrassoulis A E (1993) Solar position and right ascension. *Solar Energy* 50(5): 415-416
- Karin Hedman, Florian David (2004) Atmospheric refraction and its impact on free-space optic communication systems. *Proceedings of SPIE. Canary Islands Spain November 5572*: 199-210
- Sproul Alistair B (2007) Derivation of the solar geometric relationships using vector analysis. *Renewable Energy* 32(7): 1187-1205
- Wang Hongyan, Wang Gaili, Liu Liping (2018) Climatological Beam Propagation Conditions for China's Weather Radar Network. *Journal of Applied Meteorology and Climatology* 57(1):3-14
- Xiang Jing-song (2008) Pointing error induced by atmospheric refraction and atmospheric chromatic dispersion for earth-to-satellite laser communication. *Journal Of Chongqing University Of Posts and Telecommunications (Natural Science Edition)* 20(4): 423-426
- Yuan Hongwu, Mei Haiping, Huang Yinbo, Rao Ruizhong (2011) Research on



国际竞赛 科研科创 发表论文  
关注“有方背景提升”

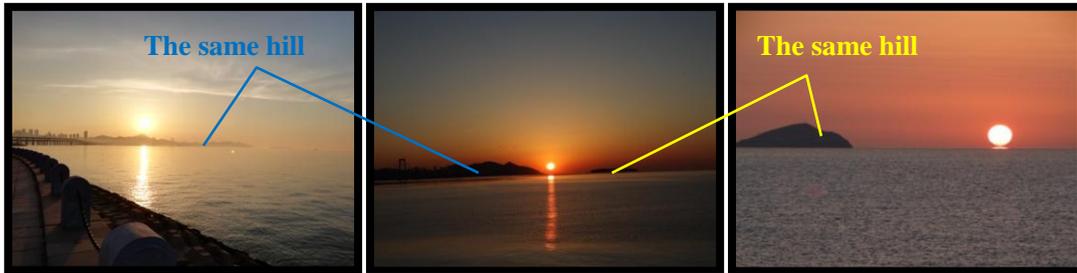
atmospheric refraction correction algorithm and model for satellite laser range-finding.

Acta Optica Sinica 31(4):21-29



## Experiment Time and Locations

During the experimental measurements, a high-resolution digital camera was used to take photos of the Sun from sunrise to sunset. The observation time included before winter solstice in 2018 (mid-December), after spring equinox in 2019 (late March) and around summer solstice in 2019 (late June). The observation locations were in Dalian, a city in northeast China, including Xinghai Bridge (at sunrise), Dalian University of Technology (at noon) and Jinshitan Beach (at sunset).



In June 2019

In March 2019

In December 2018

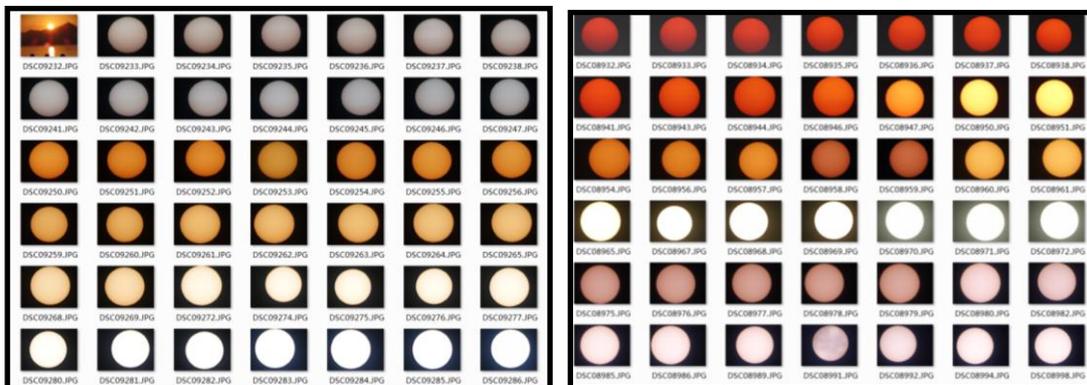
The sunrise at Xinghai Bridge (the location of the Sun varies with seasons)



At Dalian University of Technology



At Jinshitan Beach



Photos of the Sun



## Acknowledgements

I sincerely appreciate my volunteer supervisors Brahim El Fadil and Zhisong Lian, who are my teachers at Dalian Maple Leaf International School. Brahim El Fadil taught me AP Mechanics and AP Electricity & Magnetism. It was his homework project that inspired my interest in the Sun. The homework was assigned in September 2018, and was about "Deep understanding and critical thinking: find an example in physics that shows the evidence of the length contraction, Explain". I studied the Einstein's Special Relativity, but I did not find a convincing example until 4th October 2018, when I flied to take my SAT Subject Tests in Hong Kong. Seeing blue sky and white cloud through the airplane window, all of a sudden, an idea occurred to me. That was the Sun's visual shape could be an evidence of the length contraction of the Einstein's Special Relativity. I began investigating the Sun ever since. Zhisong Lian was my mathematics teacher, and taught me AP Calculus and AP Statistics. Without knowledge of Calculus and Statistics, I could not finish my investigation into the Sun. During the investigation, Brahim El Fadil and Zhisong Lian gave me lots of advice and instructions; and in order to help me taking advice from my classmates, they also gave me a chance to introduce my investigation at class.

Meanwhile, I want to say "thank you" to Jeffrey Wong, who was my teacher of AP Computer Science A and AP Computer Science Principles. Wong was not my research supervisor, however, I also got his support. Although he taught me Java language, he still allowed me to develop my MATLAB program in his class and took the MATLAB program as one of my homework project. Wong's support made me have more time to do the investigation.

Also, I thank my parents for their unconditioned love and assistance. My parents work at Dalian University of Technology. When I encountered difficulties, my father helped me search related research paper in library database of DUT and discussed with me. In order to make me observe the relative movement of the Earth, the Sun and the Moon, my mother bought me an electric three-star model. In order to take the



photos of the Sun from sunrise to sunset, my mother usually drove me to different places in one day.

I do appreciate everyone who lent me a hand. Their instruction, encouragement and love help me overcome lots of difficulties and finally complete the investigation.



The electric three-star model



## Resume of Yuanxiu Wang

### Education:

- ✓ Sep. 2017 ---- to now, 2017 ZEL Scholarship Class, Dalian Maple Leaf International School, Dalian, Liaoning, China, High school student.
- ✓ Sep. 2014 ---- Aug. 2017, Attached School of Dalian University of Technology, Dalian, Liaoning, China, Middle school student.
- ✓ Sep. 2013 ---- Aug. 2014, Lux Middle School, Lincoln, Nebraska, USA, Middle school student.
- ✓ Sep. 2010 ---- Aug. 2013, Attached School of Dalian University of Technology, Dalian, Liaoning, China, Elementary school student.
- ✓ Sep. 2009 ---- Aug. 2010, J. J. Finley Elementary School, Gainesville, Florida, USA, Elementary school student.
- ✓ Sep. 2008 ---- Aug. 2009, Attached School of Dalian University of Technology, Dalian, Liaoning, China, Elementary school student.

### Activities:

- ✓ From Grade 7 to Grade 10, as a sole author, I wrote a 100-thousand-word book to introduce my school life of the USA, and the title is "最美是上学——亲历美国中小学精英教育". The book was financed and published by the Hunan Art and Literature Press in June 2018; and from then on, the book has been sold in nationwide bookstores and on the internet.
- ✓ In Grade 11, I investigated the variation of the Sun's visual shape, and as a sole author, wrote a research paper "Investigating the Variation of the Sun's Visual Shape, Atmospheric Refraction and Einstein's Special Relativity Considered". The paper was submitted to an international SCI Journal, "Earth Science Informatics". On July 31 2019, I received positive comments from reviewers, who said, "The paper deals with a theoretical but interesting subject, that of proving that the visual shape of the Sun is an ellipse". In mid-August 2019, the



revised submission was completed.

- ✓ In Grade 10, I investigated water environment of Dalian, and under the guidance of Mr. Hughes and Dr. Bing Liu, did a set of experiments at Dalian University of Technology. Based on the work, as a sole author, I wrote a conference paper "Investigation and Analysis of Water Environment of Dalian", which was published in November 2018 and indexed by EI Compendex. View the paper online <https://doi.org/10.1088/1755-1315/189/5/052023>.
- ✓ In Grade 10, I designed two types of humidifiers to purify the air in my room, and as a sole inventor, applied for China Patents for Utility Models. The two inventions were granted in June 2018; one title is "Semiconductor condensation type air purification humidifier", the other "Air humidifying standing wave clarifier"; and their patent numbers are 201721187294.9 and 201721187313.8.
- ✓ In Grade 8, I designed an elastic belts device to assist myself to practice sit-ups, which can also be used to help some elder or sick people get up easily; and as a sole inventor, applied for a China Invention Patent. The invention was granted in September 2018; the title is “Wearable artificial abdominal muscle”, and its patent number is 201610429354.7.
- ✓ In Grade 7, I designed a device comprised of mirrors, optical fiber bundles and lenses to eliminate blind area of the right rearview mirror of my mother's car; and as the first inventor, applied for a China Invention Patent. The invention was granted in May 2017; the title is “Passive vehicle rearview mirror without blind area”, and its patent number is 201410841151.X.
- ✓ In Grade 5, I passed the electronic keyboard examination of grade 10 of China.



# Paper Detection Report




## 文本复制检测报告单(全文对照)

---

**Ne:ADBD2019R\_2019051016380020190822125601301895303716** 检测时间:2019-08-22 12:56:01

**检测文献:** 80202914692224579\_Yuanxiu Wang\_ Investigating the Variation of the Sun's Visual Shape  
**作者:** Yuanxiu

**检测范围:** 中国学术期刊网络出版总库  
 中国博士学位论文全文数据库/中国优秀硕士学位论文全文数据库  
 中国重要会议论文全文数据库  
 中国重要报纸全文数据库  
 中国专利全文数据库  
 图书资源  
 优先出版文献库  
 学术论文联合比对库  
 互联网资源(包含贴吧等论坛资源)  
 英文数据库(涵盖期刊、博硕、会议的英文数据以及德国Springer、英国Taylor&Francis 期刊数据库等)  
 港澳台学术文献库  
 互联网文档资源  
 CNKI大成编客-原创作品库  
 个人比对库

**时间范围:** 1900-01-01至2019-08-22

---

**检测结果**

去除本人已发表文献复制比: ■ 0.1% 跨语言检测结果: 0%  
 去除引用文献复制比: 0.1% 总文字复制比: 0.1%  
 单篇最大文字复制比: 0.1% ( [Vision Based Method for the Localization of Intelligent Vehicles in Loose Constraint Area](#) )

---

重复字数:	[67]	总段落数:	[5]
总字数:	[71420]	疑似段落数:	[1]
单篇最大重复字数:	[67]	前部重合字数:	[0]
疑似段落最大重合字数:	[67]	后部重合字数:	[67]
疑似段落最小重合字数:	[67]		



指标:  疑似剽窃观点  疑似剽窃文字表述  疑似自我剽窃  疑似整体剽窃  过度引用

表格: 0    公式: 0    疑似文字的图片: 0    脚注与尾注: 0

<span style="color: yellow;">■</span> 0.4% ( 67 )	80202914692224579_Yuanxiu Wang_ Investigating the Variation of the Sun's Visual Shape_第1部分 ( 总16449字 )
<span style="color: green;">■</span> 0% ( 0 )	80202914692224579_Yuanxiu Wang_ Investigating the Variation of the Sun's Visual Shape_第2部分 ( 总16393字 )
<span style="color: green;">■</span> 0% ( 0 )	80202914692224579_Yuanxiu Wang_ Investigating the Variation of the Sun's Visual Shape_第3部分 ( 总16516字 )
<span style="color: green;">■</span> 0% ( 0 )	80202914692224579_Yuanxiu Wang_ Investigating the Variation of the Sun's Visual Shape_第4部分 ( 总16395字 )
<span style="color: green;">■</span> 0% ( 0 )	80202914692224579_Yuanxiu Wang_ Investigating the Variation of the Sun's Visual Shape_第5部分 ( 总5667字 )



( 注释: ■ 无问题部分 ■ 文字复制部分 ■ 引用部分 )

---

**1. 80202914692224579\_Yuanxiu Wang\_ Investigating the Variation of the Sun's Visual Shape\_第1部分** 总字数: 16449

**相似文献列表**

去除本人已发表文献复制比: 0.4%(67)    文字复制比: 0.4%(67)    疑似剽窃观点: (0)

1   <a href="#">Vision Based Method for the Localization of Intelligent Vehicles in Loose Constraint Area</a> Xiaonan Liu;Guangming Xiong;Jianwei Gong;Yu Han;Huiyan Chen; - 《Proceedings of 2016 IEEE International Conference on Vehicular Electronics and Safety》 - 2016-07-10	0.4% ( 67 ) <b>是否引证: 否</b>
--	-------------------------------

- 1 -



原文内容		相似内容来源
1	<p>此处有 67 字相似</p> <p>e local coordinate system C-XeYe, as shown in Fig. 3. In O-XY, X and Y are along the horizontal and vertical directions of the image, respectively; in C-XeYe, the original point is set as the</p>	<p>Vision Based Method for the Localization of Intelligent Vehicles in Loose Constraint Area Xiaonan Liu; Guangming Xiong; Jianwei Gong; Yu Han; Huiyan Chen; - (Proceedings of 2016 IEEE International Conference on Vehicular Electronics and Safety) - 2016-07-10 (是否引证: 否)</p> <p>1. direction of the image. denote the Gauss filter operator. lx and ly is along the horizontal and vertical direction of the image, but the corner may be produced because of the edge of other</p>
<p>2. 80202914692224579_Yuanxiu Wang_ Investigating the Variation of the Sun's Visual Shape_第2部分 总字数: 16393</p> <p>相似文献列表</p> <p>去除本人已发表文献复制比: 0%(0) 文字复制比: 0%(0) 疑似剽窃观点: (0)</p>		
<p>3. 80202914692224579_Yuanxiu Wang_ Investigating the Variation of the Sun's Visual Shape_第3部分 总字数: 16516</p> <p>相似文献列表</p> <p>去除本人已发表文献复制比: 0%(0) 文字复制比: 0%(0) 疑似剽窃观点: (0)</p>		
<p>4. 80202914692224579_Yuanxiu Wang_ Investigating the Variation of the Sun's Visual Shape_第4部分 总字数: 16395</p> <p>相似文献列表</p> <p>去除本人已发表文献复制比: 0%(0) 文字复制比: 0%(0) 疑似剽窃观点: (0)</p>		
<p>5. 80202914692224579_Yuanxiu Wang_ Investigating the Variation of the Sun's Visual Shape_第5部分 总字数: 5667</p> <p>相似文献列表</p> <p>去除本人已发表文献复制比: 0%(0) 文字复制比: 0%(0) 疑似剽窃观点: (0)</p>		
<p>说明: 1.总文字复制比: 被检测论文总重合字数在总字数中所占的比例</p> <p>2.去除引用文献复制比: 去除系统识别为引用的文献后, 计算出来的重合字数在总字数中所占的比例</p> <p>3.去除本人已发表文献复制比: 去除作者本人已发表文献后, 计算出来的重合字数在总字数中所占的比例</p> <p>4.单篇最大文字复制比: 被检测文献与所有相似文献比对后, 重合字数占总字数的比例最大的那一篇文献的文字复制比</p> <p>5.指标是由系统根据《学术论文不端行为的界定标准》自动生成的</p> <p>6.红色文字表示文字复制部分;绿色文字表示引用部分;棕色文字表示作者本人已发表文献部分</p> <p>7.本报告单仅对您所选择比对资源范围内检测结果负责</p>		
<p> <a href="mailto:amlc@cnki.net">amlc@cnki.net</a></p> <p><a href="http://check.cnki.net/">http://check.cnki.net/</a></p> <p><a href="http://e.weibo.com/u/3194559873/">http://e.weibo.com/u/3194559873/</a></p>		
<p><a href="http://check.cnki.net">http://check.cnki.net</a></p>		

## Appendix: My MATLAB Source Code for Experimental Measurements of the Sun's Visual Shape

**% The purpose of the program is to measure image and extract the shape parameter from the image.  
% The program includes 1 main program file and 9 subprogram files as well as 1 graphical user interface**

**% The main program file**

```
function varargout = VisualShapeOfSun(varargin)
% VISUALSHAPEOFSUN M-file for VisualShapeOfSun.fig

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @VisualShapeOfSun_OpeningFcn, ...
                  'gui_OutputFcn',  @VisualShapeOfSun_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
```

```
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Outputs from this function are returned to the command line.
function varargout = VisualShapeOfSun_OutputFcn(hObject, eventdata, handles)
% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes during object creation, after setting all properties.
function listbox_FileSlected_CreateFcn(hObject, eventdata, handles)
if ispc
    set(hObject, 'BackgroundColor', 'white');
else
    set(hObject, 'BackgroundColor', get(0, 'defaultUicontrolBackgroundColor'));
end

% --- Executes during object creation, after setting all properties.
function edit_filename_CreateFcn(hObject, eventdata, handles)
if ispc
    set(hObject, 'BackgroundColor', 'white');
else
    set(hObject, 'BackgroundColor', get(0, 'defaultUicontrolBackgroundColor'));
end

% --- Executes during object creation, after setting all properties.
```

```
function edit_erode_dilate_CreateFcn(hObject, eventdata, handles)

if ispc
    set(hObject, 'BackgroundColor', 'white');
else
    set(hObject, 'BackgroundColor', get(0, 'defaultUicontrolBackgroundColor'));
end

% --- Executes during object creation, after setting all properties.
function edit_threshold_CreateFcn(hObject, eventdata, handles)

if ispc
    set(hObject, 'BackgroundColor', 'white');
else
    set(hObject, 'BackgroundColor', get(0, 'defaultUicontrolBackgroundColor'));
end

% --- Executes during object creation, after setting all properties.
function slider_Threshold_CreateFcn(hObject, eventdata, handles)

usewhitebg = 1;
if usewhitebg
    set(hObject, 'BackgroundColor', [.9 .9 .9]);
else
    set(hObject, 'BackgroundColor', get(0, 'defaultUicontrolBackgroundColor'));
end
```

```
% --- Executes during object creation, after setting all properties.
function edit_EdgeDispThickness_CreateFcn(hObject, eventdata, handles)

if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% The program developed by myself starts here
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% --- Executes just before VisualShapeOfSun is made visible.
function VisualShapeOfSun_OpeningFcn(hObject, eventdata, handles, varargin)
    % Choose default command line output for VisualShapeOfSun
    handles.output = hObject;

    %run stutas flag
    handles.fileOpenFlag = 0;
    handles.greyImageFlag = 0;
    handles.binary_image_flag = 0;
    handles.edge_image_flag = 0;
```

```
handles.mom_computing_flag=0;
handles.lsm_computing_flag=0;

handles.greyTransformMethod = 1;
handles.medianFilterFlag = 1;
handles.thresholdValue = 0.5;
handles.NumberOfErodeAndDilate = 2;
handles.numEdgeDispThickness = 3;

% Update handles structure
guidata(hObject, handles);

% load filenames
cd('e:\Image Mesurement\imageOfSun');
load_current_directory_to_listbox(hObject, handles);

% initial edit box, slider
set(handles.edit_threshold, 'string', handles.thresholdValue);
set(handles.slider_Threshold, 'Value', handles.thresholdValue);
set(handles.edit_erode_dilate, 'string', handles.NumberOfErodeAndDilate);
set(handles.edit_EdgeDispThickness, 'string', handles.numEdgeDispThickness);

%clear disp result
set(handles.text_image_size, 'string', '0 X 0');
set(handles.text_computing_method, 'string', ' ');
```



```
set(handles.text_semimajor_axis, 'string', 0);
set(handles.text_semiminor_axis, 'string', 0);
set(handles.text_center_x, 'string', 0);
set(handles.text_center_y, 'string', 0);
set(handles.text_orientation, 'string', 0);
set(handles.text_shape_ratio, 'string', 0);

set(handles.text_mom_shapeRatio, 'string', 0);
set(handles.text_area_error, 'string', 0);
set(handles.text_lsm_shapeRatio, 'string', 0);
set(handles.text_statistic_error, 'string', 0);

% --- Executes on selection change in listBox_FileSlected.
function listBox_FileSlected_Callback(hObject, eventdata, handles)

index_selected = get(handles.listBox_FileSlected, 'Value');
file_list = get(handles.listBox_FileSlected, 'String');
filename = file_list{index_selected};

if handles.is_dir(handles.sorted_index(index_selected))
    cd (filename)
    load_current_directory_to_listbox(hObject, handles);
else
    [path, name, ext, ver] = fileparts(filename);
    if strcmp(ext, '.JPG')
        set(handles.edit_filename, 'String', name);
    end
end
```



```
        fileName = strcat(name, '.JPG');
        fileName = strcat('\', fileName);
        handles.fileName = strcat(pwd, fileName);
        guidata(hObject, handles);
    else
        set(handles.edit_filename, 'String', '');
        errordlg('File Type Error', 'modal');
    end
end

% --- Executes on text change in edit box
function edit_filename_Callback(hObject, eventdata, handles)

    name = get(hObject, 'String');
    fileName = strcat(name, '.JPG');
    fileName = strcat('\', fileName);
    handles.fileName = strcat(pwd, fileName);
    guidata(hObject, handles);

% --- Executes on button press in pushbutton_open.
function pushbutton_open_Callback(hObject, eventdata, handles)

    if handles.fileOpenFlag == 1
        errordlg('File has been opened !', 'modal');
        return;
    end
```



```
photo_sun = handles.fileName;
flag = exist(photo_sun, 'file');
if flag == 2
    handles.fileOpenFlag = 1;
    photo_info=imfinfo(photo_sun);
    photo_image=imread(photo_sun);
    handles.photo_image = photo_image;
    guidata(hObject, handles);
    axes(handles.axes_original_image);
    imshow(handles.photo_image);

    strWidth = int2str(photo_info.Width);
    strHight = int2str(photo_info.Height);
    strImageSize = strcat(strWidth, 'X', strHight);
    set(handles.text_image_size, 'string', strImageSize);
else
    errordlg('File do not exist !', 'modal');
end

% --- Executes on button press in pushbutton_close_all.
function pushbutton_close_all_Callback(hObject, eventdata, handles)

if handles.fileOpenFlag == 0
    errordlg('File has not been opened !', 'modal');
    return;
```

```
end

handles.fileOpenFlag = 0;
handles.greyImageFlag = 0;
handles.binary_image_flag = 0;
handles.edge_image_flag = 0;
handles.mom_computing_flag=0;
handles.lsm_computing_flag=0;
guidata(hObject, handles);

% clear all the axial windows
bk = zeros(4000,5000);
bkImage = mat2gray(bk,[0,1]);

axes(handles.axes_original_image);
imshow(bkImage);

axes(handles.axes_grey_image);
imshow(bkImage);

set(handles.text_axial3_Tag,'string','Binary Image');
axes(handles.axes_binary_image);
imshow(bkImage);

set(handles.text_axial4_Tag,'string','Edge Image');
axes(handles.axes_edge_image);
```



```
imshow(bkImage);

% clear disp results
set(handles.text_image_size, 'string', '0 X 0');
set(handles.text_computing_method, 'string', ' ');

set(handles.text_semimajor_axis, 'string', 0);
set(handles.text_semiminor_axis, 'string', 0);
set(handles.text_center_x, 'string', 0);
set(handles.text_center_y, 'string', 0);
set(handles.text_orientation, 'string', 0);
set(handles.text_shape_ratio, 'string', 0);

set(handles.text_mom_shapeRatio, 'string', 0);
set(handles.text_area_error, 'string', 0);
set(handles.text_lsm_shapeRatio, 'string', 0);
set(handles.text_statistic_error, 'string', 0);

% Read the current directory and sort the names
% -----
function load_current_directory_to_listbox(hObject,handles)

dir_path = pwd;
dir_struct = dir(dir_path);
[sorted_names,sorted_index] = sortrows({dir_struct.name}');
```



```
handles.is_dir = [dir_struct.isdir];
handles.sorted_index = [sorted_index];
handles.file_names = sorted_names;
handles.fileName = ' ';
guidata(hObject,handles)

set(handles.listbox_FileSlected,'String',sorted_names,'Value',1);
set(handles.edit_filename,'String','');

% -----Executes on radioButton in the election of Grey_transforming_Method ---
function Grey_transforming_Method_SelectionChangeFcn(hObject, eventdata, handles)

selection = get(hObject,'Tag');
switch selection
    case 'radiobutton_Idensity'
        handles.greyTransformMethod = 1;
    case 'radiobutton_R'
        handles.greyTransformMethod = 2;
    case 'radiobutton_G'
        handles.greyTransformMethod = 3;
    case 'radiobutton_B'
        handles.greyTransformMethod = 4;
end
guidata(hObject,handles)

% --- Executes on button press in checkbox_MedianFilter.
```

```
function checkbox_MedianFilter_Callback(hObject, eventdata, handles)

    selection = get(hObject, 'Value');
    handles.medianFilterFlag = 1;
    guidata(hObject, handles)

% --- Executes on button press in pushbutton_grey_image.
function pushbutton_grey_image_Callback(hObject, eventdata, handles)

    if handles.fileOpenFlag == 0
        errorDlg('File not opened !', 'modal');
        return;
    end
    % clear axial pane
    axes(handles.axes_grey_image);
    bk = zeros(4000, 5000);
    bkImage = mat2gray(bk, [0, 1]);
    imshow(bkImage);

    greyTransformMethod = handles.greyTransformMethod;
    isFilter = handles.medianFilterFlag;
    m_RGBImage = handles.photo_image;

    [ m_greyImage ] = RGB_TransformingToGrey( m_RGBImage, greyTransformMethod, isFilter);

    handles.grey_Image = m_greyImage;
```

```
handles.greyImageFlag = 1;
guidata(hObject,handles);

axes(handles.axes_grey_image);
imshow(handles.grey_Image);

% --- Executes on slider movement.
function slider_Threshold_Callback(hObject, eventdata, handles)

    newThresholdValue = get(hObject, 'Value');
    handles.thresholdValue = newThresholdValue;
    guidata(hObject,handles);
    set(handles.edit_threshold, 'string', handles.thresholdValue);

% --- Executes on text change in edit box
function edit_erode_dilate_Callback(hObject, eventdata, handles)

    strNumberErodeDilate = get(hObject, 'String');
    newNumberErodeDilate = str2double(strNumberErodeDilate);
    num = round(newNumberErodeDilate);

    if num < 0 || num > 20 || isnan(num)
        errordlg('Eroding and Dilating number is out of range !', 'modal');
        set(handles.edit_erode_dilate, 'string', handles.NumberOfErodeAndDilate);
    else
```

```
handles.NumberOfErodeAndDilate = num;
guidata(hObject,handles);
set(handles.edit_erode_dilate,'string',handles.NumberOfErodeAndDilate);
end

% --- Executes on text change in edit box
function edit_threshold_Callback(hObject, eventdata, handles)

strThreshold = get(hObject,'String');
newThresholdValue = str2double(strThreshold);
if newThresholdValue < 0 || newThresholdValue > 1 || isnan(newThresholdValue)
    errordlg('Threshold value is out of range !','modal');
    set(handles.edit_threshold,'string',handles.thresholdValue);
else
    handles.thresholdValue = newThresholdValue;
    guidata(hObject,handles);
    set(handles.slider_Threshold,'Value',handles.thresholdValue);
end

% --- Executes on button press in pushbutton_binary_image.
function pushbutton_binary_image_Callback(hObject, eventdata, handles)

if handles.greyImageFlag == 0
    errordlg('Gray image do not exist !','modal');
    return;
end
```

```
NumberErodeDilate = handles.NumberOfErodeAndDilate;  
m_grayImage = handles.grey_Image;  
thresholdValue = handles.thresholdValue;
```

```
m_binaryImage = GrayTransformingToBinaryImage( m_grayImage,thresholdValue,NumberErodeDilate);
```

```
handles.binary_image = m_binaryImage;  
handles.binary_image_flag = 1;  
guidata(hObject,handles);
```

```
axes(handles.axes_binary_image);  
imshow(handles.binary_image);  
set(handles.text_axial3_Tag, 'string', 'Binary Image');
```

```
% --- Executes on button press in pushbutton_threshold_finding.
```

```
function pushbutton_threshold_finding_Callback(hObject, eventdata, handles)
```

```
if handles.greyImageFlag == 0  
    errordlg('Gray image do not exist !', 'modal');  
    return;  
end
```

```
NumberErodeDilate = handles.NumberOfErodeAndDilate;  
m_greyImage = handles.grey_Image;
```

```
[ xdata,ydata,newThresholdValue ] = findThresholdOfBinaryImage(m_greyImage);
```

```
handles.thresholdValue = newThresholdValue;  
guidata(hObject,handles);
```

```
set(handles.edit_threshold,'string',handles.thresholdValue);  
set(handles.slider_Threshold,'Value',handles.thresholdValue);
```

```
axes(handles.axes_binary_image);  
plot(xdata,ydata);  
set(gca,'XTick',[]);  
set(gca,'YTick',[]);  
set(handles.text_axial3_Tag,'string','Area to THreshold');
```

```
% --- Executes on text change in edit box
```

```
function edit_EdgeDispThickness_Callback(hObject, eventdata, handles)
```

```
strEdgeDispThickness = get(hObject,'String');  
numEdgeDispThickness = str2double(get(hObject,'String'));
```

```
if numEdgeDispThickness < 0 || numEdgeDispThickness > 15 || isnan(numEdgeDispThickness)  
    errordlg('Thickness is out of range !','modal');  
    set(handles.edit_EdgeDispThickness,'string', handles.numEdgeDispThickness);  
    return;
```



```
else
    handles.numEdgeDispThickness = round(numEdgeDispThickness);
    guidata(hObject,handles);
end

if handles.edge_image_flag == 1
    m_edgeImage = handles.edge_Image;
    m_Thickness = handles.numEdgeDispThickness ;
    for ii = 1:m_Thickness
        m_edgeImage = bwmorph(m_edgeImage, 'dilate');
    end
    axes(handles.axes_edge_image);
    imshow(m_edgeImage);
end

% --- Executes on button press in pushbutton_find_edge.
function pushbutton_find_edge_Callback(hObject, eventdata, handles)

if handles.binary_image_flag == 0
    errordlg('Binary image do not exist !','modal');
    return;
end

m_binaryImage = handles.binary_image;
m_edgeImage = bwmorph(m_binaryImage, 'remove');
```



```
handles.edge_Image = m_edgeImage;
handles.edge_image_flag = 1;
guidata(hObject,handles);

m_Thickness = handles.numEdgeDispThickness;
for ii = 1:m_Thickness
    m_edgeImage = bwmorph(m_edgeImage,'dilate');
end

set(handles.text_axial4_Tag,'string','Edge Image');
axes(handles.axes_edge_image);
imshow(m_edgeImage);

% --- Executes on button press in pushbutton_MOM.
function pushbutton_MOM_Callback(hObject, eventdata, handles)

if handles.binary_image_flag == 0
    errordlg('Binary image do not exist !','modal');
    return;
end

m_binaryImage = handles.binary_image;

[ M_Xc,M_Yc,M_alfa,M_ratio,M_a_ellipse,M_b_ellipse,M_area ] = momentMethod( m_binaryImage );
```



```
[ A_error ] = areaError( M_Xc,M_Yc,M_alfa,M_a_ellipse,M_b_ellipse,m_binaryImage );
```

```
% store
```

```
handles.mom_computing_flag = 1;
```

```
handles.mom_center_x = M_Xc;
```

```
handles.mom_center_y = M_Yc;
```

```
handles.mom_orientation = M_alfa;
```

```
handles.mom_shape_ratio = M_ratio;
```

```
handles.mom_semimajor_axis = M_a_ellipse;
```

```
handles.mom_semiminor_axis = M_b_ellipse;
```

```
handles.mom_area_error = A_error/M_area;
```

```
guidata(hObject,handles);
```

```
% display
```

```
set(handles.text_computing_method,'string','Moment method');
```

```
set(handles.text_semimajor_axis,'string',round(handles.mom_semimajor_axis*10)/10);
```

```
set(handles.text_semiminor_axis,'string',round(handles.mom_semiminor_axis*10)/10);
```

```
set(handles.text_center_x,'string',round(handles.mom_center_x*10)/10);
```

```
set(handles.text_center_y,'string',round(handles.mom_center_y*10)/10);
```

```
set(handles.text_orientation,'string',round(handles.mom_orientation*100)/100);
```

```
set(handles.text_shape_ratio,'string',round(handles.mom_shape_ratio*1000)/1000);
```

```
set(handles.text_mom_shapeRatio,'string',round(handles.mom_shape_ratio*1000)/1000);
```

```
set(handles.text_area_error,'string',round(handles.mom_area_error*1000)/10);
```

```
% --- Executes on button press in pushbutton_LSM.
function pushbutton_LSM_Callback(hObject, eventdata, handles)

    if handles.edge_image_flag == 0
        errordlg('Edge image do not exist !','modal');
        return;
    end

    if handles.mom_computing_flag == 0
        errordlg('Apply MOM first !','modal');
        return;
    end

    %LSM computation
    m_edgeImage = handles.edge_Image;
    [X_coordi,Y_coordi,v] = find(m_edgeImage);

    M_Xc = handles.mom_center_x;
    M_Yc = handles.mom_center_y;
    M_alfa = handles.mom_orientation;
    M_a_ellipse = handles.mom_semimajor_axis;
    M_b_ellipse = handles.mom_semiminor_axis;

    [ LSM_xc,LSM_yc,LSM_alfa,LSM_ratio,LSM_a,LSM_b,LSM_error ] = ...

    LqsMethod( X_coordi,Y_coordi,M_a_ellipse,M_b_ellipse,M_Xc,M_Yc,M_alfa);
```

```
%store main results to handles
handles.lsm_computing_flag=1;
handles.lsm_shape_ratio = LSM_ratio;
handles.lsm_statistic_error = LSM_error(1)*LSM_error(2)*3/LSM_a...
    + LSM_error(1)*LSM_error(3)*3/LSM_b;
guidata(hObject,handles);
%disp results
set(handles.text_computing_method,'string','Least square method');

set(handles.text_semimajor_axis,'string',round(LSM_a*10)/10);
set(handles.text_semiminor_axis,'string',round(LSM_b*10)/10);
set(handles.text_center_x,'string',round(LSM_xc*10)/10);
set(handles.text_center_y,'string',round(LSM_yc*10)/10);
set(handles.text_orientation,'string',round(LSM_alfa*100)/100);
set(handles.text_shape_ratio,'string',round(handles.lsm_shape_ratio*1000)/1000);

set(handles.text_lsm_shapeRatio,'string',round(handles.lsm_shape_ratio*1000)/1000);
set(handles.text_statistic_error,'string',round(handles.lsm_statistic_error*1000000)/10000);

%draw fitting ellipse

axes(handles.axes_edge_image);

drawFittingEllipse( X_coordi,Y_coordi,LSM_a,LSM_b,LSM_alfa,LSM_xc,LSM_yc);
```

```
set(handles.text_axial4_Tag, 'string', 'Fitting ellipse');
```

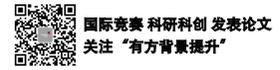
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%                                                                    %  
%      The program developed by myself ends here                    %  
%                                                                    %  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```



## % The first subprogram file

```
function [ m_binaryImage ] = GrayTransformingToBinaryImage( m_grayImage,thresholdValue,NumberErodeDilate)
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%                                                                    %  
% The program developed by myself                                     %  
%                                                                    %  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
[M,N,C]=size(m_grayImage);  
m_grayImage(3400:M,4000:N)=0;  
  
A_bw1=im2bw(m_grayImage,thresholdValue);  
nn=NumberErodeDilate;  
NHOOD=[0,1,0;1,1,1;0,1,0];  
  
for k=1:nn  
    A_bw1=bwmorph(A_bw1,'erode');  
end  
  
for k=1:nn*2  
    A_bw1=bwmorph(A_bw1,'dilate');  
end  
  
for k=1:nn  
    A_bw1=bwmorph(A_bw1,'erode');
```



```
end  
m_binaryImage = A_bw1;  
end
```

## % The second subprogram file

```
function [ xdata,ydata,threshod ] = findThresholdOfBinaryImage(m_greyImage)
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%                                                    %  
% The program developed by myself                    %  
%                                                    %
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
maxThreshold = 0.95;  
minThreshold = 0.05;  
mn=10;  
delta = (maxThreshold - minThreshold)/(mn-1);  
for ii=1:mn  
    m_threshod = minThreshold + (ii-1)*delta;  
    x(ii) = m_threshod;
```

```
y(ii) = areaOfBinaryImage( m_greyImage,m_threshod);
```

```
end  
for ii=1:mn-2  
    ddy(ii)=- (y(ii+2)-2*y(ii+1)+y(ii));  
end
```

```
meanValue = sum(y)/mn*0.5;  
temp = abs(ddy(1));
```



```
index = 1;
for ii =2:mn-2
    if temp > abs(ddy(ii)) && meanValue < y(ii)
        temp = abs(ddy(ii));
        index = ii;
    end
end
xdata = x;
ydata = y;
threshod = x(index+1);
threshod = round(threshod*1000)/1000;
end
```

**% The third subprogram**

```
function [ area ] = areaOfBinaryImage( m_greyImage,m_threshod)
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%                                                                    %
```

```
% The program developed by myself                                     %
```

```
%                                                                    %
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
    A_bw=im2bw(m_greyImage,m_threshod);
```

```
    x_v=sum(A_bw,2,'double');
```

```
    area=sum(x_v);
```

```
end
```



### % The fourth subprogram file

```
function [ M_Xc,M_Yc,M_alfa,M_ratio,M_a_ellipse,M_b_ellipse,M_area ] = momentMethod( A_bw )
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%                                                                 %
```

```
%   The program developed by myself                               %
```

```
%                                                                 %
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%% initialize
```

```
PIE=3.1415926/180;
```

```
[M,N,C]=size(A_bw);
```

```
M_vector=1:1:M;
```

```
N_vector=1:1:N;
```

```
%%%%%%%%% area
```

```
x_v=sum(A_bw,2,'double');
```

```
y_v=sum(A_bw,1,'double');
```

```
area=sum(x_v);
```

```
% area=sum(y_v);
```

```
%%%%%%%%% center
```

```
xx_v=times(x_v,M_vector');
```

```
yy_v=times(y_v,N_vector);
```

```
Xc0=sum(xx_v)/area;
```



```
Yc0=sum(yy_v)/area;

%%%%%%%%% moment
x_y=kron(M_vector',N_vector);
xMy_v=times(x_y,A_bw);
I_xy=sum(sum(xMy_v))/area-Xc0*Yc0;
%%%
N_one=ones(1,N);
x_2=times(M_vector,M_vector);
xx=kron(x_2',N_one);
xMx_v=times(xx,A_bw);
I_xx=sum(sum(xMx_v))/area-Xc0*Xc0;
%%%
M_one=ones(1,M);
y_2=times(N_vector,N_vector);
yy=kron(M_one',y_2);
yMy_v=times(yy,A_bw);
I_yy=sum(sum(yMy_v))/area-Yc0*Yc0;
%%%
if abs(I_xx-I_yy)>1.0e-4
    alfa=atan(-I_xy/(I_xx-I_yy))/PIE;
elseif abs(I_xy)>1.0e-4
    alfa=45;
else
    alfa=0;
end
```



```
I_xx0=I_xx*cos(alfa*PIE)*cos(alfa*PIE)+I_yy*sin(alfa*PIE)*sin(alfa*PIE)-2*I_xy*sin(alfa*PIE)*cos(alfa*PIE);  
I_yy0=I_yy*cos(alfa*PIE)*cos(alfa*PIE)+I_xx*sin(alfa*PIE)*sin(alfa*PIE)+2*I_xy*sin(alfa*PIE)*cos(alfa*PIE);  
  
%%%%%%%%%%ratio by moment  
M_ratio=sqrt(I_xx0/I_yy0);  
M_a_elipse_2=sqrt(sqrt(4*M_ratio*I_xx0*area/3.14159));  
% M_a_elipse_0=sqrt(area*M_ratio/3.14159);  
M_a_ellipse=M_a_elipse_2;  
M_b_ellipse=M_a_ellipse/M_ratio;  
M_alfa=-alfa;  
M_Xc=Xc0;  
M_Yc=Yc0;  
M_area=area;  
end
```

**% The fifth subprogram file**

```
function [ A_error ] = areaError( M_Xc,M_Yc,M_alfa,M_a_ellipse,M_b_ellipse,A_bw )
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% %
```

```
% The program developed by myself %
```

```
% %
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
[M,N]=size(A_bw);
```

```
[ PictureEllipse ] = GenerateEllipse(M_Xc,M_Yc,M_alfa,M_a_ellipse,M_b_ellipse,M,N);
```

```
pictureMinus = A_bw - PictureEllipse;
```

```
A_error=sum(sum(abs(pictureMinus)));
```

```
end
```

**% The sixth subprogram file**

```
function [ PictureEllipse ] = GenerateEllipse( M_Xc,M_Yc,M_alfa,M_a_ellipse,M_b_ellipse,M,N )
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%  
% The program developed by myself  
%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
PIE = 3.1415926/180;  
x = 1:1:M;  
y = 1:1:N;  
x = x-M_Xc;  
y = y-M_Yc;  
Mone = ones(M,1);  
oneN = ones(1,N);  
xx = kron(x',oneN);  
yy = kron(Mone,y);  
xx1 = xx.*cos(M_alfa*PIE) + yy.*sin(M_alfa*PIE);  
yy1 = -xx.*sin(M_alfa*PIE) + yy.*cos(M_alfa*PIE);  
b=M_b_ellipse;  
a=M_a_ellipse;  
z = 1-xx1.*xx1./a/a-yy1.*yy1/b/b;  
PictureEllipse = z>=0;
```

```
end
```



## % The seventh subprogram file

```
function [ P_xc,P_yc,P_alfa,P_ratio,P_a,P_b,P_error ] = LqsMethod( X_coordi,Y_coordi,P_a0,P_b0,P_xc0,P_yc0,P_alfa0 )
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% The program developed by myself
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
PIE=3.1415926/180;
NN=size(X_coordi);
Nn=NN(1);
XX=zeros(1,Nn);
YY=zeros(1,Nn);
sita=zeros(1,Nn);
er=zeros(Nn,1);
A=zeros(Nn,5);
for i=1:Nn
    %%% transform data
    XX(i)=(X_coordi(i)-P_xc0)*cos(P_alfa0*PIE)+(Y_coordi(i)-P_yc0)*sin(P_alfa0*PIE);
    YY(i)=-(X_coordi(i)-P_xc0)*sin(P_alfa0*PIE)+(Y_coordi(i)-P_yc0)*cos(P_alfa0*PIE);
    xi=XX(i);
    yi=YY(i);
    r=sqrt(xi*xi*P_b0*P_b0+yi*yi*P_a0*P_a0);
    sita(i)=acos(xi*P_b0/r)/PIE;
    if yi<0
```



```
        sita(i)=360-sita(i);
end
%%% ellipse coordi and tangent
xe=P_a0*cos(sita(i)*PIE);
ye=P_b0*sin(sita(i)*PIE);
xp=-P_a0*sin(sita(i)*PIE);
yp=P_b0*cos(sita(i)*PIE);
D=sqrt(xp*xp+yp*yp);
%%%%%%%%%%%%% sita modifying by tangent projection
er_x=xi-xe;
er_y=yi-ye;
er_T=(er_x*xp+er_y*yp)/D;
delta_sita=er_T/D/PIE;
sita(i)=sita(i)+delta_sita;
%%%%%%%%%%%%%
end
%%%%%%%%%%%%%   lqs matrix for x0 y0 gama
for i=1:Nn
    %%% ellipse coordi and tangent vector
    xe=P_a0*cos(sita(i)*PIE);
    ye=P_b0*sin(sita(i)*PIE);
    xp=-P_a0*sin(sita(i)*PIE);
    yp=P_b0*cos(sita(i)*PIE);
    D=sqrt(xp*xp+yp*yp);
    Nx=-yp/D;
    Ny=xp/D;
```



```
%%%%%%%%% error vector and Linear regression equation
er_x=XX(i)-xe;
er_y=YY(i)-ye;
er(i)=er_x*Nx+er_y*Ny;
A(i,1)=xe*Ny-ye*Nx;
A(i,2)=Nx;
A(i,3)=Ny;
A(i,4)=Nx*cos(sita(i)*PIE);
A(i,5)=Ny*sin(sita(i)*PIE);
end
%%%%%%%%% lqs by solving matrix equation
AA=A'*A;
B=A'*er;
delta_x=AA\B;
errorV=er-A*delta_x;
epsilon_error=sum(errorV.*errorV)/(Nn-5);
XX_1=inv(AA);
P_error(1)=sqrt(epsilon_error);
P_error(2)=XX_1(4,4);
P_error(3)=XX_1(5,5);
%%%%%%%%% coordinate transformation and output result
P_alfa=P_alfa0+delta_x(1)/PIE;
xc0=delta_x(2);
yc0=delta_x(3);
P_a=P_a0+delta_x(4);
P_b=P_b0+delta_x(5);
```



```
P_xc=xc0*cos(P_alfa0*PIE)-yc0*sin(P_alfa0*PIE)+P_xc0;  
P_yc=xc0*sin(P_alfa0*PIE)+yc0*cos(P_alfa0*PIE)+P_yc0;  
P_ratio=P_a/P_b;  
end
```

## % The eighth subprogram file

```
function drawFittingEllipse( X_coordi,Y_coordi,ah,bh,alfa,xc,yc)
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%                                                                    %  
%   The program developed by myself                                %  
%                                                                    %  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
plot(Y_coordi,-X_coordi,'-wo',...  
      'LineWidth',1,...  
      'MarkerEdgeColor','b',...  
      'MarkerFaceColor',[0.49 1 0.63],...  
      'MarkerSize',2);  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% construct ellipse %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
n=4;  
xx=zeros(360*n,1);  
yy=zeros(360*n,1);  
PIE=3.1415926/180;  
for i=1:360*n  
    sita=(i-1)/n*PIE;  
    xe=ah*cos(sita);  
    ye=bh*sin(sita);  
    xx(i)=xe*cos(alfa*PIE)-ye*sin(alfa*PIE)+xc;  
    yy(i)=xe*sin(alfa*PIE)+ye*cos(alfa*PIE)+yc;  
end
```

```
hold on
plot(yy,-xx,'-rd',...
      'LineWidth',1,...
      'MarkerSize',1);
hold off
set(gca,'XLim',[0 5182]);
set(gca,'YLim',[-3888 0]);
set(gca,'XTick',[]);
set(gca,'YTick',[]);
end
```



## % The ninth subprogram file

```
function [ m_greyImage ] = RGB_TransformingToGrey( m_RGBImage, greyTransformMethod, isFilter)
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%                                                                 %
```

```
% The program developed by myself                               %
```

```
%                                                                 %
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

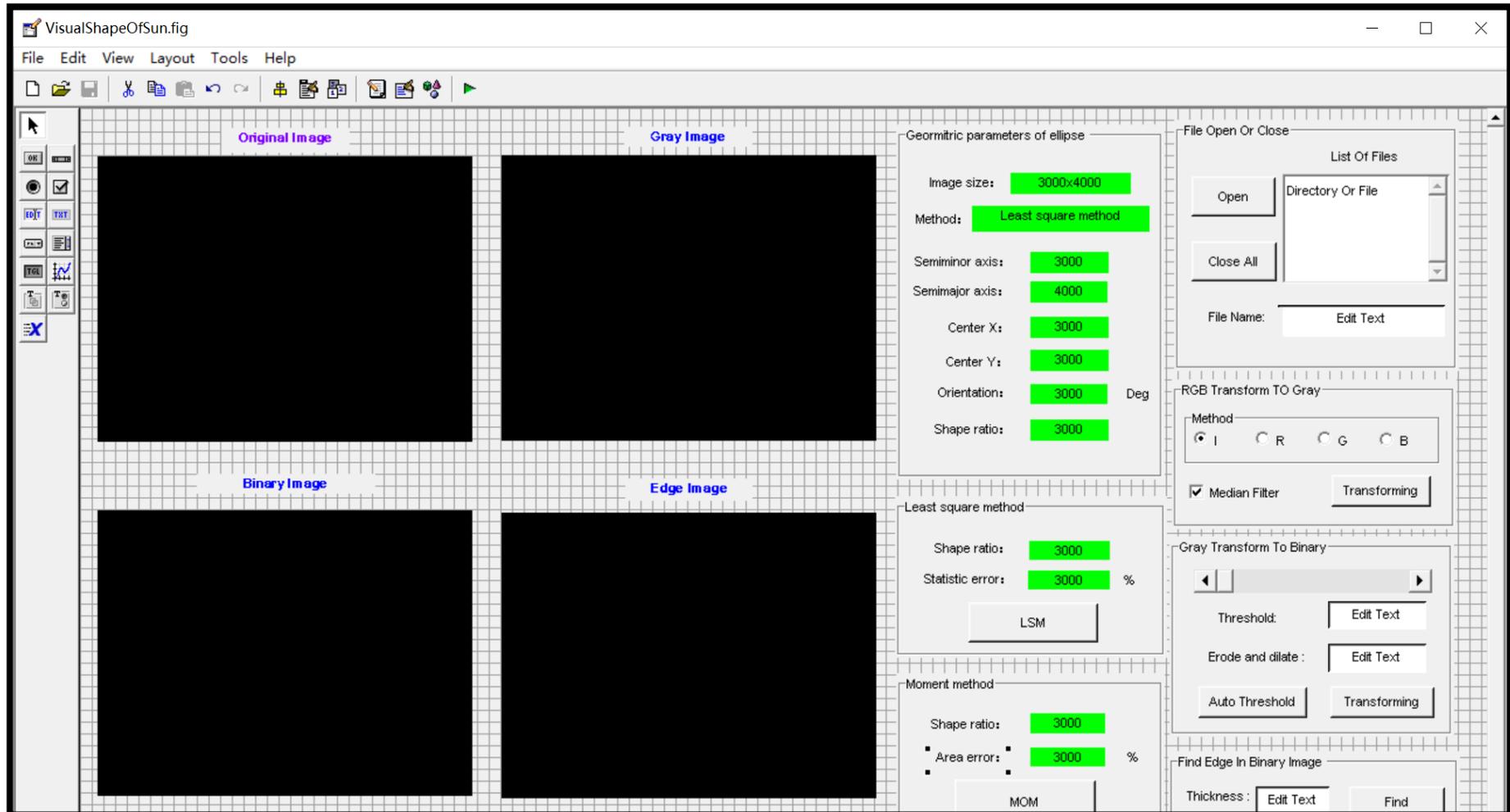
```
switch greyTransformMethod
    case 1
        A_gray0=rgb2gray(m_RGBImage);
    case 2
        imm = m_RGBImage(:, :, 1);
        dmm = double(imm);
        A_gray0 = mat2gray(dmm);
    case 3
        imm = m_RGBImage(:, :, 2);
        dmm = double(imm);
        A_gray0 = mat2gray(dmm);
    case 4
        imm = m_RGBImage(:, :, 3);
        dmm = double(imm);
        A_gray0 = mat2gray(dmm);
end
if isFilter == 1
```



国际竞赛 科研科创 发表论文  
关注“有方背景提升”

```
    for i = 1:3
        A_gray0 = medfilt2(A_gray0);
    end
end
m_greyImage = A_gray0;
end
```

% The graphical user interface developed by myself



## ☞ Running my program

The software interface, titled "Measurement and Analysis for Visual Shape of the Sun", displays the following components:

- Original Image:** A color image of the sun.
- Gray Image:** The original image converted to grayscale.
- Binary Image:** The grayscale image converted to a binary (black and white) image.
- Fitting ellipse:** The binary image with a fitted ellipse overlaid.

**Geometric parameters of ellipse:**

- Image size: 5184X3888
- Method: Least square method
- Seminor axis: 1530.3
- Semimajor axis: 1688
- Center X: 1941.2
- Center Y: 2563.8
- Orientation: 3.48 Deg
- Shape ratio: 0.907

**Least square method:**

- Shape ratio: 0.907
- Statistic error: 0.0008 %
- LSM button

**Moment method:**

- Shape ratio: 0.907
- Area error: 0.7 %
- MOM button

**File Open Or Close:**

- List Of Files: DSC08409.JPG, DSC08435.JPG, DSC08477.JPG (selected), DSC08477.TXT
- File Name: DSC08477

**RGB Transform TO Gray:**

- Method: I (selected), R, G, B
- Median Filter: checked
- Transforming button

**Gray Transform To Binary:**

- Threshold: 0.45
- Erode and dilate: 2
- Auto Threshold button
- Transforming button

**Find Edge In Binary Image:**

- Thickness: 3
- Find button