



Article Tri-SIFT: A Triangulation-Based Detection and Matching Algorithm for Fish-Eye Images

Ende Wang ¹,*, Jinlei Jiao ^{1,2}, Jingchao Yang ³, Dongyi Liang ² and Jiandong Tian ¹

- Key Laboratory of Optical Electrical Image Processing, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China; jinleijiao666@gmail.com (J.J.); tianjd@sia.cn (J.T.)
- ² College of Information Science and Engineering, Northeastern University, Shenyang 110819, China; liangdongyi2121@gmail.com
- ³ Department of electrical and Information Engineering, Hebei Jiaotong Vocational and Technical College, Shijiazhuang 050035, China; yangjingchao2121@gmail.com
- * Correspondence: 1770559@stu.neu.edu.cn

Received: 6 November 2018; Accepted: 21 November 2018; Published: 26 November 2018



Abstract: Keypoint matching is of fundamental importance in computer vision applications. Fish-eye lenses are convenient in such applications that involve a very wide angle of view. However, their use has been limited by the lack of an effective matching algorithm. The Scale Invariant Feature Transform (SIFT) algorithm is an important technique in computer vision to detect and describe local features in images. Thus, we present a Tri-SIFT algorithm, which has a set of modifications to the SIFT algorithm that improve the descriptor accuracy and matching performance for fish-eye images, while preserving its original robustness to scale and rotation. After the keypoint detection of the SIFT algorithm is completed, the points in and around the keypoints are back-projected to a unit sphere following a fish-eye camera model. To simplify the calculation in which the image is on the sphere, the form of descriptor is based on the modification of the Gradient Location and Orientation Histogram (GLOH). In addition, to improve the invariance to the scale and the rotation in fish-eye images, the gradient magnitudes are replaced by the area of the surface, and the orientation is calculated on the sphere. Extensive experiments demonstrate that the performance of our modified algorithms outweigh that of SIFT and other related algorithms for fish-eye images.

Keywords: SIFT; triangulation; detection; matching; fish-eye

1. Introduction

Visual feature extraction and matching are the most basic and difficult problems in computer vision and application of optical engineering. Many applications are built on visual feature matching, such as robotic navigation, image stitching, 3D modeling, gesture recognition, and video tracking. In most of these applications, unconventional lensed cameras with nonlinear projection exhibit numerous advantages compared to regular cameras. A camera equipped with micro-lenses and borescopes enables the visual inspection of cavities that are difficult to access [1], whereas a camera equipped with a fish-eye lens can acquire wide field-of-view (FOV) images for a thorough visual coverage of environments. Such a camera also improves the performance of geomotion estimation by avoiding the ambiguity between translation and rotation motions [2,3].

However, the visual feature matching algorithms designed for perspective images cannot handle the strong radial distortion introduced by the optics [4–7]. The entire hemispherical field is covered in front of the fish-eye camera, and the view angle of fish-eye lenses is in the range of 0° –180°. In addition, a fish-eye lenses obey other projection models because the hemispherical field of view cannot be projected on a finite image plane through a perspective projection. Thus, the fish-eye model is different from the common camera model, and the inherent distortion of a fish-eye lens is similar to that of the pinhole model [8]. Because of the distinctive feature of the camera lens and the valuable wide angle of the images, fisheye images suffer from large radial distortion and change in scale according to the image locations.

In this paper, we propose the Tri-SIFT feature matching method to overcome radial distortion of fish-eye cameras. We demonstrate how radial distortion affects the performance of the original Scale Invariant Feature Transform (SIFT) algorithm and propose a set of modifications that improve the matching effectiveness. The paper provides a detailed account of the method, presenting details of a thorough analysis and experimental validation.

In detail, we propose a triangulation-based detection and matching algorithm combined with the camera's imaging model to eliminate the impact of distortion. Our method improves the robustness of the proposed method to distortion and enhances the efficiency of feature point matching in large distortion areas.

In Section 2, we present the related works. In Section 3, we briefly introduce the SIFT algorithm. In Section 4, we describe the proposed tri-SIFT. In Section 5, we present and discuss the experimental results. Finally, we summarize the features of the proposed algorithm in Section 6.

2. Related Work

SIFT is an algorithm in computer vision used to extract and describe local features in images [9]. It is able to extract stable features from resized and rotated images. The SIFT algorithm exhibits stable performance in terms of the images' scale, size, and noise in the Gaussian scale space. Moreover, the SIFT algorithm can adapt to perspective and lighting transformations. SIFT's superior performance has rapidly made it the most commonly used feature extraction algorithm.

Recently, several algorithms concerning keypoint detection and matching in fish-eye images have been proposed [5–7,10,11]. In a series of studies [6,7], Hansen, Corke, and Boles proposed a method that involves using stereographic projections for approximating the diffusion on a sphere. In their methods, SIFT was modified for images with significant distortion. In [10], Lourenço et al. proposed the adaptive Gaussian filtering to correct the SIFT algorithm. This method detects keypoints by looking for extrema in a scale-space representation obtained using a kernel that adapts the distortion at each image pixel position. It also achieved description invariance to RD (radial distortion) by implementing implicit gradient correction using the Jacobian of the distortion function. In [11], Denny et al. described a method to photogrammetrically estimate the intrinsic and extrinsic parameters of fish-eye cameras, with the aim of providing a rectified image for scene viewing applications. While some works simply ignored the pernicious effects of the radial distortion and directly applied the original algorithm to distorted images [12], others performed a preliminary correction of distortion through image rectification and then applied SIFT [13]. The latter approach is quite straightforward, but it has two major drawbacks: the explicit distortion correction can be computationally expensive for the case of large frames; more importantly, the interpolation required by the image rectification introduces artifacts that affect the detection repeatability.

3. SIFT Algorithm Theory

In this section, we briefly introduce the SIFT algorithm, which the Tri-SIFT algorithm is inspired by. The major steps of the SIFT algorithm are: detecting the threshold in scale space, locating features, selecting the dominant orientation for feature points, and establishing the features' descriptor. For an image I(x,y),

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2 + y^2)} / 2\sigma^2$$
(1)

using the Gaussian function (1) as the convolution kernel, the scale space of a two-dimensional image can be obtained using a Gaussian kernel convolution. σ is the width parameter of the function, which controls the radial extent of the function.

$$L(x, y, \sigma) = G(x, y, \sigma) \times I(x, y)$$
⁽²⁾

The SIFT algorithm determines the feature points by detecting local keypoints in a two-dimensional Difference of Gaussian (DoG) scale space to ensure the unique and stable feature points. The DoG operator is defined as the subtraction of the two different scales of the Gaussian kernel. *k* is the scale factor. It is the approximation of the normalized 16×16 Laplacian of Gaussian (LoG)

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma)$$
(3)

The feature point requires a total of 26 neighborhood pixels to ensure that the chosen pixel is the local keypoint in the scale space and the two-dimensional image space. SIFT detects the local keypoint by fitting a three-dimensional quadratic function to determine the location and scale of feature points (up to sub-pixel accuracy). In addition, the SIFT algorithm excludes low-contrast feature points and unstable edge response points to enhance the matching stability and suppress noise. By assigning a dominant orientation for every feature point, the feature point's descriptor is described in its dominant orientation to achieve rotation invariance. The value of the gradient m(x, y) and the orientation $\theta(x, y)$ of each image L(x, y) are obtained by the differences between pixel points.

Finally, a 16×16 neighborhood window of the feature point in the rotated image is obtained, and the window is evenly divided into sub-regions. A gradient orientation histogram of eight orientations in every sub-region is calculated, and the values of all gradient d orientations are accumulated. The feature point's descriptor is a $4 \times 4 \times 8 = 128$ dimensional vector. Then, SIFT re-normalizes the dimensional vector to eliminate the impact of the light transform.

4. Tri-SIFT Algorithm

We propose the Tri-SIFT algorithm, which is an extension of the SIFT algorithm, for application to fish-eye images. The first stage of Tri-SIFT is the search for keypoints over all scales and image locations. It is implemented efficiently by using a DoG function to identify potential interest points that are invariant to the scale and orientation. For each candidate location, a detailed model is fit to determine the location and scale, and keypoints are selected based on their stability. To match the points extracted from different fish-eye images, and those obtained using the proposed algorithm, a Local Spherical Descriptor (LSD) is computed at each point on the surface of a unit sphere. The descriptor is obtained using the spherical representation of the image and consists of a set of histograms of orientations in the region around the given point. The size of the region depends on the scale (σ) at which the point has been detected. The magnitude of the LSD is calculated by the area of the triangle obtained by triangulating the points in a circular area surrounding the keypoint. The orientation of the LSD is along the normal component of the plane determined by the three vertices of the triangle.

In this section, we first introduce back-projection and triangulation and the Delaunay triangulation algorithm, and then describe the method of calculation of the dominant orientation and the descriptor construction.

4.1. Back-Projection

The distortion caused by the nonlinear projection of a fish-eye camera lens causes nonuniform compression of the image structures, which affects the SIFT matching performance. The conventional method is to rectify the fish-eye image to the undistorted image by explicitly correcting the distortion and applying classical SIFT to the rectified image [14,15]. The solution is straightforward; however, the problem is that the distortion correction by image resampling requires reconstruction of the signal from the initial discrete image. Thus, there are high-frequency components that cannot be

recovered (e.g., low resolution and aliasing), and the reconstruction filters are imperfect. This outcome negatively affects the construction of the descriptor and decreases thus the accuracy of keypoint matching on images.

In this paper, we propose a model-based approach by transforming the fish-eye image to its original state, in which the lights of the physical world pass through the camera lens. As shown in Figure 1, the projection process of the spherical model for an omnidirectional camera can be divided into two steps [16–18]. We assume a point P(X, Y, Z) in space to demonstrate these two steps. In the first step, the point is linearly projected along the incident ray to a point \tilde{p} on the unit sphere, where θ is the angle between the incident ray oP and the principal axis z_c . r is the distance between the image point and the principal point. In the second step, the point \tilde{p} is then non-linearly projected to a point p on the image plane *XOY*. There are several mathematical models to describe the second projection step, such as the following polynomial formulation.

$$r(\theta) = k_1 \theta + k_2 \theta^3 \tag{4}$$

$$\begin{pmatrix} X \\ Y \end{pmatrix} = r(\theta) \begin{bmatrix} m_u & 0 \\ 0 & m_v \end{bmatrix} \begin{pmatrix} \cos \varphi \\ \sin \varphi \end{pmatrix} + \begin{pmatrix} x_c \\ y_c \end{pmatrix}.$$
(5)

where k_1 is the radial or polar distance from the image point to the origin of the world coordinate system; i.e., k_1 is the focal length. k_2 is the distortion coefficient. *X* and *Y* are the image coordinates, and x_c and y_c are the pixel coordinates of the principal point. φ is the angle between the *X*-axis and the radial line passing through the image point *p*. m_u and m_v are the two scale factors denoting the number of pixels per unit distance in the horizontal and vertical orientations, which should be known beforehand. The mapping between the point *P* in the space and the image point *p* is reversible, and the reversal can be performed by using (5)—the detailed process is reported in [16].



Figure 1. Projection of a point.

Using the camera model, the fish-eye images can be back-projected to the unit sphere, as if the sensor is on the surface of the fish-eye lens. The scene of the actual world is linearly projected to the lens, with the scale corresponding sequentially to that scale of the real world. As a result, the light is no longer non-linearly projected to the sensor plane, and the consequent distortion is eliminated from the fish-eye image that then becomes a back-projected image.

4.2. Triangulation

If the size of the selected region is fixed on the back-projected image to calculate the orientation of the keypoint or the descriptor, the number of pixels in the region varies when the location of the region changes. In theory, the number of the pixels decreases when the polar angle θ increases, which makes the feature points rotation-variant. In Figure 2, there are four panoramic test images, which contain degrees of distortion: 10%, 20%, 30%, and 40%, respectively. In Figure 3, the groundtruth of the keypoints are detected by SIFT in the images without degree of distortion. Then, the keypoints of the four test images with different degrees of distortion are compared with the groundtruth to estimate the keypoints detection effect. Repetition represents the percentage of keypoints, which are detected both

in groundtruth and the test images. New detection represents the keypoints detected in the test images which are not detected in the groundtruth. Wrong detection represents the wrong points which are detected as keypoints in the test images. In Figure 4, shows the matching results of keypoints in four test images with groundtruth. Recall and precision of the four test images are calculated point by point. As shown in Figures 2–4, the asymmetry introduces significant changes in the gradient histogram, and consequentially affects the orientation and the descriptor of the keypoints, which increases the difficulty in matching the keypoints.



Figure 2. Test images containing varying degrees of distortion: 10%, 20%, 30%, and 40%, in order from left to right.



Figure 3. Detection estimation of the keypoints using the test images. Y-axis is the number of keypoints.



Figure 4. Recall vs. 1-precision.

In Tri-SIFT, we calculate the area of the surface in a region instead of the gradient such that the number of keypoints is invariant to orientation. An image is described in the three-dimensional space

using the coordinates of the horizontal pixel axis, vertical pixel axis, and grayscale. We assume that the gradient of a slope with a fixed orientation is *A*. If the slope has 5 pixels on it, the histogram on the particular location and orientation adds 5*A*. If the slope has 10 pixels on it, the histogram will add 10*A*. This indicates a significant difference. However, the area of the slope is fixed, and irrespective of the number of pixels on the slope, the sum of the areas is fixed.

To calculate the area of the surface, we triangulate the set of points P using Delaunay triangulation, which has a time complexity of O(nlogn). The Delaunay triangulation for a set of points P in a plane is a triangulation DT(P) such that no point P is inside the circumcircle of any triangle in DT(P), as shown in (6). *V* is the vertices of the polygon, and *E* is the edge between the vertices. Delaunay triangulations maximize the minimum angle of all the angles of the triangles in the triangulation, while avoiding skinny triangles.

$$DT = (V, E) \tag{6}$$

We use Delaunay triangulation to calculate the orientation in Section 4.3 and the descriptor in Section 4.4.

4.3. Orientation

In the tri-SIFT, we do not calculate the gradient at each point in the region to determine the dominant orientation or descriptor; instead, the gradient at each triangle is obtained using Delaunay triangulation. The gradient magnitude and orientation are replaced by the area and the normal component of the triangle, respectively.

The sphere on which the image is located is a compact manifold of constant positive curvature. After the image has been back-projected to the sphere, each point, in spherical coordinates, is a three-dimensional vector. We define the point set as P_s :

$$P_{s} = \left\{ p : p = (\operatorname{rsin}\theta \cos\varphi, \operatorname{rsin}\theta \sin\varphi, \operatorname{rcos}\theta)^{T} \middle| \varphi \in [0, 2\pi), \theta \in [0, \pi), r = 1 \right\}$$
(7)

Simultaneously, we define another point set as

$$P_g = \left\{ p : p = \left(g \sin \theta \cos \varphi, g \sin \theta \sin \varphi, g \cos \theta \right)^T \middle| \varphi \in [0, 2\pi), \theta \in [0, \pi), g \in [0, 1] \right\}$$
(8)

Obviously, the elements of P_g and P_s have the following relation functions,

$$p_s = f_{gs}(\mathbf{p}_g) \tag{9}$$

$$p_g = f_{gs}^{-1}(p_s)$$
 (10)

in which $p_s \in P_s$, $p_g \in P_g$. Then, for each considered keypoint of the sphere, we calculate the orientations of the surrounding points on the circular region with a radius 3σ , which is centered at the keypoint (where σ is the scale at which each keypoint is located). To define this region, the distance between two points on the unit sphere, $p_{s1} \equiv (\theta_1, \varphi_1)$ and $p_{s2} \equiv (\theta_2, \varphi_2)$, must be calculated. The distance can be obtained using Vincenty's formulae. The angular distance $\Delta\sigma$ is

$$\Delta \sigma = \arctan\left(\frac{\sqrt{(\sin\theta_2 \sin\Delta\varphi)^2 + (\sin\theta_1 \cos\theta_2 - \cos\theta_1 \sin\theta_2 \cos\Delta\varphi)^2}}{\cos\theta_1 \cos\theta_2 + \sin\theta_1 \sin\theta_2 \cos\Delta\varphi}\right)$$
(11)

and the distance between the two points is

$$d = r\Delta\sigma \tag{12}$$

The points on the plane are back-projected to the unit sphere, and the distance between the two points on the plane is not the same as that on the surface of the unit sphere. In a fish-eye image with radial distortion, the distance between two adjacent pixels near the principal point is different from that near the edges. According to the fish-eye camera model, the distortion at the principal point is almost 0. Therefore, at the same scale, we back project two adjacent pixels at the principal point on the plane to the sphere and compute the distance μ_{fg} using (11) and (12). We take this value as the unit of measurement on the sphere and we can obtain the transformation of the distance on the plane using (13)

$$dis_{sphere} = \mu_{fg} dis_{flat} \tag{13}$$

where dis_{sphere} is the distance between the two points on the plane, and dis_{flat} is the distance between the two points on the sphere.

Assuming that p_{se} is a keypoint in P_s , we select a circular window with the center at p_{se} and radius $3\mu_{fg}\sigma$. The keypoints P_{sori} within the circular window are shown in Figure 5. Then, we obtain another point set $P_{gori} = f_{gs}(P_{sori})$, as shown in Figure 6. The point set P_{gori} is Delaunay triangulated to the triangle set S_{tris} , as shown in Figure 6:



Figure 5. The point cloud. The red points are in the circular window. The left figure shows the point set P_{sori} . The right figure shows the point set P_{gori} . X,Y,Z-axis represent rsin $\theta \cos \varphi$, rsin $\theta \sin \varphi$, rcos θ .



Figure 6. Result of triangulating the point set P_{gori} . X,Y,Z-axis represent gsin $\theta \cos \varphi$, gsin $\theta \sin \varphi$, gcos θ .

Each triangle in S_{tris} is calculated individually. Let us define two vectors $\overrightarrow{G_1G_3}$ and $\overrightarrow{G_1G_2}$ to represent the two edges of the triangle (cf. Figure 7). We have

$$\vec{G_1G_3} \times \vec{G_1G_2} = \begin{pmatrix} \vec{\alpha_1} & \vec{\alpha_2} & \vec{\alpha_3} \\ g_1 & g_2 & g_3 \\ h_1 & h_2 & h_3 \end{pmatrix} = (n_{\alpha_1}, n_{\alpha_2}, n_{\alpha_3})^T = \vec{n_{tri}}$$
(14)



Figure 7. A triangle after triangulation.

We use the incenter of the triangle as its location. The incenter is the center of the inscribed circle and must be located in the triangle. Compared with the circumcenter and other representations, the incenter is more representative of the location of the triangle. The incenter O_{tri} is obtained by

$$\overrightarrow{G_2G_3} \times \overrightarrow{O_{tri}G_1} + \overrightarrow{G_1G_3} \times \overrightarrow{O_{tri}G_2} + \overrightarrow{G_1G_2} \times \overrightarrow{O_{tri}G_3} = \overrightarrow{0}$$
(15)

After we have obtained the normal and location of each triangle, we calculate the dominant orientation based on this information. Unlike SIFT, we compute the area of the triangle instead of the gradient magnitudes by using

$$A = \frac{1}{2} \| \overrightarrow{n_{tri}} \| \tag{16}$$

where *A* is the area of a triangle.

We denote the point $O_{tri}(\theta, \varphi, g)$ as the location of a triangle and obtain $O_{stri} = f_{gs}(O_{tri})$, as shown in Figure 8. In the Cartesian coordinate system α (the basis is $(\vec{\alpha}_1, \vec{\alpha}_2, \vec{\alpha}_3)$), the coordinates of O_{stri} are

$$O_{stri} = (k_1, k_2, k_3)^T = (\cos\varphi\sin\theta, \sin\varphi\sin\theta, \cos\theta)^T$$
(17)



Figure 8. α and β coordinate systems.

To facilitate the computation of the orientation of the triangle, we convert the basis from $(\vec{\alpha}_1, \vec{\alpha}_2, \vec{\alpha}_3)$ to $(\vec{\beta}_1, \vec{\beta}_2, \vec{\beta}_3)$, where $\vec{\beta}_3$ is a unit vector of \vec{OP} ; $\vec{\beta}_2$ is the unit vector of the tangent vector that is in the meridian through O_{stri} and $\vec{\beta}_1$ is determined by $\vec{\beta}_2$ and $\vec{\beta}_3$ according to the right-hand rule

$$\alpha = \left(\stackrel{\rightarrow}{\alpha}_1, \stackrel{\rightarrow}{\alpha}_2, \stackrel{\rightarrow}{\alpha}_3 \right) = I_n \tag{18}$$

$$\vec{\beta}_{20} = k_1 \vec{\alpha}_1 + k_2 \vec{\alpha}_2 + k_3 \vec{\alpha}_3,$$

$$\vec{\beta}_{30} = -k_1 \vec{\alpha}_1 - k_2 \vec{\alpha}_2 + \frac{\sin^2 \theta}{\cos \theta} \vec{\alpha}_3,$$

$$\vec{\beta}_{10} = \vec{\beta}_{20} \times \vec{\beta}_{30}$$
(19)

$$\beta = \left(\vec{\beta}_{1}, \vec{\beta}_{2}, \vec{\beta}_{3}\right) = \left(\frac{\vec{\beta}_{10}}{\left\|\vec{\beta}_{10}\right\|}, \frac{\vec{\beta}_{20}}{\left\|\vec{\beta}_{20}\right\|}, \frac{\vec{\beta}_{30}}{\left\|\vec{\beta}_{30}\right\|}\right)$$
(20)

where α is the basis of the original coordinate system, and β is the basis of the new coordinate system. Because α is a unit matrix, we compute the transitional matrix $T_{\alpha\beta}$ using

$$\beta = \alpha T_{\alpha\beta} \Rightarrow T_{\alpha\beta} = \alpha^{-1}\beta = \beta$$
(21)

According to $T_{\alpha\beta}$, we can convert $\overrightarrow{n_{tri}}$ to $\overrightarrow{n_{\beta tri}}$ into the β coordinate system

$$\overrightarrow{n_{\beta tri}} = (n_{\beta 1}, n_{\beta 2}, n_{\beta 3})^T = T_{\alpha \beta} \overrightarrow{n_{tri}}$$
(22)

We define the orientation of the triangle by

$$\phi_{ori} = \arctan(\frac{n_{\beta 2}}{n_{\beta 1}}) \tag{23}$$

The area value of each triangle adds to the histogram after being weighted by a Gaussian centered on the keypoint with 1.5 times that of the keypoint.

Finally, once the histogram has been computed, the dominant orientation is calculated. If there are bins greater than 0.8 times the biggest bin, they are also considered. This results in multiple dominant orientations for the same point.

The pseudo algorithm for the computation of the dominant orientation is presented as Algorithm 1. The pseudo algorithom for the computation of the dominant orientation is presented as Algorithm 1.

Algorithm 1 Algorithm for the computation of the dominant orientation

- 1. for each considered keypoint (x_i, y_i) **do**
- 2. $bin \leftarrow \emptyset$ the histogram of orientations
- 3. $(x_i, y_i) \rightarrow (\theta_i, \varphi_i, 1)$ back-projected to the unit sphere
- 4. Select a circular region of size $3\mu_{fg}\sigma$ centered at $(\theta_i, \varphi_i, 1)$ and obtain the point set P_{sori}

5. Obtain the point set
$$P_{gori} = \{ p_g : p_g = f_{gs}^{-1}(p_s) | p_s \in P_{sori} \}$$

- 6. Triangulate the point set P_{gori}
- 7. bin-Compute the orientation and the area of each triangle after being weighted by Gaussian operators
- 8. max←maximum value inside bin
- 9. **for** each bin value $\geq 0.8 \max \mathbf{do}$
- 10. create a feature with corresponding orientation
- 11. end for
- 12. end for

4.4. The Descriptor Construction

The descriptors of the considered keypoints are computed using their corresponding dominant orientations as reference. This descriptor is a three-dimensional histogram of orientations (two spatial dimensions and one dimension for orientations) in which all the orientations are considered with respect to the dominant orientation.

9 of 15

We set a keypoint p_{se} as the center, with a circular window of radius $r = 3\mu_{fg}\sigma$ to compute the descriptors. All points within the window are represented by the point set P_{sdes} . To achieve invariance to rotation, *P_{sdes}* is rotated by the angle of the dominant orientation. As shown in Figure 8, we define the dominant orientation in the β coordinate system, and convert the coordinates of all points in the window from the α to the β coordinate system. After the rotation, the coordinate of $p'_{sdes\beta}$ is converted to p'_{sdes} in the original α coordinate system. The calculation process is given below

$$p_{sdes\beta} = (\mathbf{k}_1, \mathbf{k}_2, \mathbf{k}_3)^T = T_{\alpha\beta} p_{sdes}$$
⁽²⁴⁾

$$p'_{sdes\beta} = \begin{pmatrix} \cos\phi_{ori} & -\sin\phi_{ori} & 0\\ \sin\phi_{ori} & \cos\phi_{ori} & 0\\ 0 & 0 & 1 \end{pmatrix} p_{sdes\beta}$$
(25)

$$p'_{sdes} = T_{\alpha\beta}^{-1} p'_{sdes\beta} \tag{26}$$

where $p_{sdes} \in P_{sdes}, p'_{sdes} \in P'_{sdes}$. With the new point set P'_{sdes} , we triangulate the point set $f_{gs}^{-1}(P'_{sdes})$ and obtain the set of triangles. The method for computing the location and the orientation of each triangle is described in Section 4.3. The form of our descriptor is an extension of GLOH, whose histogram has 17×8 bins (17 bins for the spatial dimension and 8 bins for the orientations). The descriptor is constructed on the sphere, and so, a square window such as the one in SIFT, is difficult to select, which increases the difficulties in the calculation. GLOH [19] is based on a circular window, which is easy to calculate on the sphere, and thus, its performance is better than SIFT.

Our descriptor is computed for a log-polar location grid with 3 bins in the radial orientation (the radius is set to 0.2, 0.5 and 1.0 of the original radius) and 8 bins in the angular orientation, which results in 17 location bins. Note that the central bin is not divided in angular orientations to avoid sudden changes in the location of the window. The gradient orientations are quantized in 8 bins. The central bin is not divided in angular orientations to avoid sudden changes in the location of the window. The gradient orientations are quantized in 8 bins. Each bin value corresponds to the weighted sum of the surface areas of the triangles, which are triangulated from the set of points inside the window, at the spatial location and orientation defined by the bin. The weight value is defined by a Gaussian centered on the keypoint and having a standard deviation of 1.5 .

To avoid boundary effects, the values of each area sample are distributed by linear interpolation into adjacent histogram bins. The resulting histogram is normalized; each bin has a threshold of 0.2 and is normalized again, in order to make the histogram robust to contrast changes. The algorithm for computing the Local Spherical Descriptors is summarized in Algorithm 2.

Algorithm 2 Algorithm for the computation of LSD

- 1. **for** each considered keypoint (x_i, y_i) **do**
- 2. $bin \leftarrow \emptyset$ the histogram of orientations
- 3. $(x_i, y_i) \rightarrow p_{se}(\theta_i, \varphi_i, 1)$ back-projected to the unit sphere
- 4. Computation the dominant orientation ϕ_{ori}
- 5. $P_{sdes} \leftarrow$ the points in a circular window (the center is p_{se} , the radius is $3\mu_{fg}\sigma$)
- 6. Compute the transition matrix of p_{se}
- 7. P_{sdes} is rotated into P'_{sdes} with the angle ϕ_{ori}
- 8. Triangulating the point set $f_{gs}^{-1}(P'_{sdes})$
- 9. The window is divided into 17 bins
- 10. for each triangle do
- 11. Compute the orientation and the location of the triangle
- 12. Determine the location in the histogram using the angle, the orientation and the distance
- 13. bin←compute the area of the triangle after being weighted by Gaussian operators
- 14. end for
- 15. Descriptor vector ← transform bin
- 16. **end for**

5. Experiment

To determine the losses of generality, the experimental image data contain various configurations of the camera: scaling, translation, affine transformation, and varying degrees of distortion. To test the matching performance in fish-eye images, tri-SIFT is compared with the standard SIFT algorithm, rect-SIFT (image correction before using SIFT algorithm) and RD-SIFT (radial distortion SIFT) [10].

Figure 9 shows the experiments panoramic image pairs. In Figure 9a, the scale of left images is different to the scale of right images, and each two different scaled image is an image pair. In Figure 9b, the translation of left images is different to the translation of right images, which are captured by the same camera in the same scale. In Figure 9c, the left images have different affined angle with the right images, which are captured by the same camera. We matched the images in Figure 9 with the standard SIFT algorithm, rect-SIFT, RD-SIFT and tri-SIFT, sequentially. We removed the false matching points by using the RANSAC (Random Sample Consensus) algorithm to obtain the appropriate match points and plotted the 1-precision versus recall curves of the four algorithms, as shown in Figure 10. By observing and comparing the various curves, we can see that the tri-SIFT algorithm shows a generally good performance in terms of the distortion degrees and various changes in scale, translation, and affine. The more distortion degrees, the worse matching results performed. While in the four methods, the matching result of proposed method has the smallest impact of distortion degrees, and the matching result of standard SIFT is most seriously influenced by the distortion degrees. The standard SIFT algorithm can obtain more points at a 10% degree of distortion. However, without any compensation for distortions in fisheye images, the performance of the standard SIFT dramatically decreases when the degree of distortion is more than 20%. For the RD-SIFT algorithm, the performance is better at 10% and 20% degrees of distortion. However, when the degree of distortion continues to increase, the performance is not as exceptional, although it is still better than that of the standard SIFT and worse than that of the rect-SIFT. The proposed tri-SIFT algorithm is superior to the rect-SIFT in terms of performance at small degrees of distortion (10% and 20%), but it is inferior to the RD-SIFT algorithm. However, in the case of a smaller number of match points, the proposed method shows better matching performance. In tri-SIFT, the calculation of the points is replaced by the calculation of the triangles. The method is thus more adaptable in the large distortion region and can obtain more initial and accurate match points.



Figure 9. Image pairs used in the experiment: the numbers in the left column are degree of distortion percentages; (**a**) shows the scaled image pairs; (**b**) shows the translated image pairs; (**c**) shows the affined image pairs.



Figure 10. 1-precision vs. recall curves of the standard SIFT, rect-SIFT, RD-SIFT and tri-SIFT algorithms.

The influence of the algorithms considering various poses and orientations is shown in Table 1 and Figure 11. In Table 1, we list the resulting matches of the standard SIFT, rect-SIFT, RD-SIFT and tri-SIFT for various changes in the camera pose (near-far, translation, affine) and a degree of distortion of 20%. In Table 1, the initial match is keypoints matching without RANSAC algorithm, and correct match is the keypoints matching using RANSAC algorithm. Since there are some mismatching keypoints, the initial match can match more keypoints than the correct match. Compared with the SIFT, the tri-SIFT improves the matching performance by 24.5%, 12.1%, and 10.6% under the conditions of scaling, translating and affine, respectively. Compared with the other three methods, under the influence of conditions of scaling, translating and affine, respectively. although RD-SIFT can get the

13 of 15

highest initial match, correct match is not as high as Tri-SIFT. Besides, according to the initial match and correct match numbers, our method is much more than the standard SIFT and rect-SIFT. While, considering the correct match can more obviously reflect the matching performance, our method gets the most correct match numbers. Thus, our method can get the best matching performance. From the data shown in Figure 11, we analyze the results of the standard SIFT and tri-SIFT, and the distribution of the match points obtained by the standard SIFT at the center of the image. The distortion at the image center is negligible. The distortion and scale on the image periphery are remarkable, which makes the standard SIFT algorithms unsuitable to be applied to the peripheral area. When the images have translation and affine distortion simultaneously, the matching becomes more complicated. However, because of tri-SIFT concerns about distortion, the matching points obtained by tri-SIFT can be distributed anywhere in the images, as shown in Figure 11b.

	SIFT		Rect-SIFT		RD-SIFT		Tri-SIFT	
_	Initial Match	Correct Match	Initial Match	Correct Match	Initial Match	Correct Match	Initial Match	Correct Match
Scale	170	153	238	198	251	209	243	216
Translation	97	83	102	91	116	89	108	98
Affine	114	102	128	106	151	119	136	125

Table 1. M	latching results	of the four	algorithms.
------------	------------------	-------------	-------------



(a)



(b) Figure 11. Matching Results: (a) for SIFT (b) for tri-SIFT.

6. Conclusions

In this study, we investigated the problem of matching feature points in fisheye images. A triangulation-based detection and matching algorithm in fish-eye images combined with the camera's imaging model to eliminate the impact of distortion was proposed. This paper has demonstrated how radial distortion affects the performance of the original Scale SIFT algorithm. Then, we proposed the method that calculates the area of the surface in a region instead of the gradient such that the number of keypoints is invariant to orientation and the Delaunay triangulation is used to calculate the orientation and the descriptor. The experiments validate the robustness of the proposed method to distortion and demonstrate the achieved high efficiency in matching feature points in large distortion areas. Compared with SIFT algorithm, rect-SIFT and RD-SIFT, the proposed method can achieve the best matching performance. Besides, the Tri-SIFT can be applied into several robot vision tasks, 3D reconstruction based on panoramic images and other matching tasks with large distortion areas.

However, the proposed method also cannot match the image pairs with larger light effects correctly. The keypoints in images captured from the ground and images captured from sky, which have large affine transformation, cannot be match well. This kind of matching problem also needs other information to help the keypoints to match. Thus, the matching problems for the images captured from sky with large affine influence is the key future work, and it is useful for many driving automations, detailed 3D reconstruction and so on.

Author Contributions: Data curation, D.L.; Methodology, J.J.; Project administration, E.W.; Resources, J.T.; Software, J.Y.; Visualization, J.J.; Writing—original draft, E.W.; Writing—review & editing, J.J.

Funding: This research was funded by Natural Science Foundation of China grant number 61401455 and the authors also thank the support by Youth Innovation Promotion Association CAS.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Barreto, J.; Santos, J.M.; Menezes, P.; Fonseca, F. Ray-based calibration of rigid medical endoscopes. Available online: https://www.researchgate.net/publication/29622069_Raybased_Calibration_of_Rigid_Medical_ Endoscopes (accessed on 23 November 2018).
- Baker, P.; Fermuller, C.; Aloimonos, Y.; Pless, R. A spherical eye from multiple cameras (makes better models of the world). In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Kauai, HI, USA, 8–14 December 2001.
- 3. Liu, N.; Zhang, B.; Jiao, Y.; Zhu, J. Feature matching method for uncorrected fisheye lens image. In *Seventh International Symposium on Precision Mechanical Measurements*; SPIE Press: Bellingham, WA, USA, 2016.
- 4. Kim, D.; Paik, J. Three-dimensional simulation method of fish-eye lens distortion for a vehicle backup rear-view camera. *JOSA A* **2015**, *32*, 1337–1343. [CrossRef] [PubMed]
- Zhang, B.; Jia, Y.; Röning, J.; Feng, W. Feature matching method study for uncorrected fish-eye lens image. In *Intelligent Robots and Computer Vision XXXII: Algorithms and Techniques*; SPIE Press: Bellingham, WA, USA, 2015.
- 6. Hansen, P.I.; Corke, P.; Boles, W.; Daniilidis, K. Scale-invariant features on the sphere. In Proceedings of the IEEE 11th International Conference on Computer Vision, Rio de Janeiro, Brazil, 14–20 October 2007.
- 7. Hansen, P.; Corke, P.; Boles, W. Wide-angle visual feature matching for outdoor localization. *Int. J. Robotics Res.* **2010**, *29*, 267–297. [CrossRef]
- 8. Miyamoto, K. Fish eye lens. JOSA 1964, 54, 1060–1061. [CrossRef]
- 9. Lowe, D.G. Object recognition from local scale-invariant features. In Proceedings of the International Conference on Computer Vision, Kerkyra, Greece, 20–27 September 1999; pp. 1150–1157.
- 10. Lourenço, M.; Barreto, J.P.; Vasconcelos, F. Srd-sift: Keypoint detection and matching in images with radial distortion. *T-RO* **2012**, *28*, 752–760. [CrossRef]
- 11. Hughes, C.; Denny, P.; Glavin, M.; Jones, E. Equidistant fish-eye calibration and rectification by vanishing point extraction. *TPAMI* **2010**, *32*, 2289–2296. [CrossRef] [PubMed]

- 12. Se, S.; Lowe, D.; Little, J. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *Int. J. Robotics Res.* **2002**, *21*, 735–758. [CrossRef]
- Bay, H.; Ess, A.; Tuytelaars, T.; Van Gool, L. Speeded-up robust features (SURF). *Comput. Vis. Image Und.* 2008, 110, 346–359. [CrossRef]
- 14. Wan, L.; Li, X.; Feng, W.; Long, B.; Zhu, J. Matching method of fish-eye image based on CS-LBP and MSCR regions. Available online: http://cea.ceaj.org/EN/column/column105.shtml (accessed on 22 November 2018).
- 15. DING, C.X.; JIAO, Y.K.; LIU, L. Stereo Matching Method of Fisheye Lens Image Based on MSER and ASIFT. Available online: http://en.cnki.com.cn/Article_en/CJFDTOTAL-HGZD201801008.htm (accessed on 22 November 2018).
- 16. Kannala, J.; Brandt, S.S. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *TPAMI* **2006**, *28*, 1335–1340. [CrossRef] [PubMed]
- 17. Arfaoui, A.; Thibault, S. Fisheye lens calibration using virtual grid. *Appl. Opt.* **2013**, *52*, 2577–2583. [CrossRef] [PubMed]
- Kanatani, K. Calibration of Ultrawide Fisheye Lens Cameras by Eigenvalue Minimization. *TPAMI* 2013, 35, 813–822. [CrossRef] [PubMed]
- Mikolajczyk, K.; Schmid, C. A performance evaluation of local descriptors. *TPAMI* 2005, 27, 1615–1630. [CrossRef] [PubMed]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).