TECHNICAL RESEARCH REPORT

Ego-Motion Estimation using Fewer Image Feature Points

by Morimichi Nishigaki, 2003-2004 ISR Visiting Scientist,
Honda R&D

TR 2004-32

The Final Report

Visiting Scientist Program

Honda R&D Company, Japan

Institute for Systems Research

The University of Maryland

Morimichi Nishigaki

7/22/2004

Preface

Environment recognition using images is a worthwhile research subject. 3D information is valuable information to recognize surrounding, because currently it is difficult for machines to understand 3D structure in the scene with only one image, even though a human can understand the scene structure from a picture.

3D information can be obtained from stereo and motion disparities. Since the stereo camera is assumed to be calibrated, the 3D shape can be calculated from disparities. On the other hand, since the camera motion, so called ego-motion, is not known, even though motion disparities is obtained, 3D shape cannot be calculated until ego-motion is estimated. In addition, the 3D shape recovered from motion disparities is determined up to a scale. Generally, the accuracy of 3D reconstruction by stereo camera depends on the baseline. A stereo camera usually cannot take long baseline, while camera motion can produce long baseline.

Since each stereo and motion has information on 3D shape, the combination of stereo and motion disparities could complementally produce better 3D reconstruction than only one of them. Researches in this report were taken placed toward the 3D reconstruction using stereo and motion information.

Acknowledgment

On the occasion of the ending of my research at the University of Maryland in the Honda Visiting Scientist Program, I would like to express my gratitude to people involved this program. Fortunately, I could have professor Aloimonos as my faculty host. Since he allowed me take his class, I could learn a lot from him. I could also take Dr. Fermuller's class. She gave me many advices to my researches. I often visited Abhijit, who is a PhD candidate, to ask questions, and he replied to all my questions. Jan and Patrick also joined to answer my questions. Karen and Jeff took care of me to stay here comfortably. Although I could not have many chances to meet ISR director Dr. Abed, he gave me beneficial advices. People in the general affairs department in Honda R&D Americas supported my residence. Finally, I was very happy that I could spend a lot of time to study with Wei-Lun.

Contents

Part I

Researches

1 Geometorical Understanding of Calibration

1.1 One-dimensional Camera Calibration in a Two-dimensional Space

The objective is to find two-dimensional rigid motion parameters, such as rotation and translation, and calibration parameters, such as focal length, pixel size, and center of image from images of a one-dimensional calibration target. In a two-dimensional space, a one-dimensional camera can take one-dimensional images of a one-dimensional calibration target. After the images of the calibration target are taken, the homography matrixes can be computed. The homographies indicate the transformation from the coordinates on the calibration target to the coordinates on image. The coordinate on the calibration target will be referred as the global coordinate. The coordinate on the image will be referred to as the image coordinate. The coordinate in which the origin is camera center is referred to as the camera coordinate.

$$H = \begin{pmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{pmatrix} \tag{1}$$

In this formula, H is the homography matrix.

If the rotation, translation, and the calibration matrix are known, the transformation of coordinates should be as follows in the homogeneous coordinate, using the ration angle θ, the translation vector (t_x, t_y), the focal length f, and image center x_0:

$$RT = \begin{pmatrix} cos(\theta) & -sin(\theta) & t_x \\ sin(\theta) & cos(\theta) & t_y \end{pmatrix} \tag{2}$$

$$K = \begin{pmatrix} f & x_0 \\ 0 & 1 \end{pmatrix} \tag{3}$$

Thus, when a point on the calibration target is $M = (X, Y, 1)^T$ in the homogeneous coordinate of the global coordinate, and a corresponding point in the homogeneous coordinate of the image coordinate is $m = (x, y)^T$, then the transformation between the point is

$$m = K \cdot RT \cdot M \tag{4}$$

There is no loss of generality in assuming that the Y coordinate of any point on the calibration target in the global coordinate is zero. Then, the homography between $M' = (X, 1)^T$ and $m = (x, y)^T$ is supposed to be:

$$m = K \cdot L \cdot M \tag{5}$$

Where

$$L = \begin{pmatrix} cos(\theta) & t_x \\ sin(\theta) & t_y \end{pmatrix} \tag{6}$$

The homography should be

$$H = \alpha \cdot K \cdot L \tag{7}$$

α is an unknown scale factor.

This notation shows that after rigid motion, an affine transformation is applied. In thinking about this two-step transformation, the second one is

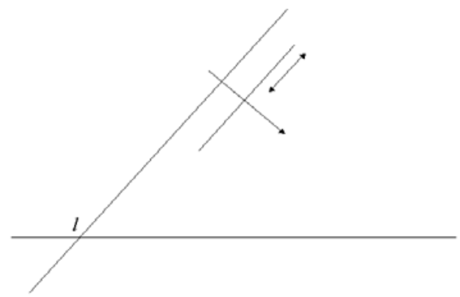

Figure 1: Calibration Target and Camera Coordinate System

only moving the image plane and stretching the plane without changing the angle of the image plane.

Thus, the point l where the calibration target, (the x-axis of the global coordinate) and the x-axis of the camera coordinate intersect doesn't change in any matrix of K. This is equivalent to the property of the affine transformation, which doesn't change the vanishing line of canonical position.

$$K^T \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{8}$$

Thus,

$$\begin{aligned} (H^{-1})^{-T} \begin{pmatrix} 0 \\ 1 \end{pmatrix} &= \alpha \cdot L^T \cdot K^T \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ &= \alpha \cdot L^T \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ &= \begin{pmatrix} v_x \\ v_y \end{pmatrix} \end{aligned} \tag{9}$$

In this formula, K has no effect.

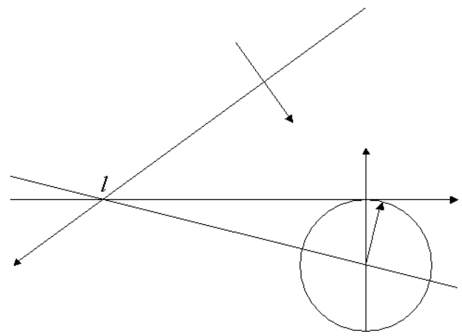

Figure 2: Intersection of X-axes

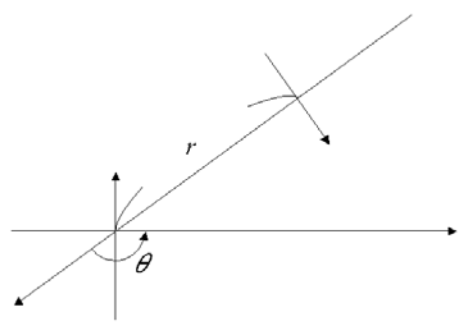

Figure 3: Retaken Coordinate Systems

When the homography is given, the point l can be computed as the intersect between the x-axis and the line that is perpendicular to $(v_x, v_y)^T$ and passes the point $(0, -1)^T$.

Since the point l can be computed from the homography matrix, it is possible to take the global coordinate where the point l is to be $(0, 0)^T$. Thus, without any loss of generality, it is assumed that the x-axis of the camera coordinate passes through the origin of the global coordinate.

When the angle between the x-axis of the camera coordinate and the x-axis of global coordinate is θ and the length between the origin of the camera coordinate and the origin of the global coordinate is r, the transformation matrix between coordinates is as follows:

$$RT = \begin{pmatrix} cos(\theta) & -sin(\theta) & r \\ sin(\theta) & cos(\theta) & 0 \end{pmatrix} \tag{10}$$

$$L = \begin{pmatrix} cos(\theta) & r \\ sin(\theta) & 0 \end{pmatrix} \tag{11}$$

So, the homography should be

$$\begin{aligned} H &= \alpha \cdot K \cdot L \\ &= \alpha \begin{pmatrix} fcos(\theta) + x_0 sin(\theta) & fr \\ sin(\theta) & 0 \end{pmatrix} \end{aligned} \tag{12}$$

Although the homography given before has three rigid motion parameters (θ, t_x, t_y), there are only two parameters (θ, r) in this notation. It is noticed that this reduction of parameters is made by the property of the calibration matrix as an affine transformation.

When the homography is given,

$$(H^{-1})^{-T} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = H^T \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} h_{21} \\ h_{22} \end{pmatrix} \tag{13}$$

the point l is computed as $(\frac{-h_{22}}{h_{21}}, 0)^T$ as the intersection of the x-axis and the line that is perpendicular to the vector $(h_{21}, h_{22})^T$ and passes through $(0, -1)^T$. So, the global coordinate can be redefined so that the point l is the origin. It is same as multiplying the homography by the following matrix.

$$\begin{pmatrix} 1 & \frac{-h_{22}}{h_{21}} \\ 0 & 1 \end{pmatrix} \tag{14}$$

So, the homography becomes:

$$H' = \begin{pmatrix} h_{11} & \frac{-h_{12}h_{22}}{h_{21}} + h_{12} \\ h_{21} & 0 \end{pmatrix} \tag{15}$$

Thus, because the $(2, 2)$ component of the homography can always be zero, it is assumed that the homography given is:

$$H = \begin{pmatrix} h_{11} & h_{12} \\ h_{21} & 0 \end{pmatrix} \tag{16}$$

Although the calibration parameters are unknown, the form of K^{-1} is as follows:

$$K^{-1} = \begin{pmatrix} \frac{1}{f} & \frac{-x0}{f} \\ 0 & 1 \end{pmatrix} \tag{17}$$

Multiply the homography by the K^{-1}

$$\begin{aligned} A &= K^{-1} \cdot H \\ &= \begin{pmatrix} \frac{h_{11}-x_0h_{21}}{f} & \frac{h_{12}}{f} \\ h_{21} & 0 \end{pmatrix} \end{aligned} \tag{18}$$

Since A should be the rigid motion matrix multiplied scale, the first column can be decomposed to the scale and the normal vector to make the length of the first column to be one.

$$\begin{aligned} A &= \frac{\sqrt{(h_{11} - x_0h_{21})^2 + f^2h_{21}^2}}{f} \begin{pmatrix} \frac{h_{11}-x_0h_{21}}{\sqrt{(h_{11}-x_0h_{21})^2+f^2h_{21}^2}} & \frac{h_{12}}{\sqrt{(h_{11}-x_0h_{21})^2+f^2h_{21}^2}} \\ \frac{fh_{21}}{\sqrt{(h_{11}-x_0h_{21})^2+f^2h_{21}^2}} & 0 \end{pmatrix} \\ &= \alpha L \end{aligned} \tag{19}$$

Where,

$$\alpha = \frac{\sqrt{(h_{11} - x_0 h_{21})^2 + f^2 h_{21}^2}}{f} \tag{20}$$

$$L = \begin{pmatrix} \frac{h_{11} - x_0 h_{21}}{\sqrt{(h_{11} - x_0 h_{21})^2 + f^2 h_{21}^2}} & \frac{h_{12}}{\sqrt{(h_{11} - x_0 h_{21})^2 + f^2 h_{21}^2}} \\ \frac{f h_{21}}{\sqrt{(h_{11} - x_0 h_{21})^2 + f^2 h_{21}^2}} & 0 \end{pmatrix} \tag{21}$$

Comparing this equation to equation (11), then :

$$\theta = sin^{-1}\left(\frac{f h_{21}}{\sqrt{(h_{11} - x_0 h_{21})^2 + f^2 h_{21}^2}}\right) \tag{22}$$

$$r = \frac{h_{12}}{\sqrt{(h_{11} - x_0 h_{21})^2 + f^2 h_{21}^2}} \tag{23}$$

Since this defines the map from (θ, r) to (f, x_0), the inverse map can be defined as :

$$f = \frac{h_{12}}{h_{21}} \frac{sin\theta}{r} \tag{24}$$

$$x_0 = \pm \frac{h_{12}}{h_{21}} \frac{cos\theta}{r} + \frac{h_{11}}{h_{21}} \tag{25}$$

When the r is constant, the graph for (f, x_0) becomes a circle with a parameter of θ,

$$f^2 + \left(x_0 - \frac{h_{11}}{h_{21}}\right)^2 = \left(\frac{h_{12}}{h_{21}}\right)^2 \frac{1}{r^2} \tag{26}$$

As expected, the θ should be between 0 and π and r is positive to take the image of the calibration target, and f should be positive. Knowing this, the half of the circle of the equation (26) is comprise of candidates of calibration parameters.

Fig.4 shows the candidates of calibration parameters. Actually, the candidates cover half of the (f, x_0) space.

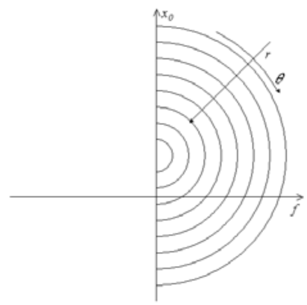

Figure 4: Trace of Calibration Parameters

Calibration is identifying one of the candidates in (f, x_0) space from images. In a one-dimensional camera in a two-dimensional space, if other images of the calibration target are taken, other circles in other centers are drawn. Then the intersections of the circles are the calibration parameters to satisfy all images. But even though a lot of images of calibration targets are taken, the intersections cannot be identified because the circles cover half of the (f, x_0) space.

Thus, a one-dimensional camera in a two-dimensional space cannot be calibrated by using images of a calibration target when the extrinsic parameters between the camera and the calibration target are unknown.

1.2 Two-dimensional Camera Calibration in a Three-dimensional Space

From the explicit form of the map between (θr) and (f, x_0) in a one-dimensional camera case, the meaning of calibration becomes clear to identify the intersection of mapped curves in calibration parameters space. In this section, the same study is done in a two-dimensional camera case.

First, the image of the calibration target is taken; then, the homography between the calibration target and the image is computed. It is supposed to be

$$H = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \tag{27}$$

When the rotation and translation follows

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \tag{28}$$

$$T = \begin{pmatrix} t_x & t_y & t_z \end{pmatrix}^T \tag{29}$$

, and the calibration matrix is

$$K = \begin{pmatrix} f_x & 0 & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{30}$$

, then the homography is

$$H = \alpha \cdot K \cdot L \tag{31}$$

Here, since the calibration target is assumed to be on the x-y plane in a global coordinate,

$$L = \begin{pmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ r_{31} & r_{32} & t_z \end{pmatrix} \tag{32}$$

and α is an unknown scale.

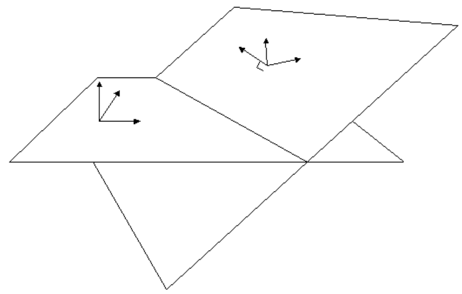

Figure 5: Intersection of x-y Plane between Camera and Global Coordinate Systems

Since the calibration matrix is an affine matrix, there are no effects of K when the vanishing line in the canonical position is mapped from image to the calibration target plane.

$$(H^{-1})^T \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}^T = \begin{pmatrix} h_{31} & h_{32} & h_{33} \end{pmatrix}^T = \alpha \begin{pmatrix} r_{31} & r_{32} & t_z \end{pmatrix}^T \tag{33}$$

It means that even if the calibration parameters are unknown, the intersection between the x-y plane of the camera coordinate and the x-y plane of the global coordinate is known. Line l indicates this line of intersection.

The line l is represented as

$$h_{31}x + h_{32}y + h_{33} = 0 \tag{34}$$

By multiplying H by

$$S = \begin{pmatrix} \frac{h_{31}}{\sqrt{h_{31}^2+h_{32}^2}} & \frac{h_{32}}{\sqrt{h_{31}^2+h_{32}^2}} & -\frac{h_{33}h_{31}}{h_{31}^2+h_{32}^2} \\ \frac{h_{32}}{\sqrt{h_{31}^2+h_{32}^2}} & \frac{-h_{31}}{\sqrt{h_{31}^2+h_{32}^2}} & -\frac{h_{33}h_{32}}{h_{31}^2+h_{32}^2} \\ 0 & 0 & 1 \end{pmatrix} \tag{35}$$

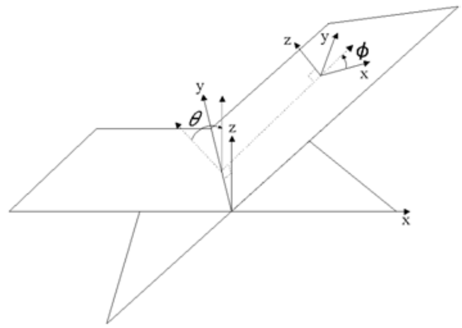

Figure 6: Retaken Coordinate Systems

, the intersection l becomes identical to the y-axis.

$$H \cdot S = \begin{pmatrix} \frac{h_{31}h_{11}+h_{32}h_{12}}{\sqrt{h_{31}^2+h_{32}^2}} & \frac{h_{32}h_{11}-h_{31}h_{12}}{\sqrt{h_{31}^2+h_{32}^2}} & -\frac{h_{33}(h_{31}h_{11}+h_{32}h_{12})}{h_{31}^2+h_{32}^2} \\ \frac{h_{31}h_{21}+h_{32}h_{22}}{\sqrt{h_{31}^2+h_{32}^2}} & \frac{h_{32}h_{21}-h_{31}h_{22}}{\sqrt{h_{31}^2+h_{32}^2}} & -\frac{h_{33}(h_{31}h_{21}+h_{32}h_{22})}{h_{31}^2+h_{32}^2} \\ \sqrt{h_{31}^2+h_{32}^2} & 0 & 0 \end{pmatrix} \tag{36}$$

Since the intersection l is known after the homography is calculated, it is assumed that the line l is identical to the y-axis. Thus, the homography's form is supposed to be

$$H = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & 0 & 0 \end{pmatrix} \tag{37}$$

Since the x-y plane of the camera coordinate includes the y-axis of the global coordinate, the rigid motion is described by only two rotation parameters and two translation parameters, thus

$$R = \begin{pmatrix} cos\phi & -sin\phi & 0 \\ sin\phi & cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} cos\theta & 0 & -sin\theta \\ 0 & 1 & 0 \\ sin\theta & 0 & cos\theta \end{pmatrix} \tag{38}$$

$$T = \begin{pmatrix} t_x & t_y & 0 \end{pmatrix}^T \tag{39}$$

$$L = \begin{pmatrix} cos\phi cos\theta & -sin\phi & t_x \\ sin\phi sin\theta & cos\phi & t_y \\ -sin\theta & 0 & 0 \end{pmatrix} \tag{40}$$

Even though the calibration matrix K is unknown, the relationship between the calibration parameters and the rigid motion parameters can be shown by multiplying H by K^{-1}.

$$K^{-1} \cdot H = \sqrt{(h_{12}/f_x)^2 + (h_{22}/f_y)^2} \cdot \begin{pmatrix} \frac{h_{11}/f_x - x_0 h_{31}/f_x}{\sqrt{(h_{12}/f_x)^2+(h_{22}/f_y)^2}} & \frac{h_{12}/f_x}{\sqrt{(h_{12}/f_x)^2+(h_{22}/f_y)^2}} & \frac{h_{13}/f_x}{\sqrt{(h_{12}/f_x)^2+(h_{22}/f_y)^2}} \\ \frac{h_{21}/f_y - y_0 h_{31}/f_y}{\sqrt{(h_{12}/f_x)^2+(h_{22}/f_y)^2}} & \frac{h_{22}/f_y}{\sqrt{(h_{12}/f_x)^2+(h_{22}/f_y)^2}} & \frac{h_{23}/f_y}{\sqrt{(h_{12}/f_x)^2+(h_{22}/f_y)^2}} \\ \frac{h_{31}}{\sqrt{(h_{12}/f_x)^2+(h_{22}/f_y)^2}} & 0 & 0 \end{pmatrix} \tag{41}$$

The result of this equation is supposed to be $\alpha \cdot L$, compared to equation

(40) :

$$\theta = sin^{-1}\left(\frac{-h_{31}}{\sqrt{(h_{12}/f_x)^2 + (h_{22}/f_y)^2}}\right) \tag{42}$$

$$\phi = cos^{-1}\left(\frac{h_{22}/f_y}{\sqrt{(h_{12}/f_x)^2 + (h_{22}/f_y)^2}}\right) \tag{43}$$

$$t_x = \frac{h_{13}/f_x}{\sqrt{(h_{12}/f_x)^2 + (h_{22}/f_y)^2}} \tag{44}$$

$$t_y = \frac{h_{23}/f_y}{\sqrt{(h_{12}/f_x)^2 + (h_{22}/f_y)^2}} \tag{45}$$

It should be noticed that there is no x_0 and y_0, and above equations define a map from the (f_x, f_y) space to the (θ, ϕ, t_x, t_y) space. It is because two of the four rigid motion parameters are not independent that equation (41) is supposed to be a rigid motion matrix; thus, the first column vector and second column vector of the matrix in equation (41) are supposed to be normal perpendicular.

The inverse of the map is

$$f_x = \pm\frac{h_{12}}{h_{31}}\frac{sin\theta}{cos\phi} \tag{46}$$

$$f_y = -\frac{h_{22}}{h_{31}}\frac{sin\theta}{sin\phi} \tag{47}$$

Comparing equation (40) and (41), the following equations hold :

$$x_0 = \frac{h_{11} \pm h_{12}\frac{cos\theta}{tan\phi}}{h_{31}} \tag{48}$$

$$y_0 = \frac{h_{21} \pm h_{22}cos\theta tan\phi}{h_{31}} \tag{49}$$

Equations (46),(47),(48), and (49) draw a two dimensional surface with parameters θ and ϕ in the calibration parameters space (f_x, f_y, x_0, y_0).

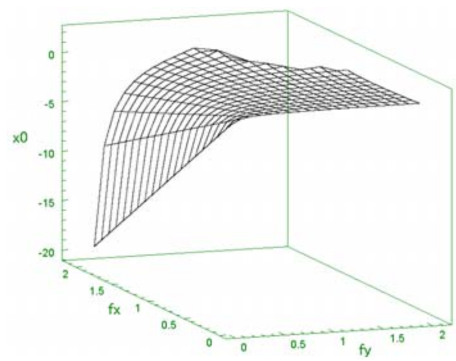

Figure 7: the Surface on which Calibration Parameters are Supposed Be

Fig.7 shows the projection on the (f_x, f_y, x_0) space of the surface in the (f_x, f_y, x_0, y_0) space.

The calibration means finding the intersection of the surface in this space. Since the surface is two-dimensional in four-dimensional space, the intersection of the two surfaces is a point. That point indicates the calibration parameters that hold in two views of the calibration target.

1.3 Conclution

The calibration is discussed in a two-dimensional space in order to understand it easily. The geometrical restriction of the relation between the camera coordinate and global coordinate is shown, due to affinity of the calibration matrix when the image of calibration target was taken in two-dimensional space. Even though the calibration for a one-dimensional camera was impossible, it became clear that calibration means finding a unique intersection of curves mapped from rigid motion space in calibration parameters space.

It is shown that the same things are happening in three-dimensional space. It became clear that rigid motion parameters are represented by only two rotation parameters and two translation parameters, thanks to the affinity of the calibration matrix. The map between the calibration parameters and the rigid motion parameters is revealed. In general, the unique intersection in calibration parameters space is found by two images of calibration target in different views.

Zhang's Algorithm which is often used for a single camera calibration. The Caltech's Calibration Toolbox and the calibration routine of Intel's Open Computer Vision Library are based on Zhang's Algorithm. The affine property of the calibration matrix is not used in it. So, it needs at least three images of calibration target in different views ([17][8]).

In some algorithms of calibration, the calibration parameters that are found by restriction of rigid motion are used as an initial value of optimization to reduce the error due to noise and distortion ([15][17]). The study in this report made the geometrical meaning and restrictions between rigid motion parameters and calibration parameters of the first step of those calibration algorithms clear.

It would be possible to look at the intersections of the surface that is described in the previous section and eliminate surfaces that produce intersections that are far from the others as wrong data. It would be possible to figure out in what case it becomes difficult to find the intersection or in what case the intersection becomes robust to noise geometrically.

2 Ego-Motion Estimation Using Fewer Image Feature Points

2.1 Background

Ego-motion estimation is useful for both 3-D shape reconstruction and motion segmentation. Ego-motion estimation algorithms are roughly classified into two categories. One is the area-based method, which utilize image gradients or optical flows all over the images. The other one is the feature-based method, which track feature points or find corresponding points between two images.

The area-based method can be used only for a small motion, because infinitesimal operations are necessary to measure image gradients or optical flows. When the motion is small, different motion can produce same motion on the imaging surface. Therefore, the area-based method has ambiguity due to small motion. Although the feature based method can be used for relatively large motion, finding corresponding points, so called matching problem, are difficult in general. When the motion is large, feature points looks different due to the viewpoint difference. Excluding independently moving area or points are challenging issue for both methods.

In order to avoid ambiguity due to a small motion, relatively large motion is used in this study; therefore, this study is classified as feature based method. Even though finding a corresponding point to an assigned point are difficult, corresponding points to fewer feature points can be found reliably. Therefore, a feature-based method using fewer image feature points is studied to estimate ego-motion.

2.2 Motion Parameterizations

The camera motion is usually represented by a rotation matrix and a translation matrix as Fig.8.

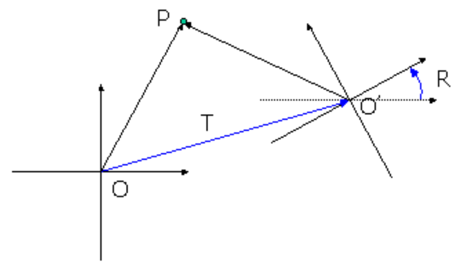

Figure 8: Motion Parametrization

In Fig.8, R is the rotation matrix and T is the translation matrix. O and O' are camera centers which are origins of camera coordinate systems. P is a 3-dimentional point. Motion parameters, R and T are measured in the reference coordinate system O in this case. The transformation between two coordinate systems is written as follows:

$$p' = R^{-1} \cdot (p - T) \tag{50}$$

The rotation and translation can be measured in the other coordinate system as Fig.9.

In Fig.9, $R^{'}$ is the rotation matrix and $T^{'}$ is the translation matrix. The transformation between coordinate systems is written as follows:

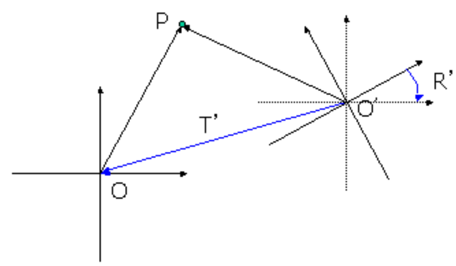

Figure 9: Motion Parameterizations

$$p = R^{'-1} \cdot \left(p' - T^{'}\right) \tag{51}$$

$$p' = R^{'} \cdot p + T^{'} \tag{52}$$

When the motion is parameterized by a rotation and translation, those parameters depend on the reference coordinate system. Therefore, the equation (50) and equation (52) are not equivalent.

The epipoles can be used as ego-motion parameters as Fig.10. Here, e

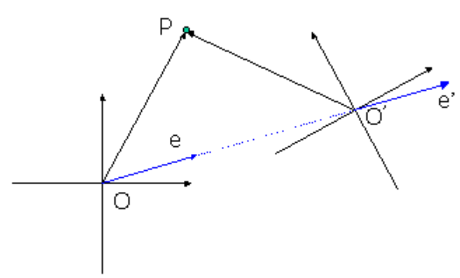

Figure 10: Motion Parametrization with Epipoles

and $e^{'}$ are epipoles of each camera respectively. Although $e^{'}$ is usually taken to the opposite direction, e and $e^{'}$ are taken as same direction in order to take the sign While a rotation and translation have six motion parameters, two epipoles have only four parameters. Therefore, other two parameters are necessary to determine the motion. One of them is the distance between two camera centers. The other one is the angle along the line going through two camera centers.

2.3 Epipolar Constrain with Epipoles

The epipolar constrain is written using a rotation and a translation as follows:

$$p^T \cdot R \cdot T_\times \cdot p' = 0 \tag{53}$$

Here, $T_\times$ is the matrix which satisfy $T_\times \cdot p = T \times p$. Since it is assumed that both cameras are calibrated, the camera matrices are identical matrices, and feature points are projected to the normal cameras. This equation can be rewritten using epipoles. While the rotation and translation matrix have five parameters, two epipoles have only four parameters. The missing parameter is a rotation angle along the line going through two camera centers. Instead of using the rotation angle, another pair of corresponding points p_1 and $p_1^{'}$ can be utilized to determine the motion. R and $R^{'}$ is defined as follows:

$$R = \begin{pmatrix} \frac{e \times p_1}{|e \times p_1|} \times e & \frac{e \times p_1}{|e \times p_1|} & e \end{pmatrix} \tag{54}$$

$$R^{'} = \begin{pmatrix} \frac{e^{'} \times p_1^{'}}{|e^{'} \times p_1^{'}|} \times e^{'} & \frac{e^{'} \times p_1^{'}}{|e^{'} \times p_1^{'}|} & e^{'} \end{pmatrix} \tag{55}$$

Then, the original epipolar constrain, the equation (53), is rewritten as follows:

$$p^T \cdot R \cdot R^{'T} \cdot (e' \times p^{'}) = 0 \tag{56}$$

While the original epipolar constrain (53) involves one pair of corresponding points and five motion parameters, the derived one (56) involves two pair of corresponding points and four motion parameters. Although motion parameters in the original epipolar constrain are combination of angles and direction, motion parameters in the derived one are two direction.

The equation (56) is written concretely using coordinate of two feature points and epipoles. It is assumed that the imaging surface is sphere; therefore, feature points are unit vectors. The two feature points can be moved to the points p_1 and p_2 as Fig.11 by a certain rotation. $p_1^{'}$ and $p_2^{'}$ are also rotated as well as p_1 and p_2 by another certain rotation. Then, p_1 and $p_1^{'}$ are on the north pole of the sphere respectively. p_2 and $p_2^{'}$ are on the grate circle on the $x - z$ plane respectively.

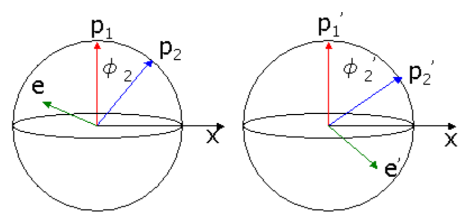

Figure 11: Locations of Feature Points and Epipoles

The coordinate of feature points and epipoles are written using longitude and latitude as follows:

$$p_1 = p_1^{'} = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}^T \tag{57}$$

$$p_2 = \begin{pmatrix} sin(\phi_2) & 0 & cos(\phi_2) \end{pmatrix}^T \tag{58}$$

$$p_2^{'} = \begin{pmatrix} sin(\phi_2^{'}) & 0 & cos(\phi_2^{'}) \end{pmatrix}^T \tag{59}$$

$$e = \begin{pmatrix} cos(\phi)sin(\theta) & sin(\phi)sin(\theta) & cos(\phi) \end{pmatrix}^T \quad (60)$$

$$e^{'} = \begin{pmatrix} cos(\phi^{'})sin(\theta^{'}) & sin(\phi^{'})sin(\theta^{'}) & cos(\phi^{'}) \end{pmatrix}^T \quad (61)$$

In this case, the R and $R^{'}$ in the equations (54) and (55) are written as follows:

$$R = \begin{pmatrix} cos\phi cos\theta & -sin\theta & sin\phi cos\theta \\ cos\phi sin\theta & cos\theta & sin\phi sin\theta \\ -sin\phi & 0 & cos\phi \end{pmatrix} \quad (62)$$

$$R^{'} = \begin{pmatrix} cos\phi^{'} cos\theta^{'} & -sin\theta^{'} & sin\phi^{'} cos\theta^{'} \\ cos\phi^{'} sin\theta^{'} & cos\theta^{'} & sin\phi^{'} sin\theta^{'} \\ -sin\phi^{'} & 0 & cos\phi^{'} \end{pmatrix} \quad (63)$$

Then, the constrain (56) is written as follows:

$$sin\theta sin\phi_2 \left\{ sin\phi^{'} cos\phi_2^{'} - cos\phi^{'} cos\theta^{'} sin\phi_2^{'} \right\}$$
$$-sin\theta^{'} sin\phi_2^{'} \left\{ sin\phi cos\phi_2 - cos\phi cos\theta sin\phi_2 \right\} = 0 \quad (64)$$

Multiplying the equation (64) by $sin\phi sin\phi^{'}$, the constrain is written using inner products and outer products as follows:

$$\{e \cdot (p_1 \times p_2)\}\{(e^{'} \times p_1^{'}) \cdot (e^{'} \times p_2^{'})\}$$
$$-\{e^{'} \cdot (p_1^{'} \times p_2^{'})\}\{(e \times p_1) \cdot (e \times p_2)\} = 0 \quad (65)$$

Since the all terms in equation (65) are invariant with rotation, the equation holds for any general position of p_1, p_2, $p_1^{'}$ and $p_2^{'}$ even though the equation is derived from the special position. Moreover, even thought p_1 and $p_1^{'}$ are given instead of giving the rotation angle along the line going through both camera centers, the p_1 and p_2 are exchangeable.

The equation (65) is rewritten as follows:

$$\frac{(e \times p_1) \cdot (e \times p_2)}{e \cdot (p_1 \times p_2)} = \frac{(e^{'} \times p_1^{'}) \cdot (e^{'} \times p_2^{'})}{e^{'} \cdot (p_1^{'} \times p_2^{'})} \tag{66}$$

In this expression, the left term is a value measured only in one coordinate system, and the right term is a value measured only in the other coordinate system.

2.4 Another Derived Constrain

The epipolar constrain for feature point P_1 means that the line $OO^{'}$ and the light rays Op_1 and $O^{'}p_1^{'}$ in the Fig.12 are on the same plane, so-called epipolar plane. For another feature point P_2, there exist another epipolar plane. Therefore, the angle between the two epipolar planes measured in one

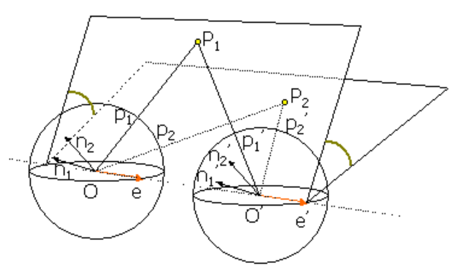

Figure 12: Angles between Epipolar Planes

coordinate system must be same as the angle between the two epipolar planes measured in the other coordinate system. The normal vectors of the two epipolar planes measured in each coordinate system are written respectively as follows:

$$n_1 = \frac{e \times p_1}{|e \times p_1|} \tag{67}$$

$$n_2 = \frac{e \times p_2}{|e \times p_2|} \tag{68}$$

$$n_1^{'} = \frac{e^{'} \times p_1^{'}}{|e^{'} \times p_1^{'}|} \tag{69}$$

$$n_2^{'} = \frac{e^{'} \times p_2^{'}}{|e^{'} \times p_2^{'}|} \tag{70}$$

Since the angles between the epipolar planes measured in each coordinate system must be same, the following equation is derived.

$$n_1 \cdot n_2 = n_1^{'} \cdot n_2^{'} \tag{71}$$

In general, the following equations hold with regard to three-dimensional unit column vectors a, b and c:

$$(a \times b) \cdot (a \times c) = |a|^2 (b \cdot c) - (a \cdot b)(a \cdot c) \tag{72}$$

$$= b^T \left(I - aa^T\right) c \tag{73}$$

$$|a \times b| = \sqrt{1 - (a \cdot b)^2} \tag{74}$$

$$= \sqrt{b^T \left(I - aa^T\right) b} \tag{75}$$

Here, I is 3×3 identical matrix.

Q and $Q^{'}$ are defined as follows:

$$Q = 1 - ee^T \tag{76}$$

$$Q^{'} = 1 - e^{'} e^{'T} \tag{77}$$

P and $P^{'}$ are defined as follows:

$$P = \begin{pmatrix} p_1 & ... & p_i & ... & p_n \end{pmatrix}^T \tag{78}$$

$$P^{'} = \begin{pmatrix} p_1^{'} & ... & p_i^{'} & ... & p_n^{'} \end{pmatrix}^T \tag{79}$$

Here, p_i is ith feature point in one coordinate system, and $p_i^{'}$ is its corresponding point in the other coordinate system. C and $C^{'}$ are defined as follows:

$$C = P^T \cdot Q \cdot P \tag{80}$$

$$C^{'} = P^{'T} \cdot Q^{'} \cdot P^{'} \tag{81}$$

Using the formulas (74) and (75), the constrain (71) is rewritten with regard with ith and jth feature points as follows:

$$\frac{C_{ij}}{\sqrt{C_{ii}C_{jj}}} = \frac{C_{ij}^{'}}{\sqrt{C_{ii}^{'}C_{jj}^{'}}} \tag{82}$$

Eventually, each left and right term of the constrain (71) is measured in the matrix C and $C^{'}$ respectively. C is simply obtained by multiplication between the measurement matrix P and the motion parameters matrix Q. C' is also obtained similarly.

The equations (66) and (71) are written under the definition (57)-(61) respectively as follows:

$$\frac{sin\phi cos\phi_2 - cos\phi cos\theta sin\phi_2}{sin\theta sin\phi_2} = \frac{sin\phi^{'} cos\phi_2^{'} - cos\phi^{'} cos\theta^{'} sin\phi_2^{'}}{sin\theta^{'} sin\phi_2^{'}} \tag{83}$$

$$\frac{sin\phi cos\phi_2 - cos\phi cos\theta sin\phi_2}{\sqrt{1-(sin\phi cos\theta sin\phi_2 + cos\phi cos\phi_2)^2}}$$
$$= \frac{sin\phi' cos\phi_2' - cos\phi' cos\theta' sin\phi_2'}{\sqrt{1-\left(sin\phi' cos\theta' sin\phi_2' + cos\phi' cos\phi_2'\right)^2}} \quad (84)$$

The denominator of equation (84) is rewritten as follows:

$$\begin{aligned}
&\sqrt{1-(sin\phi cos\theta sin\phi_2 + cos\phi cos\phi_2)^2} \\
&= \big(1 - sin^2\phi_2 sin^2\phi cos^2\theta - 2sin\phi_2 cos\phi_2 sin\phi cos\phi cos\theta cos\theta \\
&\quad -cos^2\phi_2 + cos^2\phi_2 sin^2\phi\big)^{\frac{1}{2}} \\
&= \big(1 - sin^2\phi_2 cos^2\theta + sin^2\phi_2 cos^2\phi cos^2\theta \\
&\quad -2sin\phi_2 cos\phi_2 sin\phi cos\phi cos\theta cos\theta - cos^2\phi_2 + cos^2\phi_2 sin^2\phi\big)^{\frac{1}{2}} \\
&= \big\{1 - sin^2\phi_2 cos^2\theta - cos^2\phi_2 \\
&\quad + (sin\phi_2 cos\phi cos\theta - cos\phi_2 sin\phi)^2\big\}^{\frac{1}{2}} \\
&= \big\{1 - cos^2\phi_2 - sin^2\phi_2 + sin^2\phi_2 sin^2\theta \\
&\quad + (sin\phi_2 cos\phi cos\theta - cos\phi_2 sin\phi)^2\big\}^{\frac{1}{2}} \\
&= sin\phi_2 sin\theta \sqrt{1 + \frac{sin\phi_2 cos\phi cos\theta - cos\phi_2 sin\phi}{sin\phi_2 sin\theta}} \qquad (85)
\end{aligned}$$

Here, $sin\phi_2 sin\theta$ is assumed to be positive. Therefore, suppose s and s' is defined as left and right term of the equation (83) respectively, the equation (84) is rewritten as follows:

$$\frac{s}{\sqrt{1-s^2}} = \frac{s'}{\sqrt{1-s'^2}} \qquad (86)$$

In general, suppose x is defined as $cot\alpha$, $cos\alpha$ is expressed by $\frac{x}{\sqrt{1-x^2}}$. Since each left and right term of the equation (71) is the dot product of normal vectors on epipolar planes, each term is written as $cos\alpha$ and $cos\alpha^{'}$ respectively. The α and $\alpha^{'}$ are the angles between two epipolar planes measured in each coordinate system. Eventually, two derived constrains (66) and (71) are rewritten using the angles α and $\alpha^{'}$ respectively as follows:

$$cot\alpha = cot\alpha^{'} \tag{87}$$

$$cos\alpha = cos\alpha^{'} \tag{88}$$

Therefore, both constrains mean the agreement about the angles between epipolar planes measured in each coordinate system. While the original epipolar constrain is comparing the epipolar line of a feature point and its corresponding point, the derived constrain is comparing between angles. In other words, the original epipolar constrain is comparing different type of things, a line and a point, whereas the derived constrain is comparing same type of things, angles.

2.5 Constrain in One Coordinate System

Since the derived constrain (66) is separated in each coordinate system, estimation of the motion parameter, which is same as the epipole estimation, in only one coordinate system can be examined assuming the epipole in the other coordinate system is estimated. Suppose the right term of the constrain is constant C.

$$\frac{(e \times p_1) \cdot (e \times p_2)}{e \cdot (p_1 \times p_2)} = C \tag{89}$$

Then, a function F of e is defined as follows:

$$F(e) = (e \times p_1) \cdot (e \times p_2) - C \cdot \{e \cdot (p_1 \times p_2)\} \tag{90}$$

The e which satisfy $F(e) = 0$ satisfy the constrain (89). As examples, values of function $F(e)$ are color coded on a sphere in Fig.13. The black solid curves

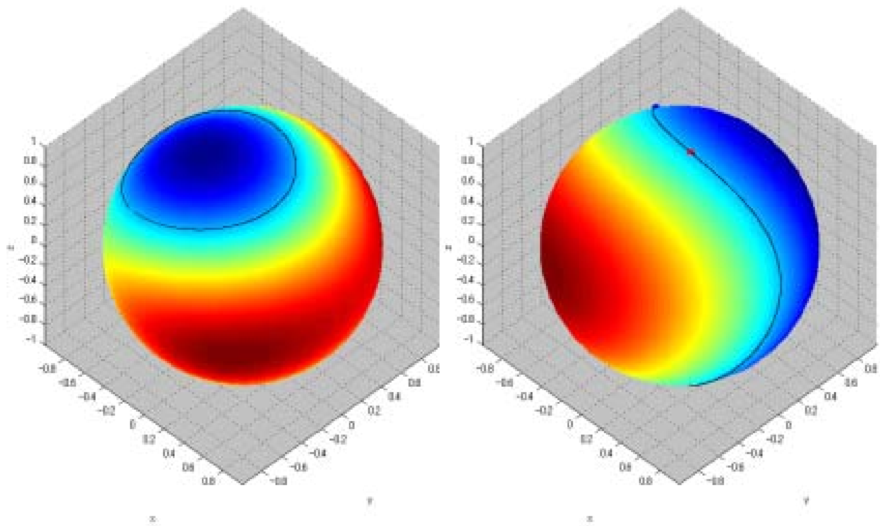

Figure 13: Examples of $F(e)$

are the curves on which the function $F(e)$ is zero.

The function $F(e)$ has very simple shape. There are only one or two zero-curves. The zero-curves are closed. The zero-curves do not intersect each other. Therefore, one side of a zero-curve is negative and the other side is positive. By the zero-curves, the imaging surface is divided into two or three areas.

When more than two feature points are given, the zero-curves are drawn for each two pair of feature points. Fig.14 shows an example of simulated corresponding points. Corresponding points measured in each coordinate system are plotted in one figure in Fig.14. The same color points are corresponding points each other. Then, zero-curves are drawn as in Fig.15 Since

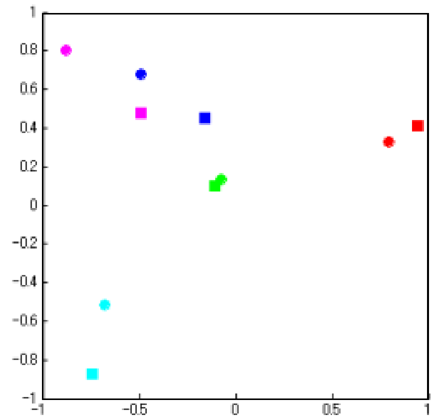

Figure 14: Corresponding Points

each zero-curve are points that satisfy the equation (89), the intersection of zero-curves for all the corresponding pairs is the point that satisfy all constrains. In Fig.15, the sum of square $F(e)$ is also color-coded. It shows that the global minimum is at the intersection.

When the C in the equation (89) is given with correct $e^{'}$, zero-curves for all correspondent point pairs intersect at one point on the sphere (Fig.15); however, zero-curves do not intersect coincidently when the C is given with wrong $e^{'}$ (Fig.16). In the Fig.16, the $e^{'}$ is 10 deg different from correct epipole. Therefore, zero-curves do not intersect at one point in Fig.16. The sum of square $F(e)$ is also shown in color code on the sphere.

2.6 Minimization

By multiplying the constrain (66) by both denominator and moving right

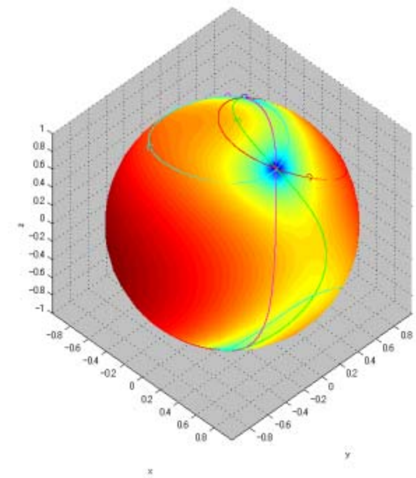

Figure 15: Intersection of Zero-curves

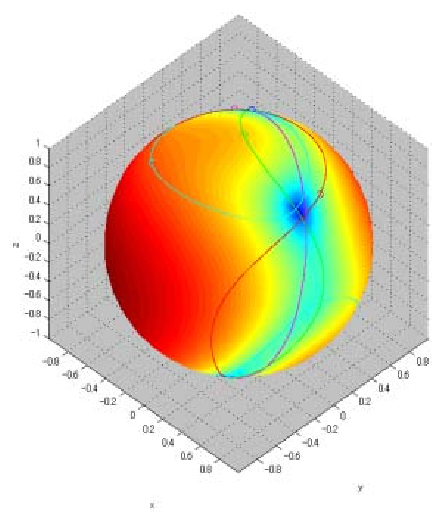

Figure 16: Intersection of Zero-curves with Wrong $e^{'}$

term to left, A function $F_{ij}\left(e, e^{'}\right)$ is defined as follows:

$$F_{ij}\left(e, e^{'}\right) = e^{'} \cdot (p_i^{'} \times p_j^{'}) \cdot (e \times p_i) \cdot (e \times p_j) - e \cdot (p_i \times p_j) \cdot (e^{'} \times p_i^{'}) \cdot (e^{'} \times p_j^{'}) \quad (91)$$

When e and $e^{'}$ satisfy the constrain (66), the function $F_{ij}\left(e, e^{'}\right)$ is zero. Therefore, epipoles can be estimated by minimizing sum of square $F_{ij}\left(e, e^{'}\right)$.

$$\min_{e, e^{'}} \sum_{i,j}^{n} \left\{ F_{ij}\left(e, e^{'}\right) \right\}^2 \quad (92)$$

However, the nonlinear minimization result depends on the initial value. Fig.17-a and Fig.17-b show two results by Levenberg-Marquardt method, and in both case, estimated epipoles are different from correct epipoles. The small red circles indicate the estimated epipole in one coordinate system. In

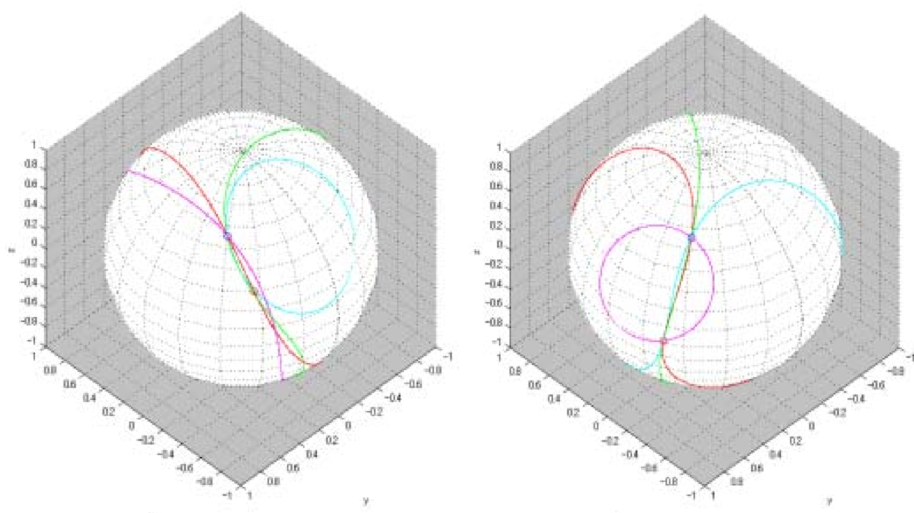

Figure 17: a)Local Minimum and b)Another Solution

Fig.17-a, since the zero-curves do not intersect at the estimated epipole, the

estimated epipole do not satisfy constrains. As well as Fig.16, even though all zero-curves do not intersect at one point, there still exist minimum point on the sphere. It is considered that those minimum points can be local minimum. On the other hand, in Fig.17-b, although the estimated epipoles are different from the correct epipoles, there exist coincident intersection at the estimated epipole. It is understood that these epipoles are another solution because the coincident intersection means the point satisfy all constrains. In other word, when fewer image feature points are used, it is possible that there is more than one solution.

2.7 Voting Algorithm

In order to estimate epipoles, there are some requirements to an algorithm based on the analysis until previous section. First, the algorithm should find the intersections of zero-curves directly. Looking for epipoles in the area where no zero-curves go through is not only waste, but also risk of local minimum. Second, since one unique solution is not guaranteed, the algorithm should find all the possible solutions. In general, minimization algorithms are used to find unique global minimum. Therefore, minimization algorithms do not work for this problem. Third, the algorithm should detect no solution or ignore outliers. It is assumed there are some independently moving objects in the scene. When the feature points are extracted from background and those objects, zero-curves of the independently moving feature points do not intersect at the epipole. Fourth, the algorithm should not demand initial estimation.

A voting algorithm is suited in order to satisfy those requirements. The idea is the following. A unit sphere surface in each coordinate system is divided into small patches. In one coordinate system, a point is chosen for each patch, such as a corner of a patch. For each patch in one coordinate system, the chosen point is a tentative epipole. For each tentative epipole in one coordinate system, zero-curves are drawn on a unit sphere in the other coordinate system. All the patches on which a zero-curve passes over a patch gain one vote. The patches gain one vote for one zero-curve. Since when correct epipole is given in one coordinate system, all the zero-curves in the other coordinate system intersect at one point, the patch on which the intersection exist gains maximum vote. The voting is taken place for all tentative epipoles. Fig.18 shows two examples of voting result. The vote scores are color-coded. Fig.18-a shows the voting result when the tentative epipole is close to the correct epipole. Fig.18-b shows the voting result when the tentative epipole is 10 deg different from the correct epipole. For the examples, spheres surface are divided into each $\frac{1}{40}$ longitude and $\frac{1}{40}$ latitude patch, and the tentative epipole is the top left corner of the patch. Since when the tentative epipole is close to the correct epipole, the patch on which all zero-curves intersect gain more votes than any patch of the case when the tentative epipole is far from the correct epipole. Therefore, the patch that gains maximum votes contains the epipole, and the tentative epipole in the other coordinate system is close to the other correct epipole.

2.8 Algorithm Implementation

The zero-curves are obtained as a set of points. Suppose p_1 and p_2 are the consecutive two points of a zero-curve, the normal vector of the great

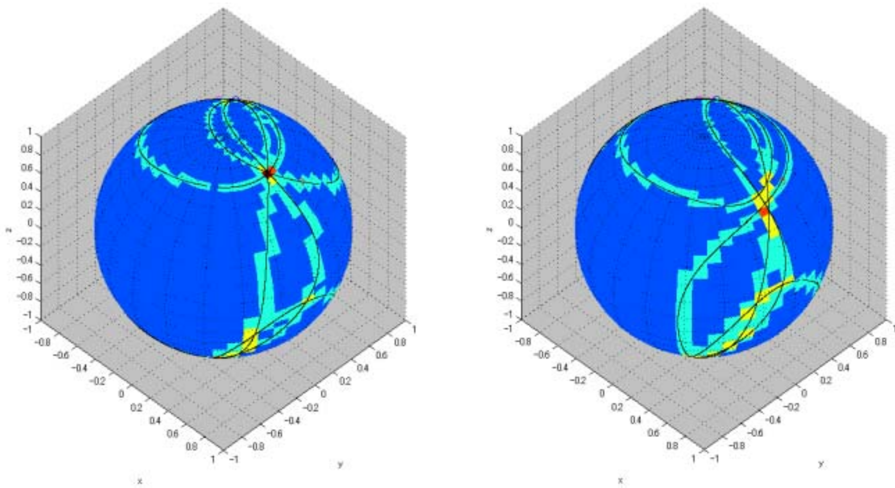

Figure 18: a)Voting with correct epipole and b)Voting with wrong epipole

circle passing through p_1 and p_2 is obtained as follows:

$$n = p_1 \times p_2 \tag{93}$$

Since a patch is small, a boundary of a patch is represented by a part of a great circle. Suppose $n^{'}$ is the normal vector of the great circle, the intersection p between the great circle passing p_1 and p_2 and the great circle of the patch boundary is obtained as follows:

$$p = n \times n^{'} \tag{94}$$

In case the sphere is divided into patches by longitude and latitude, a longitudinal boundary is expressed as follows:

$$n^{'} = \begin{pmatrix} cos\theta & sin\theta & 0 \end{pmatrix}^T \tag{95}$$

Fig.19 shows the procedure. The white curve is the zero-curve, and black line lattices are boundaries of patches. After finding all intersections between

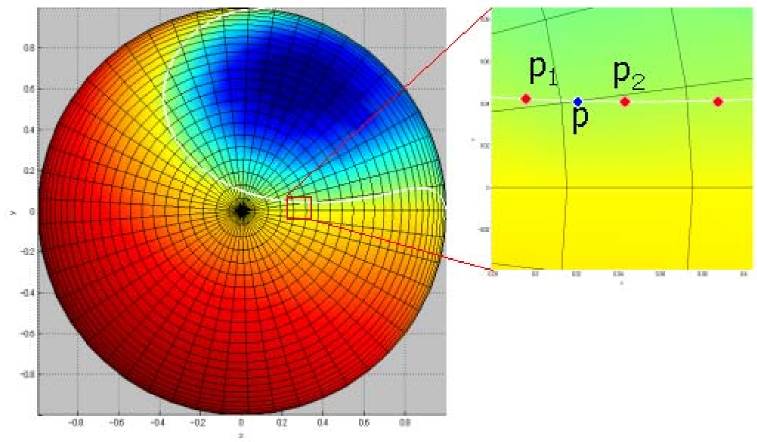

Figure 19: Intersection between Patch Boundary and Zero-curve

a zero-curve and patch boundaries, one vote is given to patches between the intersections. Same procedure is taken place for other zero-curves.

As another implementation of voting algorithm is following. It is assumed that a patch is represented by a polygon. A sign of $F(e)$ for each apex is calculated regarding with a zero-curve. If all signs of apexes of a patch are same, it is regarded that the zero-curve does not pass through the patch, because the $F(e)$ is positive in one side of zero-curve and negative in the other side of zero-curve.

$$n^{'} = \begin{pmatrix} cos\theta & sin\theta & 0 \end{pmatrix}^T \tag{96}$$

Fig.19 shows the procedure. The white curve is the zero-curve, and black line lattices are boundaries of patches.

Eventually, a zero-curve is represented by patches with one vote like in Fig.21.

Since the vote is taken placed on the sphere in one coordinate system for

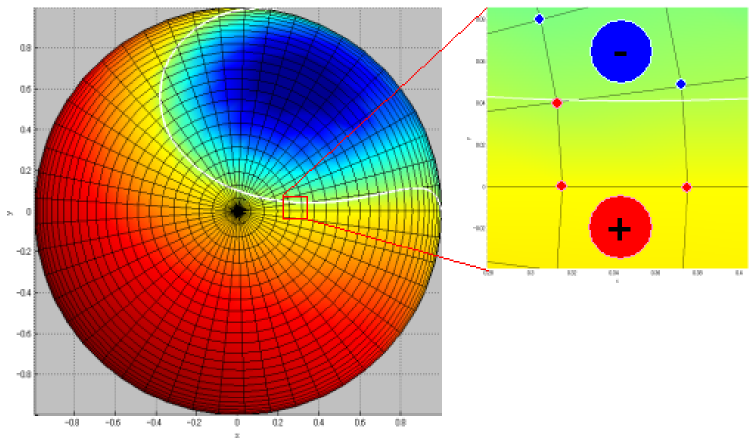

Figure 20: Finding Patches On Which Zero-curve Pass Through

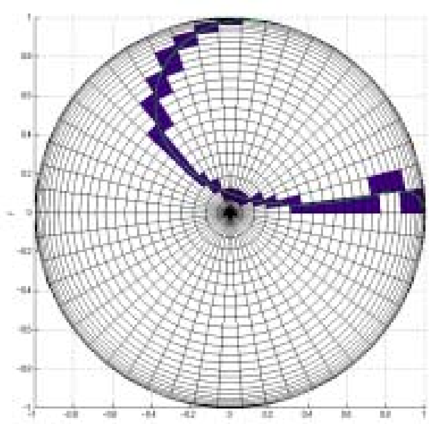

Figure 21: Voting Result for a Zero-curve

each tentative epipole in the other coordinate system, the voting table is four dimensional of two unit vectors. Although the voting table is shown in two dimensions in Fig.22, each column represents the vote score on a spherical surface. A column of the voting table is shown as a spherical surface in Fig.23.

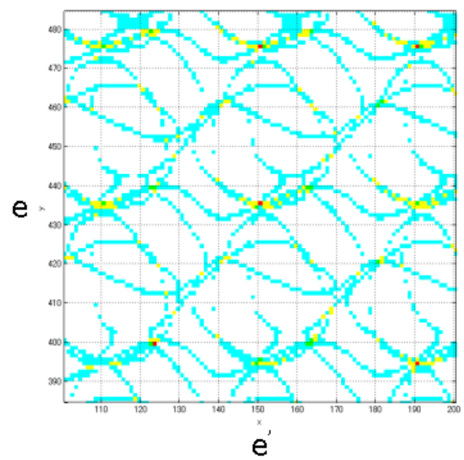

Figure 22: Voting Table

2.9 Algorithm Results

When number of feature points is small, there are some possible solutions. Fig24 shows an example result when there are five feature points. It shows patches that gain maximum vote in green. The red spot is the correct epipole. Although the patch on which the correct epipole exist gain maximum vote, some other patches also gain same amount of vote. As the number of feature points increasing, number of possible solution decrease. Fig25 shows an example result when there are ten feature points. Even though the number

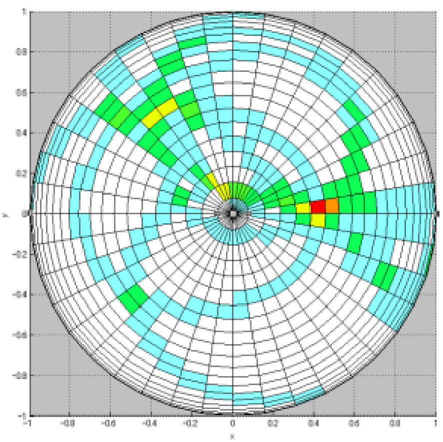

Figure 23: A Column of the Voting Table

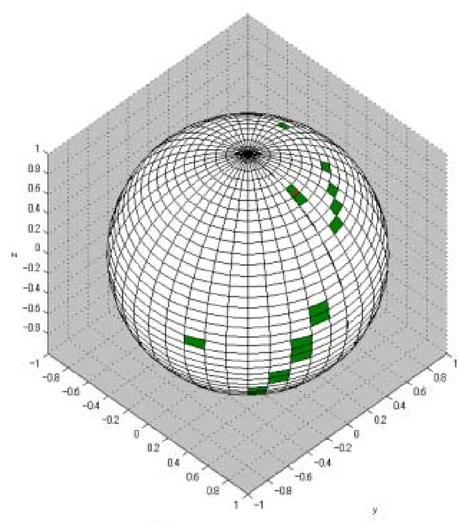

Figure 24: An Example Result of Five Feature Points

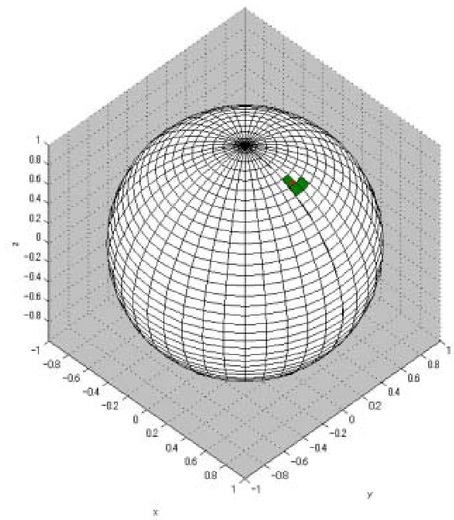

Figure 25: An Example Result of Ten Feature Points

of solution decreases, they contain the correct epipole.

When the motion is small, in other word, the distance between two camera centers is small; there can be many possible solutions. It is considered that the ambiguity of motion increase, as the distance between two camera centers decrease. Fig.26 shows an example result that is obtained when the distance between two camera centers is 0.02. Since both cameras are normalized, the distance is normalized too. To compare with Fig.25 in which the distance is 2.2, possible solutions increased and become broad. It is considered that the area on which the possible solutions are spread is the ambiguity of motion, so called valley.

When a few outliers are involved in a set of feature points, the voting algorithm can ignore the outliers. Corresponding point mismatching, independently moving objects, etc produce the outlier feature points. When the

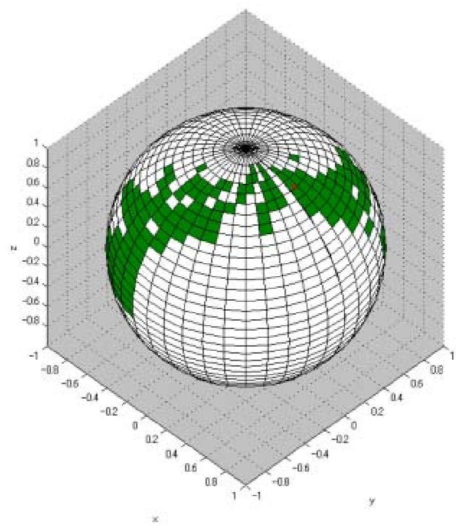

Figure 26: An Example Result of Small Motion

tentative epipole is the correct epipole, all zero-curves are intersecting at one point except zero-curves of outlier. As long as there is no another motion perception involving outliers, the patch on which the epipole exist gain the maximum vote. Fig.27 shows an example of zero-curves of feature points including outliers in one coordinate system in case that the correct epipole in the other coordinate system is given. The zero-curves in green and blue are outliers; therefore, those curves do not intersect the point where other zero-curves intersect. Fig.28 shows the result of voting regarding with the same situation as Fig.27. It is shown in Fig.27 that even though a set of feature points involves outliers, the voting algorithm can ignore outliers. However, it is possible that motion of feature points including outliers can be percept as another motion of no outliers, especially when the number of feature points is small. In that case, the voting score is the maximum at the epipole for the

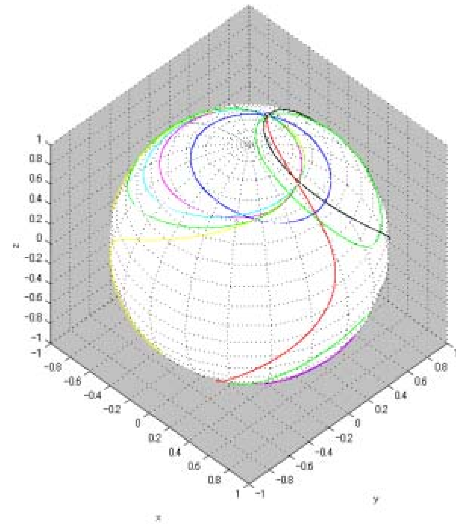

Figure 27: Zero-curves of Outliers

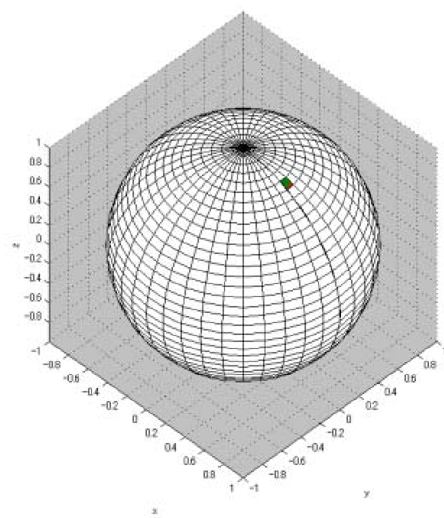

Figure 28: An Example Result of Outliers

solution.

The voting algorithm has to have a certain size patches; therefore, a patch on which all zero-curves pass through can gain maximum vote score, even though zero-curves do not intersect at one point. If the patch size is enough small, the patch can be the approximation of the intersection. If the patch size is not enough small, some patches that gain maximum vote are not solutions due to the size of patch. In order to eliminate such patches and to find all solutions, minimization method, equation (92), can be utilized when the patch size is not enough small. The patches that gain maximum vote are used to give an initial value for minimization method. Then the minimum point found by the minimization method is still on the patch, the minimum point is considered as a solution. Fig.29 shows the result of six feature points. The yellow points are solutions found by the combination of the voting algorithm and the minimization method.

2.10 Conclusions

Two epipoles are used as motion parameters instead of a translation and a rotation to determine the ego-motion. The epipolar constrain is written with epipoles. Since two epipoles have only four parameters, another pair of a feature point is used in order to determine the angle along the line passing through both camera centers. Then, the derived constrain involves two pair of corresponding points, while original one involves one pair of corresponding points. The original epipolar constrain tells that a feature point measured in one coordinate system has to be on the epipolar line of a feature point measured in the other coordinate system. Briefly, the original epipolar con-

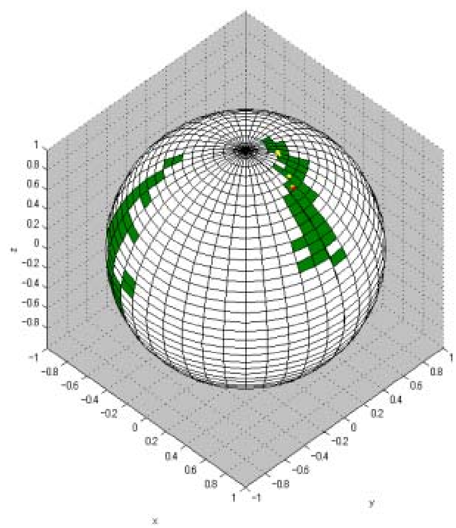

Figure 29: An Example Result of Voting and Minimization Combination

strain compare different types of things, a line and a point. On the other hand, the derived constrain is equivalent to the agreement of angles between epipolar planes measured in each coordinate system. The derived constrain compare two angles. Moreover, on one hand the original constrain involve deferent types of motion parameters, which are rotations and directions; on the other hand, the derived constrain involve two direction in motion parameters. Therefore, the derived constrain is more even to two coordinate systems.

Since the derived epipole is balanced to each coordinate system, the constrain is separated into two values in same expression measured in each coordinate system. The value is cotangent of an angle between epipolar planes. Since the constrain is separated in each coordinate system, the constrain can be examined in one coordinate system by assuming that the values measured in the other coordinate system is fixed. Moreover, since the constrain involve only four motion parameters, an epipole in each coordinate system, the constrain is visualized on a sphere as zero-curves of function $F(e)$. Then, the motion estimation is recognized as finding a coincident intersection of the zero-curves.

A voting algorithm is introduced to find the coincident intersection. Since one unique solution is not guaranteed, it is necessary to find all the solutions. Those solutions are considered as different perceptions of the motion. In addition, outliers have to be disregarded because independently moving objects are assumed in the scene. The voting algorithm satisfies those requirements. The qualitative results are shown in regard to the number of feature points, the size of motion and outliers. Although it shows positive results, combination of voting and minimization is necessary due to the patch size, when the patch size is not enough small.

Part II

Appendix

A Single Camera Calibration

A.1 Fundamentals

Calibration is the process to find the some of camera intrinsic parameters and extrinsic parameters or all of them. Intrinsic parameters are the parameters that depend on the camera itself, such as the focal length, the principal point, and pixel size. Extrinsic parameters are the camera position and direction. When the camera coordinate (x_c, y_c, z_c) are placed in the global coordinate (x_w, y_w, z_w), the transform from the global coordinate to the camera coordinate is as follows [1]:

$$M_c = (R|\mathbf{t}) \cdot M_w = (R| - RT) \cdot M_w \tag{97}$$

$$M_c = \begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} \quad M_w = \begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix} \tag{98}$$

Here, M_w is the global coordinate expression of a point, and M_c is the camera coordinate expression of the point. The $\mathbf{t}$ is the vector to the global coordinate origin in the camera coordinate system. The T is the vector to the camera coordinate origin in the global coordinate system. The $\mathbf{R}$ is the rotation matrix which rotate the camera coordinate axes to be same as the global coordinate axes.

The transform from the camera coordinate (x_c, y_c, z_c) to the image coordinate (u_p, v_p) is as follows:

$$wm_p = A_p \cdot M_c \tag{99}$$

$$m_p = \begin{pmatrix} u_p \\ v_p \\ 1 \end{pmatrix} \tag{100}$$

The image coordinate (u_p, v_p) is the coordinate which indicate a point in the projecting plane. The m_p express the point in which a point in camera coordinate is projected. Here, w is a scale factor. The point in the image coordinate can be transformed to the pixel coordinate (u_q, v_q) as follows:

$$m_q = A_q \cdot m_p \tag{101}$$

$$m_p = \begin{pmatrix} u_q \\ v_q \\ 1 \end{pmatrix} \tag{102}$$

$A = A_q \cdot A_p$ is the matrix which express the projection between camera coordinate and pixel coordinate. The **Projection Matrix** P is defined as follows:

$$P = A \cdot (R|t) = A \cdot (R| - RT) \tag{103}$$

The relationship of a point between the global coordinate and pixel coordinate is expressed follows:

$$wm_q = P \cdot M_w \tag{104}$$

When the camera is described as the pinhole model the A_p is expressed as follows:

$$A_p = \begin{pmatrix} f & f/\cot\phi & 0 \\ 0 & f/\sin\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{105}$$

The A_q is expressed as follows:

$$A_q = \begin{pmatrix} 1/d_u & 0 & u_0 \\ 0 & 1/d_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{106}$$

Therefore, the A is as follows:

$$A = \begin{pmatrix} f/d_u & f/(d_u \cot\phi) & u_0 \\ 0 & f/(d_v \sin\phi) & v_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{107}$$

Here, the f is the focal length, d_u、d_v are the pixel width and height, u_0, v_0 is the principal point. d_u、d_v generally disagree with the imaging element size because of the D/A and A/D transform. The ϕ is the angle between axes in the image coordinate, and when ϕ is 90°, $\cot\phi = 0$, $\sin\phi = 1$ are hold, then the A_p is as follows:

$$A_p = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{108}$$

Therefore the A is expressed as follows:

$$A = \begin{pmatrix} f/d_u & 0 & u_0 \\ 0 & f/d_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \alpha_x & 0 & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{109}$$

The pinhole model is applied and the ϕ is supposed to be 90° bellow. Then, the transform from camera coordinate to the image coordinate in the equation (99) is called as **Center Projection** and expressed as follows:

$$wm_p = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} M_c \tag{110}$$

The transform to the pixel coordinate in the equation (101) is expressed as follows:

$$m_q = \begin{pmatrix} 1/d_u & 0 & u_0 \\ 0 & 1/d_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} m_p \tag{111}$$

The lens aberration sometimes cannot be neglected. Whereas a point supposed to be projected to (u_p, v_p) in the image coordinate by the projection equation (110) without any aberration, the point is projected to (u_d, v_d) due to the aberration. The difference between those projected points with aberration and without aberration is defined as the model of aberration. Therefore, the differences are expressed as follows:

$$\begin{pmatrix} u_d \\ v_d \end{pmatrix} = \begin{pmatrix} u_p \\ v_p \end{pmatrix} + \begin{pmatrix} \delta u \\ \delta v \end{pmatrix} \tag{112}$$

or

$$\begin{pmatrix} u_p \\ v_p \end{pmatrix} = \begin{pmatrix} u_d \\ v_d \end{pmatrix} + \begin{pmatrix} \delta u' \\ \delta v' \end{pmatrix} \tag{113}$$

Then, $(\delta u, \delta v)$ or $(\delta u', \delta v')$ is defined with some aberration parameters. The (u_p, v_p) is called as the image coordinate without aberration, and the (u_d, v_d) is called as image coordinate with aberration bellow in this report.

The lens aberration model by Weng et al. is the most common aberration model. It express the change $(\delta u, \delta v)$ by the aberration using some parameters as follows:

$$\begin{aligned}
\delta u &= \kappa_1 u_p \left(u_p^2 + v_p^2\right) \\
&+ p_1 \left(3u_p^2 + v_p^2\right) + 2p_2 u_p v_p \\
&+ s_1 \left(u_p^2 + v_p^2\right) \qquad (114) \\
\delta v &= \kappa_1 v_p \left(u_p^2 + v_p^2\right) \\
&+ 2p_1 u_p v_p + p_2 \left(u_p^2 + 3v_p^2\right) \\
&+ s_1 \left(u_p^2 + v_p^2\right) \qquad (115)
\end{aligned}$$

When only the radial distortion is considered, the aberration is described as follows:

$$\begin{aligned}
\begin{pmatrix} \delta u \\ \delta v \end{pmatrix} &= \delta(r) \begin{pmatrix} u_p \\ v_p \end{pmatrix} \qquad (116) \\
\delta(r) &= \kappa_1 r^2 + \kappa_2 r^4 + \cdots \qquad (117) \\
r &= \sqrt{u_p^2 + v_p^2}
\end{aligned}$$

It is also described by the aerated point (u_d, v_d) as follows:

$$\begin{aligned}
\begin{pmatrix} \delta u' \\ \delta v' \end{pmatrix} &= \delta(r') \begin{pmatrix} u_d \\ v_d \end{pmatrix} \qquad (118) \\
\delta'(r') &= \kappa_1' r'^2 + \kappa_2' r'^4 + \cdots \qquad (119) \\
r' &= \sqrt{u_d^2 + v_d^2}
\end{aligned}$$

m_d is defined as follows:

$$m_d = \begin{pmatrix} u_d \\ v_d \\ 1 \end{pmatrix} \qquad (120)$$

Then, the transform from image coordinate to the pixel coordinate in the equation (111) is expressed as follows:

$$m_q = \begin{pmatrix} 1/d_u & 0 & u_0 \\ 0 & 1/d_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} m_d \tag{121}$$

The aerated pixel coordinate is obtained by the equation above.

The purpose of a single camera calibration is to obtain the matrix P, A, R or t. Some algorithms obtain the optimal matrices to the observed data neglecting the relationship between elements of the matrices and the physical parameters; others obtain the physical parameters such as focal length and so on. The calibration process obtain some of the extrinsic parameters, R and t, and intrinsic parameters, f, d_u, d_v, （α_x、α_y）, u_0 , v_0, κ_1 or all of them. Then, it finds the matrix P, or A. In the case of obtaining not all parameters, the rest parameters are supposed to be known.

A.2 Tsai's Algorithm

Tsai's Algorithm [15] using coplanar points can obtain the extrinsic parameters, R, t and the intrinsic parameters, f, κ_1. Other parameters, d_u, d_v, u_0, v_0, are supposed to be known. In addition, the aberration is supposed to occurred due to only the radial distortion, and it is approximated by the first term of the equation (117).

A calibration plane on which lattice pattern is drawn like chessboard is placed to be that the plane is not parallel to the imaging plane. The origin of the global coordinate is places on the plane. The z_w axis of the global coordinate is placed as it is parallel to the normal vector of the calibration plane, so that the $z_w = 0$ is hold on the calibration plane In addition, the origin of the global coordinate is placed not close to the $x_c z_c$ plane to make $ty \neq 0$ hold.

Suppose that n pairs of the global coordinate of the grids $(x_{wi}, y_{wi}, 0)$ and the projected points, (u_{qi}, v_{qi}), in the pixel coordinate is supposed to be obtained, the camera parameters are calculated as in bellow. Here, i, $i = 1, ..., n$ indicates each lattice.

A.2.1 Process

1. Transform from the pixel coordinate to the image coordinate with aberration

   From the equation (121), the transform from the pixel coordinate m_q to the image coordinate with aberration m_d is expressed as follows:

$$m_d = \begin{pmatrix} d_u & 0 & -d_u u_0 \\ 0 & d_v & -d_v v_0 \\ 0 & 0 & 1 \end{pmatrix} m_q \tag{122}$$

   Therefore, (u_{qi}, v_{qi}) is transformed to $(x_{di}, y_{di}, 1)$.

2. Extrinsic Parameters Calculation (1)

   Tsai noticed the restriction[1] in case that the aberration is occurred due to only radial distortion, and show the equation as follows [15]:

$$\frac{u_d}{v_d} = \frac{u_p}{v_p} = \frac{x_c}{y_c} \tag{123}$$

[1] Tsai calls as RAC(radial alignment constraint)

The first equation is derived from the fact that the (u_d, v_d) and (u_p, v_p) are proportional, which is proven by substituting the equation (120) for the equation (113). The second equation is derived from the equation (110)which express center projection. The $\mathbf{R}$, t_x and t_y is obtained from the equation above. The $(R|t)$ in the equation (97) which express the transform from the global coordinate to the camera coordinate is written in its elements as follows:

$$(R|t) = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{pmatrix} \tag{124}$$

Then, from the equation (97), the equation (123) is rewritten as follows:

$$\frac{u_d}{v_d} = \frac{r_{11}x_w + r_{12}y_w + r_{13}z_w + t_x}{r_{21}x_w + r_{22}y_w + r_{23}z_w + t_y} \tag{125}$$

Since points used in Tsai's algorithm are coplanar, the z_w axis of the global coordinate is defined to be parallel to the plane's normal vector, then always $z_w = 0$ is hold.

$$\frac{u_d}{v_d} = \frac{r_{11}x_w + r_{12}y_w + t_x}{r_{21}x_w + r_{22}y_w + t_y} \tag{126}$$

The following equation is derived from the equation (126):

$$v_d(r_{11}x_w + r_{12}y_w + t_x) - u_d(r_{21}x_w + r_{22}y_w + t_y) = 0$$

$$v_d x_w r_{11} + v_d y_w r_{12} + v_d t_x - u_d x_w r_{21} - u_d y_w r_{22} - u_d t_y = 0$$

$$v_d x_w r'_{11} + v_d y_w r'_{12} + v_d t'_x - u_d x_w r'_{21} - u_d y_w r'_{22} = u_d$$

$$\begin{pmatrix} v_d x_w & v_d y_w & v_d & -u_d x_w & -u_d y_w \end{pmatrix} \begin{pmatrix} r'_{11} \\ r'_{12} \\ t'_x \\ r'_{21} \\ r'_{22} \end{pmatrix} = u_d \qquad (127)$$

Here, the symbols above are defined as follows:

$$r'_{11} = \frac{r_{11}}{t_y} \quad r'_{12} = \frac{r_{12}}{t_y} \quad t'_x = \frac{t_x}{t_y} \quad r'_{21} = \frac{r_{21}}{t_y} \quad r'_{22} = \frac{r_{22}}{t_y} \qquad (128)$$

r'_{12}、t'_x、r'_{21}、r'_{22} which hold the equation above for all $(x_{wi}, y_{wi}, 0)$and (u_{di}, v_{di}) is obtained. Since the actual observed data contain errors, the parameters are calculated by solving the least square solution. The global coordinate must be placed to be $t_y \neq 0$ so that the equation (128) is available.

3. Extrinsic Parameters Calculation (2)

Since R is a rotation matrix, the norm of each column and row must be 1. Therefore, the rotation matrix is written as follows:

$$R =$$

$$\begin{pmatrix} r'_{11}t_y & r'_{12}t_y & \pm\sqrt{1-(r'^2_{11}+r'^2_{12})t_y^2} \\ r'_{21}t_y & r'_{22}t_y & \pm\sqrt{1-(r'^2_{21}+r'^2_{22})t_y^2} \\ \pm\sqrt{1-(r'^2_{11}+r'^2_{21})t_y^2} & \pm\sqrt{1-(r'^2_{12}+r'^2_{22})t_y^2} & \pm\sqrt{1-S_r t_y^2} \end{pmatrix} \tag{129}$$

Here the symbol is defined as follows:

$$S_r = r'^2_{11} + r'^2_{12} + r'^2_{21} + r'^2_{22} \tag{130}$$

In addition, the fist and second column vectors in R must be perpendicular. It is described as follows:

$$\begin{aligned} & r'_{11}t_y r'_{12}t_y + r'_{21}t_y r'_{22}t_y \pm \sqrt{1-(r'^2_{11}+r'^2_{12})t_y^2}\sqrt{1-(r'^2_{21}+r'^2_{22})t_y^2} = 0 \\ & (r'_{11}r'_{12}t_y^2 + r'_{21}r'_{22}t_y^2)^2 = \{1-(r'^2_{11}+r'^2_{12})t_y^2\}\{1-(r'^2_{21}+r'^2_{22})t_y^2\} \\ & (r'_{11}r'_{12} + r'_{21}r'_{22})^2 t_y^4 = 1 - S_y t_y^2 + (r'^2_{11}+r'^2_{21})(r'^2_{12}+r'^2_{22})t_y^4 \\ & (r'^2_{11}r'^2_{12} + r'^2_{21}r'^2_{22} + 2r'_{11}r'_{12}r'_{21}r'_{22})t_y^4 \\ & \quad = (r'^2_{11}r'^2_{12} + r'^2_{11}r'^2_{22} + r'^2_{21}r'^2_{12} + r'^2_{21}r'^2_{22})t_y^4 - S_y t_y^2 + 1 \\ & (r'^2_{11}r'^2_{22} + r'^2_{12}r'^2_{21})^2 t_y^4 - S_r t_y^2 + 1 = 0 \end{aligned} \tag{131}$$

Therefore, t_y^2 is calculated as follows:

$$t_y^2 = \frac{S_r - \sqrt{S_r^2 - 4(r'^2_{11}r'^2_{22} + r'^2_{12}r'^2_{21})^2}}{2(r'^2_{11}r'^2_{22} + r'^2_{12}r'^2_{21})^2} \tag{132}$$

Although t_y can be positive or negative value, it is assumed to be positive. Then, that is r_{11}、r_{12}、r_{21}、r_{22}、t_x can be obtained by the equation (128). The following values are calculated using a observed point, $(x_{wi}, y_{wi}, 0)$、(u_p, v_p) which is apart from the principal point.

$$x_i = r_{11}x_{wi} + r_{12}y_{wi} + t_x \quad y_i = r_{21}x_{wi} + r_{22}y_{wi} + t_y \tag{133}$$

Then, if the sign of x_i and u_p and the sign of y_i and v_p agree, the assumption of the sign of t_y is correct. If not, the assumption is wrong, so change the sign of t_y, then, r_{11}, r_{12}, r_{21}, r_{22}, t_x are recalculated.

r_{13}, r_{23}, r_{31}, r_{32} and r_{33} are obtained by the equation (129). First, under the assumption of $r_{13} \geq 0$, r_{13} is calculated as follows:

$$r_{13} = \sqrt{1 - r_{11}^2 - r_{12}^2} \tag{134}$$

The sign of r_{23} is known because the inner product between the first column and the second column vectors must be 0. r_{23} is calculated as follows:

$$r_{23} = \begin{cases} \sqrt{1 - r_{23}^2 - r_{22}^2} & \text{if } r_{11}r_{21} + r_{12}r_{22} \geq 0 \\ -\sqrt{1 - r_{23}^2 - r_{22}^2} & \text{if } r_{11}r_{21} + r_{12}r_{22} < 0 \end{cases} \tag{135}$$

Since the third column must be the outer product between the first and second column vector in R, r_{31}, r_{32} and r_{33} are calculated as follows:

$$r_{31} = r_{12}r_{23} - r_{13}r_{22} \quad r_{32} = r_{13}r_{21} - r_{11}r_{33} \quad r_{33} = r_{11}r_{22} - r_{12}r_{21} \tag{136}$$

4. Intrinsic Parameter Calculation (1)

First, f、t_z is calculated neglecting lens aberration. The elements of the camera coordinate is expressed from the equation (97) using $z_w = 0$ as follows:

$$\begin{aligned} y &= r_{21}x_w + r_{22}y_w + t_x \\ z &= r_{31}x_w + r_{32}y_w + t_x \end{aligned} \tag{137}$$

The equation (110) indicate $w = z$. Defining $h = r_{31}x_w + r_{32}y_w$, the following equation is hold.

$$\begin{aligned} (h + t_z)v_p &= fy \\ \begin{pmatrix} y & -v_p \end{pmatrix} \begin{pmatrix} f \\ t_z \end{pmatrix} &= hv_p \end{aligned} \tag{138}$$

Here, the aberration is neglected, (u_p, v_p) and (u_d, v_d) are identical. Using the (u_d, v_d) and $(x_{wi}, y_{wi}, 0)$ that are obtained by 1, the linear equation (138) can be solved to find f and t_z by the least square solution. Here, in order that the equation (138) is linear independent and f and t_z have unique solution, the calibration plane must not be parallel to the imaging plane. If $f < 0$, the assumption, $r_{13} \geq 0$, in 3 is not correct. Then, the signs of r_{13}、r_{23}、r_{31}、r_{32} are switched to be opposite, and f and t_z are recalculated.

5. Intrinsic Parameter Calculation (2)

When it is assumed that the lens aberration is approximated the first term of (117) and (119), the equation (118) and the equation (120) is rewritten by substituting them as follows:

$$\begin{pmatrix} u_d \\ v_d \end{pmatrix} = \begin{pmatrix} u_p \\ v_p \end{pmatrix} + \kappa_1 r^2 \begin{pmatrix} u_p \\ v_p \end{pmatrix} \tag{139}$$

$$\begin{pmatrix} u_p \\ v_p \end{pmatrix} = \begin{pmatrix} u_d \\ v_d \end{pmatrix} + \kappa_1' r'^2 \begin{pmatrix} u_d \\ v_d \end{pmatrix} \tag{140}$$

The following equation is derived from the equation (140), (110) and the equation (138).

$$(r_{31}x_w + r_{32}y_w + t_z)(1 + \kappa_1' r'^2)v_d = f(r_{21}x_w + r_{22}y_w + t_y) \tag{141}$$

Substituting the equation (122) for the equation above, it is expressed as follows:

$$\begin{aligned}(r_{31}x_w + r_{32}y_w + t_z)(1 + \kappa_1' r'^2)d_v(v - v_0) &= f(r_{21}x_w + r_{22}y_w + t_y) \\ d_v(v - v_0)(1 + \kappa_1' r'^2) &= f\frac{r_{21}x_w + r_{22}y_w + t_y}{(r_{31}x_w + r_{32}y_w + t_z)}\end{aligned} \tag{142}$$

f, t_z and κ_1' are obtained by solving the equation (142) as a nonlinear optimization problem with the initial values, f, t_z and $\kappa_1' = 0$ which are obtained by the process of 4.

A.2.2 Features

Tsai's algorithm is a classic camera calibration method based on the pinhole model. The algorithm is known as very realistic method because of its less computation power and the accuracy. The negative point of Tsai's algorithm is that the pixel size and the principal point have to be known.

A.2.3 Accuracy

As an example of the accuracy, a paper reported that after calculating f and κ_1', the average of the differences of the image points and calculated points by the matrix P is about 0.6 pixels in the image resolution 640x480 and focal length 4.2mm experiment. [8]. In addition, the paper [8] reports that when the given pixel size and the principal point have errors, the calculated parameters are affected. While the effect of the principal point error is small, and a few pixel errors can be eliminated by measurement repetition, the effect of the pixel size error cannot be eliminated by the repetition. The difference between the real pixel height and its design value of the device is usually small, but since the pixel width depends on the sampling timing, measuring the accurate pixel width is difficult.

The calibration plane is placed to be slanted to the imaging plane as described in the process 4. It is reported that the accuracy is better when it is slanted along the vertical axis than when it is slanted along the horizontal axis, because of the difference between pixel width and height.

A.3 Zhang's Algorithm

In Zhang's algorithm [8] [17], the equation (107) is rewritten as follows:

$$A = \begin{pmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{143}$$

The intrinsic parameter, α, β, γ, u_0 and v_0, and the extrinsic parameters, R, t can be obtained by the algorithm. The calibration plane with lattice pattern is used as well as Tsai's algorithm. Images of the plane are taken in some different positions and angles. In each imaging, the plane must not be parallel. The global coordinate is changed in each imaging, and the origin is placed to be on the calibration plane to hold $z_w = 0$ on the plane. In each imaging, the grid point in the global coordinate, $(x^i_{wj}, y^i_{wj}, 0)$, and the projected point in the pixel coordinate (u^i_j, v^i_j) is observed as the calibration data. Here, the upper letter indicates the imaging number, and the lower letter indicates the grid number.

A.3.1 Process

1. Finding the Homography $\mathbf{H}$

   The equation (103) and (103) are rewritten using $\mathbf{R} = \begin{pmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 \end{pmatrix}$, and $z_w = 0$ as follows:

$$w\mathbf{m} = \mathbf{A} \begin{pmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{t} \end{pmatrix} \begin{pmatrix} x_w \\ y_w \\ 0 \\ 1 \end{pmatrix}$$

$$= \mathbf{A} \begin{pmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{pmatrix} \begin{pmatrix} x_w \\ y_w \\ 0 \\ 1 \end{pmatrix}$$

$$= \mathbf{H} \begin{pmatrix} x_w \\ y_w \\ 1 \end{pmatrix} \tag{144}$$

   Here, $\mathbf{H} = \mathbf{A} \begin{pmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{pmatrix}$ is defined. In each imaging, the $\mathbf{H}$ is calculated from the observed data [17]. Since the multiplied H for the equation (144), the scale factor can not be found.

2. Without Aberration

i) Calculating Initial Value of Intrinsic and Extrinsic Parameters

The $\mathbf{H}$ which is obtained in the process 1 is redefined using the scale factor λ as follows:

$$\mathbf{H} = \begin{pmatrix} \mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3 \end{pmatrix} = \lambda \mathbf{A} \begin{pmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{pmatrix} \tag{145}$$

Since the $\mathbf{r}_1$ and $\mathbf{r}_2$ are perpendicular and identical size, the following equations are hold:

$$\begin{aligned} \mathbf{h}_1 \mathbf{A}^{-T} \mathbf{A}^{-1} \mathbf{h}_2 &= 0 && (146) \\ \mathbf{h}_1 \mathbf{A}^{-T} \mathbf{A}^{-1} \mathbf{h}_1 &= \mathbf{h}_2 \mathbf{A}^{-T} \mathbf{A}^{-1} \mathbf{h}_2 && (147) \end{aligned}$$

Here, The $\mathbf{A}^{-T}\mathbf{A}^{-1}$ can be rewritten using the parameters in the equation 143 as the follows ($\mathbf{A}^{-T}\mathbf{A}^{-1}$ is known as the projection of the absolute conic on the image plane [16]) :

$$\begin{aligned} \mathbf{B} &= \mathbf{A}^{-T}\mathbf{A}^{-1} \\ &= \begin{pmatrix} \frac{1}{\alpha^2} & -\frac{\gamma}{\alpha^2\beta} & \frac{\gamma v_0 - \beta u_0}{\alpha^2\beta} \\ -\frac{\gamma}{\alpha^2\beta} & \frac{\gamma^2}{\alpha^2\beta^2} + \frac{1}{\beta^2} & -\frac{\gamma(\gamma v_0 - \beta u_0)}{\alpha^2\beta^2} - \frac{v_0}{\beta^2} \\ \frac{\gamma v_0 - \beta u_0}{\alpha^2\beta} & -\frac{\gamma(\gamma v_0 - \beta u_0)}{\alpha^2\beta^2} - \frac{v_0}{\beta^2} & \frac{(\gamma v_0 - \beta u_0)^2}{\alpha^2\beta^2} + \frac{v_0^2}{\beta^2} + 1 \end{pmatrix} \\ &\equiv \begin{pmatrix} B_{11} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{pmatrix} && (148) \end{aligned}$$

Besides, the symbols are defined as follows:

$$\mathbf{b} = \begin{pmatrix} B_{11} & B_{12} & B_{22} & B_{13} & B_{23} & B_{33} \end{pmatrix}^T$$

$$\mathbf{h_i} = \begin{pmatrix} h_{i1} & h_{i2} & h_{i3} \end{pmatrix}^T$$

$$\mathbf{v}_{ij} = \begin{pmatrix} h_{i1}h_{j1} & h_{i1}h_{j2} + h_{i2}h_{j1} & h_{i2}h_{j2} & h_{i3}h_{j1} + h_{i1}h_{j3} \\ & & & h_{i3}h_{j2} + h_{i2}h_{j3} & h_{i3}h_{j3} \end{pmatrix}^T$$

Then, the following equation is hold:

$$\mathbf{h}_i^T \mathbf{B} \mathbf{h}_j = \mathbf{v}_{ij}^T \mathbf{b} \qquad (149)$$

In addition, the equation (146) and (147) are summarized as follows:

$$\mathbf{V}\mathbf{b} = \mathbf{0} \tag{150}$$

$$\mathbf{V} = \begin{pmatrix} \mathbf{v}_{12}^T \\ (\mathbf{v}_{11} - \mathbf{v}_{22})^T \end{pmatrix} \tag{151}$$

The equation (150) gives two fouler as the constrain of the unknown six- dimensional vector **b**. Since the equation (150) is given by each **H**, **b** can be solved by at least three **H**. The solution of the equation (150) is calculated as the eigenvector of the minimum eigenvalue of $\mathbf{V}^T\mathbf{V}$ or it is also obtained by finding zero-space using singular value decomposition [8]. Once the **b** is obtained the intrinsic parameters are obtained form the equation (148) as follows:

$$\begin{aligned}
v_0 &= (B_{12}B_{13} - B_{11}B_{23})/(B_{12}B_{22} - B_{12}^2) \\
\lambda &= B_{33} - \{B_{13}^2 + v_0(B_{12}B_{13} - B_{11}B_{23})/B_{11} \\
\alpha &= \sqrt{\lambda/B_{11}} \\
\beta &= \sqrt{\lambda B_{11}/(B_{12}B_{22} - B_{12}^2)} \\
\gamma &= -B_{12}\alpha^2\beta/\lambda \\
u_0 &= \gamma v_0/\alpha - B_{13}\alpha^2/\lambda
\end{aligned}$$

In addition, the extrinsic parameters are obtained from the equation (145) as follows:

$$\begin{aligned}
\mathbf{r}_1 &= \lambda\mathbf{A}^{-1}\mathbf{h}_1 \\
\mathbf{r}_2 &= \lambda\mathbf{A}^{-1}\mathbf{h}_1 \\
\mathbf{r}_3 &= \mathbf{r}_1 \times \mathbf{r}_2 \\
\mathbf{t} &= \lambda\mathbf{A}^{-1}\mathbf{h}_3
\end{aligned}$$

ii) Refine the Extrinsic Parameter R

The obtained $\mathbf{R}$ above is not generally a rotation matrix because of errors. A rotation matrix $\hat{\mathbf{R}}$ which has minimum Frobenius norm of $\hat{\mathbf{R}} - \mathbf{R}$ is calculated bellow. The problem is define the formulae as follows:

$$\min_{\hat{\mathbf{R}}} ||\hat{\mathbf{R}} - \mathbf{R}||^2 \tag{152}$$

The $\hat{\mathbf{R}}$ holds for the following equation.

$$\mathbf{R}^T \mathbf{R} = \mathbf{I} \tag{153}$$

Since the norm is expressed as flows, the minimization in the equation (152) is equal to maximization of $trace(\hat{\mathbf{R}}^T \mathbf{R})$.

$$\begin{aligned} ||\hat{\mathbf{R}} - \mathbf{R}||_F^2 &= trace(\hat{\mathbf{R}} - \mathbf{R})^T(\hat{\mathbf{R}} - \mathbf{R})) \\ &= 3 + trace(\mathbf{R}\mathbf{R}) - 2trace(\hat{\mathbf{R}}^T \mathbf{R}) \end{aligned} \tag{154}$$

The singular value decomposition of $\mathbf{R}$ can be expressed as follows:

$$\mathbf{R} = \mathbf{U}\mathbf{S}\mathbf{V}^T \quad \mathbf{S} = diag\begin{pmatrix} \sigma_1 & \sigma_2 & \sigma_3 \end{pmatrix} \tag{155}$$

The normal matrix $\mathbf{z_w}$ is defined as $\mathbf{z_w} = \mathbf{V}^T \mathbf{R}^T \mathbf{U}$. The following equations are derived:

$$\begin{aligned} trace(\hat{\mathbf{R}}^T \mathbf{R}) &= trace(\hat{\mathbf{R}}^T \mathbf{V}\mathbf{U}\mathbf{S}\mathbf{V}^T) \\ &= trace(\mathbf{V}^T \hat{\mathbf{R}}^T \mathbf{U}\mathbf{S}) \\ &= trace(\mathbf{z_w}\mathbf{S}) \\ &= \sum_{i=1}^{3} z_i i \sigma_i \leq \sum_{i=1}^{3} \sigma_i \end{aligned} \tag{156}$$

Then, $\mathbf{z_w} = \mathbf{I}$ is hold by defining $\hat{\mathbf{R}} = \mathbf{U}\mathbf{V}^T$, and $trace(\hat{\mathbf{R}}^T\mathbf{R})$ is the maximum. It give the minimum of the equation (152) The obtained $\hat{\mathbf{R}}$ is rewritten as the estimated rotation matrix $\mathbf{R}$.

iii) Intrinsic and Extrinsic Parameter Optimization

Using the obtained intrinsic and extrinsic parameters, the projected points of the calibration data, $\mathbf{M}_j^i = (x_{wj}^i, y_{wj}^i, 0)$, are calculated. It is considered as a function of the intrinsic and extrinsic parameters, and defined as $\hat{\mathbf{m}}(\mathbf{A}, \mathbf{R}^i, \mathbf{t}^i, \mathbf{M}_j^i)$. Under the definition, $\mathbf{m}_j^i = (u_j^i, v_j^i)$, the parameters can be optimized to minimize the following equation.

$$\sum_{i=1}^{n}\sum_{j=1}^{m} ||\mathbf{m}_j^i - \hat{\mathbf{m}}(\mathbf{A}, \mathbf{R}^i, \mathbf{t}^i, \mathbf{M}_j^i)||^2 \qquad (157)$$

Since that is non-linier optimization problem, it is solved by Levenberg-Marquardt method [17].

3. With Aberration

i) Aberration Factor Initial Value Calculation

It is assumed that the aberration is described by the equation (118) and (117) and the effects of κ_3 and more are neglected. The image coordinate without aberration and with aberration is indicated as (u, v) and $(\tilde{u}, \tilde{v})$ respectively. The equation (118) is expressed as the relationship between (u, v) and $(\tilde{u}, \tilde{v})$ as follows:

$$\begin{pmatrix} \tilde{u} \\ \tilde{v} \end{pmatrix} = \begin{pmatrix} u \\ v \end{pmatrix} + \{\kappa_1(x^2 + y^2) + \kappa_2(x^2 + y^2)^2\} \begin{pmatrix} u - u_0 \\ v - v_0 \end{pmatrix} \qquad (158)$$

A symbol is defined as follows.

$$\mathbf{D} = \begin{pmatrix} (u-u_0)(x^2+y^2) & (u-u_0)(x^2+y^2)^2 \\ (v-v_0)(x^2+y^2) & (v-v_0)(x^2+y^2)^2 \end{pmatrix} \quad (159)$$

$$\kappa = \begin{pmatrix} \kappa_1 \\ \kappa_2 \end{pmatrix} \quad (160)$$

$$\mathbf{d} = \begin{pmatrix} \tilde{u}-u \\ \tilde{v}-v \end{pmatrix} \quad (161)$$

Then, the equation (158) is expressed as follows:

$$\mathbf{D}\kappa = \mathbf{d} \quad (162)$$

Here, since the aberration is considered, the calibration data are $\mathbf{M}_j^i = (x_{wj}^i, y_{wj}^i, 0)$ and $\tilde{\mathbf{m}}_j^i = (\tilde{u}_j^i, \tilde{v}_j^i)$. It is supposed that intrinsic and extrinsic parameters are obtained from the result without aberration. The image coordinate without aberration $\mathbf{m}_j^i = (u_j^i, v_j^i)$ are obtained from $\mathbf{M}_j^i = (x_{wj}^i, y_{wj}^i, 0)$ It is supposed that $\mathbf{m}_j^i$ と $\tilde{\mathbf{m}}_j^i$ is hold for 式 (162). Then, κ is calculated by solving the least square minimization as follows:

$$\kappa = (\mathbf{D}^T\mathbf{D})^{-1}\mathbf{D}^T\mathbf{d} \quad (163)$$

iia) Intrinsic and Extrinsic Parameter Optimization

Once κ_1 and κ_2 are obtained, the $(\tilde{u}_j^i, \tilde{v}_j^i)$ are calcurated by substituting $\hat{\mathbf{m}}(\mathbf{A}, \mathbf{R}^i, \mathbf{t}^i, \mathbf{M}_j^i)$ for the equation (158). Using the $\hat{\mathbf{m}}(\mathbf{A}, \mathbf{R}^i, \mathbf{t}^i, \mathbf{M}_j^i)$ instead of the $(\tilde{u}_j^i, \tilde{v}_j^i)$ in the equation (157), the estimated parametes can be optimized. The process to estimate the intrinsic and extrinsic parameters using $(\tilde{u}^i_j, \tilde{v}^i_j)$ in the equation (157) and the process to estimate κ_1 and κ_2 by the equation (163) are excited alternatively until convergence.

iib) Intrinsic and Extrinsic Parameter Optimization

The projected point of $\mathbf{M}^i_j = (x^i_{wj}, y^i_{wj}, 0)$ are defined as $\tilde{\mathbf{m}}(\mathbf{A}, \kappa_1, \kappa_2, \mathbf{R}^i, \mathbf{t}^i, \mathbf{M}^i_j)$ using κ_1 and κ_2. Instead of iia), The equation (157) is expanded under the definition as follows:

$$\sum_{i=1}^{n}\sum_{j=1}^{m} ||\mathbf{m}^i_j - \tilde{\mathbf{m}}(\mathbf{A}, \kappa_1, \kappa_2, \mathbf{R}^i, \mathbf{t}^i, \mathbf{M}^i_j)||^2 \qquad (164)$$

Then, the optimal parameter can be found by minimizing it. Zhang recommends iib) [17]. It is a nonlinear optimization problem, so it is solved by Levenberg-Marquardt method.

iii) Summary

As a summary, the parameters with the aberration is estimated as follows:

(a) The intrinsic and extrinsic parameters are calculated by the process 2-i)

(b) The R is refined by the process 2-ii)

(c) The aberration factors are calculated by the process 3-i)

(d) The parameters are optimized with initial values which are obtained in (a), (b) and (c) by the process 3-iii)

A.3.2 Feature

Zhang's algorithm can estimate not only the five intrinsic and extrinsic parameters, but also the aberration parameters. For the calibration, at least three imaging is needed with different camera direction or the calibration plane's slanting angle

A.3.3 Accuracy

According to Zhang [17], when the noise subjecting to the normal distribution with average 0 and standard deviation σ pixel is added to the projected points on the image in the simulation, the relative error [%] of α、β、γ and the absolute error [pixel] are increased lineally with increasing σ. When the $\sigma = 0.5$, the relative errors of α and β are less than 0.3%, and the absolute errors of u_0 and v_0 are less than about 1pixel. The errors are tend to be reduced with increasing the calibration data, but when it is larger than 3, the reduction is small. Besides, the angle of the calibration plane affects the errors, and it is said that about $45deg$ is optimal [17]. In an experiment with 5 calibration target images with 640480 resolution, the square root of averaged square error (RMS) between the observed data and calculated data by the estimated intrinsic and extrinsic parameters are 0.335 pixel [17].

A.4 Summary

Zhang's algorithm needs at least three times observation, while Tsai's algorithm needs only one time for the calibration and easy to proceed. On the other hand, the principal point and the pixel size are needed to be given for

Tsai's algorithm. When the given pixel size has error, it affect the calibration. In the case of estimating the camera translation and rotation serially, such as the image mosaicking, it is supposed that the intrinsic parameters are changeless; therefore one time calibration is enough. First the principal point and the pixel size are obtained by Zhang's algorithm, and after that, the parameters are given to the Tsai's algorithm. Then the extrinsic parameters can be estimated. In case that the camera has zooming function, it is supposed that the principal point and the pixel size are changeless, and the calibration is executed equally.

B Stereo Calibration

B.1 Stereo Geometry

The camera projection matrix is expressed by the equation (103). It is written using the vector T from the origin to the camera coordinate origin in the global coordinate as follows:

$$P = A\,(R|t) = A\,(R| - RT) \tag{165}$$

The projection matrices of camera1 and camera2 are defined as follows:

$$P_1 = A_1\,(R_1|t_1) = A_1\,(R_1| - R_1T_1) \tag{166}$$

$$P_2 = A_2\,(R_2|t_2) = A_1\,(R_2| - R_2T_2) \tag{167}$$

A point in the global coordinate indicated by the projective coordinate, M_w, is projected to the projective coordinate of the camera1 image, m_1, and the camera2 image, m_2. The image of camera1 is called image1, and the image of camer2 is called image2 bellow.

$$w_1 m_1 = P_1 M_w = A_1 R_1 M_w - A_1 R_1 T_1 \tag{168}$$

$$w_2 m_2 = P_2 M_w = A_2 R_2 M_w - A_2 R_2 T_2 \tag{169}$$

The following equation is derived by eliminating M_w in the equations above:

$$w_1 (A_1 R_1)^{-1} m_1 - w_2 (A_2 R_2)^{-1} m_2 = T_2 - T_1 \tag{170}$$

The following equation is derived under the definition $R = R_1 R_2^{-1}$、$\mathbf{t} = -R\{R2(T1 - T2)\}$:

$$w_1 A_1^{-1} m_1 - w_2 R A_2^{-1} m_2 = \mathbf{t} \tag{171}$$

The R expresses the rotation that rotates the axes to be same direction as the camera2 coordinate axes in the camera1 coordinate. The $\mathbf{t}$ is the vector from the origin to the image2 coordinate origin in the imag1 coordinate. Therefore, the equation (171) does not depends on the location in the global coordinate system.

The matrix T_t is defined as for any vector $\mathbf{x}$, $\mathbf{t} \times \mathbf{x} = T_t\mathbf{x}$ is hold. Then, since the equation (171) tells that $\mathbf{t}$ is expressed by the linear combination of $A_1^{-1}m_1$ and $RA_2^{-1}m_2$, and $A_1^{-1}m_1$, $A_2^{-1}m_2$ and $\mathbf{t}$ are coplanar, $A_1^{-1}m_1 \perp T_t(RA_2^{-1}m_2)$ is hold and the following equation is derived:

$$m_1^T(A_1^{-1})^T T_t R A_2^{-1} m_2 = 0 \tag{172}$$

Here, the following symbols are defined:

$$E = T_t R \tag{173}$$

$$F = (A_1^{-1})^T E A_2^{-1} \tag{174}$$

Then, the equation (172) is rewritten as follows:

$$m_1^T(A_1^{-1})^T E A_2^{-1} m_2 = 0 \tag{175}$$

It is also rewritten as follows:

$$m_1^T F m_2 = 0 \tag{176}$$

These equations dose not depends on the location of the two cameras in the global coordinate system, but depend only on the relationship between the two cameras. They are fundamental equations for the stereo camera. The E in the equation (173) is called as **Essential Matrix**, and the F in the equation (174) is called as **Fundamental Matrix**. The equation (176) is called as **Fundamental Equation**. The Essential Matrix express the stereo camera with the two camera location and direction without the intrinsic parameters, while the Fundamental Matrix express the stereo camera as the relationship between two images comprehending the intrinsic parameters.

B.2 Eight Point Algorithm

Since the equation (176) gives one equation for each corresponding point pair, eight corresponding point pairs are enough to decide the Fundamental Matrix up to the scale factor. As an example method to find the corresponding points, Harris operator can be used to find corners in images. Next, for each pair in both images of corner points, the mating value, such as sum of absolute difference or sum of squared difference is calculated. Then, the matched pairs are adopted as the corresponding points. Generally, the least square solution is calculated using more than eight point pairs.

B.3 Kruppa's equation [16] [6]

The intrinsic parameters can be found from two images without the calibration target. It is called as self-calibration.

The camera projection is described by the 3x4 projection matrix, $P = A[R| - R\mathbf{T}]$, using the upper triangle matrix A, the rotation matrix R and the translation matrix $\mathbf{T}$ as in the equation (103).

The absolute conic is the conic that is considered being on the infinity plane in the projective space and is expressed as follows:

$$x^2 + y^2 + z^2 = 0 \tag{177}$$

$$t = 0 \tag{178}$$

Here, the $(x, y, z, t)^T$ is the projective coordinate of a point in the three-dimensional space

The $(x, y, z, 0)^T$ is assumed to be on the absolute conic. The projective coordinate on the image, m, is expressed as $m = A[R| - R\mathbf{T}] \cdot (x, y, z, 0)^T$ from the equation (104) which express the projection. Under the definition $\mathbf{x} = (x, y, z)^T$, it is written as $m = ARx$. It is written as $x = R^T A^{-1} m$ because the R is normal matrix. Since $(x, y, z, 0)^T$ is on the absolute conic, $\mathbf{x}^T\mathbf{x} = 0$ must be hold. Therefore, the following equation is derived:

$$\begin{aligned} \mathbf{x}^T\mathbf{x} &= \mathbf{m}^T A^{-T} R R^T A^{-1} \mathbf{m} \\ &= \mathbf{m}^T A^{-T} A^{-1} \mathbf{m} = 0 \end{aligned} \tag{179}$$

The image of the absolute conic is the set of points m which subject to the equation above, and $A^{-T}A^{-1}$ can be equated with the image of the absolute conic. It is shown that the absolute conic doesn't have any real coordinate, but its image can be expressed by symmetric 3x3 real matrix. The inverse matrix AA^T of the $A^{-T}A^{-1}$ is called as the dual image of the absolute conic.

Under the definition $C = AA^T$, the tangential line l of the image of absolute conic subject to the following equation:

$$l^T C l = 0 \tag{180}$$

Here, it is assumed that the Fundamental Matrix, F, is obtained by two images. The epipoles $\mathbf{e}$ and $\mathbf{e}'$ in each image are obtained by $F^T\mathbf{e} = 0$ and $F\mathbf{e}' = 0$. The epipolar line for a point $\mathbf{y}$ in each image are $\mathbf{e} \times \mathbf{y}$ and $F^T y$.

Since when the one of the epipolar line is tangential to the absolute conic, the corresponding epipolar line is also tangential to the absolute conic, the following equation is derived:

$$(\mathbf{e} \times \mathbf{y})^T C (\mathbf{e} \times \mathbf{y}) = 0 \quad (181)$$

$$\left(F^T y\right)^T C' \left(F^T y\right) = 0 \quad (182)$$

$$(183)$$

The second order equations of the τ are obtained by substituting the infinity point in the image $\mathbf{y} = (1, \tau, 0)^T$ for the equations above respectively. They are written as $k_0 + k_1\tau + k_2\tau^2 = 0$ and $k'_0 + k'_1\tau + k'_2\tau^2 = 0$. The solutions of the both equation, τ, must be identical because it corresponds to the $\mathbf{y}$. Therefore the following equation is hold.

$$\frac{k_0}{k'_0} = \frac{k_1}{k'_1} = \frac{k_2}{k'_2} \quad (184)$$

This equation is called as **Kruppa's Equation**

When the principal point is known, and the axes are perpendicular (equation (105), $\phi = 0$, the intrinsic parameter, A, is rewritten by replacing the origin to the principal point, $A = diag(f/d_u, f/d_v, 1)$. Under the definition, $\kappa_u = f/d_u$、$\kappa_v = f/d_v$, it is rewritten as $C = AA^T = diag(\kappa_u^2, \kappa_v^2, 1)$. F is expressed as follows:

$$F = \begin{pmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{pmatrix}$$

It is assumed the F is known and the epipole $\mathbf{e} = (\mathbf{e_1}, \mathbf{e_2}, \mathbf{e_3})$ is obtained from the F here. Under the assumption where the intrinsic parameters of the two cameras are identical ($C' = C$), The following equation is derived by substituting them for the equation (181) and (182).

$$\frac{e_3^2\tau^2}{k_u^2} + \frac{e_3^2}{k_v^2} + (e_1\tau - e_2)^2 = 0 \quad (185)$$

$$\frac{(f_{11} + f_{21}\tau)^2}{k_u^2} + \frac{(f_{12} + f_{22}\tau)^2}{k_v^2} + (f_{13} + f_{23})^2 = 0 \quad (186)$$

Then, the Kruppa's equation is rewritten as follows:

$$\frac{k_u^2 k_v^2 e_2^2 + k_u^2 e_3^2}{k_v^2 f_{11}^2 + k_u^2 f_{12}^2 + k_u^2 k_v^2 f_{13}^2} =$$

$$-\frac{k_u^2 k_v^2 e_1 e_2}{k_v^2 f_{11} f_{21} + k_u^2 f_{12} f_{22} + k_u^2 k_v^2 f_{13} f_{23}} = \frac{k_u^2 k_v^2 e_1^2 + k_v^2 e_3^2}{k_v^2 f_{21}^2 + k_u^2 f_{22}^2 + k_u^2 k_v^2 f_{23}^2} \quad (187)$$

The k_u and k_v can be solved from the equation above and

Here, the case of identical two intrinsic parameters is explained above, but even though the intrinsic parameters are not identical, the unknown intrinsic parameters are found as well as the method above.

C Rotation

When a point is stationary and a coordinate system is rotated centered the original point, the coordinate in the rotated coordinate system is represented using rotation matrix R.

$$M' = R^{-1} \cdot M \tag{188}$$

Here, R represents rotation based on the original coordinate system, M is the point's coordinate in the original coordinate system, and the M' is the point's coordinate in the rotated coordinate system.

There are three representations of rotation matrix. First one is that using rotation angles of each coordinate axis. Second one is that using normal perpendicular vectors of each coordinate axis. The third one is that using a vector which direction represents rotation axis and which length represents rotation angle.

C.1 Rotation Angle

The rotation of the coordinate system can be considered that fist, the coordinate system is rotated around x axis by angle α, and then rotated around y-axis by angle β and finally, rotated around z-axis by angle γ. So, the rotation matrix is represented as

$$R = Rz \cdot Ry \cdot Rx \tag{189}$$

$$Rx = \begin{pmatrix} 1 & 0 & 0 \\ 0 & cos\alpha & -sin\alpha \\ 0 & sin\alpha & cos\alpha \end{pmatrix} \tag{190}$$

$$Ry = \begin{pmatrix} cos\beta & 0 & sin\beta \\ 0 & 1 & 0 \\ -sin\beta & 0 & cos\beta \end{pmatrix} \tag{191}$$

$$Rz = \begin{pmatrix} cos\gamma & -sin\gamma & 0 \\ sin\gamma & cos\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{192}$$

C.2 Normal Vectors

Let a normal vector Vx represents the x coordinate axis of the rotated coordinate system in the original coordinate axis. The x component of M' is written as $Vx^T \cdot M$. Since other component is also written in same way using y axis normal vector Vy and z axis normal vector Vz, the rotation matrix can be represented as

$$R = \begin{pmatrix} Vx \\ Vy \\ Vz \end{pmatrix} \tag{193}$$

Here, Vx, Vy and Vz are mutually perpendicular.

C.3 Rotate Vector

Although the actual rotation may not occurred around only one axis, the relation between the rotated coordinate system and the original coordinate system can be represented as a rotation around one axis.

When a vector ω represents the rotation axis and $\alpha = |\omega|$ represent the rotation angle, the rotation matrix is represented as

$$R = e^{[\omega]_\times} = I + \frac{sin\alpha}{\alpha} + \frac{1 - cos\alpha}{\alpha^2} [\omega]_\times^2 \tag{194}$$

The equation is called as Rodrigues equation.

Here, $[\omega]_\times$ is the matrix by which $[\omega]_\times \cdot a = \omega \bigwedge a$, thus

$$[\omega]_\times = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_z & 0 \end{pmatrix} \tag{195}$$

C.4 The Relation Between Rotation Angles And The Rotation Vector

When the rotation is represented by a rotation vector ω as a rotation axis and the rotation angle, the same relation between the rotated coordinate system and the original coordinate system is represented by the following.

First, rotate the coordinate system around x-axis so that the ω in rotated coordinate system is on the z-x plane in the original coordinate system.

$$R_\theta = \begin{pmatrix} 1 & 0 & 0 \\ 0 & cos\theta & -sin\theta \\ 0 & sin\theta & cos\theta \end{pmatrix} \tag{196}$$

$$R_\theta \cdot \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} = \begin{pmatrix} \omega_x \\ \omega_y cos\theta - \omega_z sin\theta \\ \omega_z sin\theta + \omega_z sin\theta \end{pmatrix} \tag{197}$$

$$\omega_y cos\theta - \omega_z sin\theta = 0 \tag{198}$$

$$tan\theta = \frac{\omega_y}{\omega_z} \tag{199}$$

Then, rotate the coordinate system around y-axis so that the ω in rotated coordinate system matches z-axis in the original coordinate system.

$$R_\phi = \begin{pmatrix} cos\phi & 0 & sin\phi \\ 0 & 1 & 0 \\ -sin\phi & 0 & cos\phi \end{pmatrix} \tag{200}$$

$$R_\phi \cdot \begin{pmatrix} \omega_x \\ \omega_y cos\theta - \omega_z sin\theta \\ \omega_z sin\theta + \omega_z sin\theta \end{pmatrix} = \begin{pmatrix} \omega_x cos\phi + \omega_y sin\theta sin\phi + \omega_z cos\theta sin\phi \\ \omega_y cos\theta - \omega_z sin\theta \\ -\omega_x sin\phi + \omega_y sin\theta cos\phi + \omega_z cos\theta cos\phi \end{pmatrix} \tag{201}$$

$$\omega_x cos\phi + \omega_y sin\theta sin\phi + \omega_z cos\theta sin\phi = 0 \tag{202}$$

$$\begin{aligned} tan\phi &= -\frac{\omega_x}{\omega_y sin\theta + \omega_z cos\theta} \\ &= -\frac{1}{cos\theta}\frac{\omega_x \omega_z}{\omega_y^2 + \omega_z^2} \end{aligned} \tag{203}$$

Next, rotate $\alpha = |\omega|$ around z-axis, because the rotation axis is identical to z-axis here.

$$R_\alpha = \begin{pmatrix} cos\alpha & -sin\alpha & 0 \\ sin\alpha & cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{204}$$

$$R_\alpha^{-1} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x cos\alpha + y sin\alpha \\ -x sin\alpha + y cos\alpha \\ z \end{pmatrix} \tag{205}$$

In this case, a point (x, y, z) is rotated $-\alpha$ because the ω doesn't represents the rotation of points, but coordinate system. The rotation of points is opposite rotation to the rotation of coordinate system relatively.

In this notation, the rotation matrix R is subject to

$$R^{-1} = R_\theta^{-1} R_\phi^{-1} R_\alpha^{-1} R_\phi R_\theta \tag{206}$$

Since the rotation is identical between using rotation angles and rotation vector, the both representation indicate same rotation matrix (see Appendix C.5).

$$R = R_\theta^{-1} R_\phi^{-1} R_\alpha R_\phi R_\theta = I + \frac{sin\alpha}{\alpha}\left[\omega\right]_\times + \frac{1-cos\alpha}{\alpha^2}\left[\omega\right]_\times^2 \tag{207}$$

This equation (207) is called as Rodrigues equation.

C.5 Proof of Rodrigues Equation

$$\begin{aligned}
R_\phi R_\theta &= \begin{pmatrix} cos\phi & 0 & sin\phi \\ 0 & 1 & 0 \\ -sin\phi & 0 & cos\phi \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & cos\theta & -sin\theta \\ 0 & sin\theta & cos\theta \end{pmatrix} \\
&= \begin{pmatrix} cos\phi & sin\phi sin\theta & sin\phi cos\theta \\ 0 & cos\theta & -sin\theta \\ -sin\theta & cos\phi sin\theta & cos\phi cos\theta \end{pmatrix} \\
&= cos\theta cos\phi \begin{pmatrix} \frac{1}{cos\theta} & tan\phi tan\theta & tan\phi \\ 0 & \frac{1}{cos\phi} & -\frac{tan\theta}{cos\phi} \\ -\frac{tan\phi}{cos\theta} & tan\theta & 1 \end{pmatrix} \\
&= cos\theta cos\phi \begin{pmatrix} \frac{1}{cos\theta} & tan\phi tan\theta & tan\phi \\ 0 & \frac{1}{cos\phi} & -\frac{tan\theta}{cos\phi} \\ -\frac{tan\phi}{cos\theta} & tan\theta & 1 \end{pmatrix} \\
&= cos\theta cos\phi \begin{pmatrix} \frac{1}{cos\theta} & -\frac{1}{cos\theta}\frac{\omega_x\omega_y}{\omega_y^2+\omega_z^2} & -\frac{1}{cos\theta}\frac{\omega_x\omega_z}{\omega_y^2+\omega_z^2} \\ 0 & \frac{1}{cos\phi} & -\frac{1}{cos\phi}\frac{\omega_y}{\omega_z} \\ -\frac{1}{cos^2\theta} & \frac{\omega_y}{\omega_z} & 1 \end{pmatrix}
\end{aligned} \tag{208}$$

There are two θ between $-\pi$ and π which subject to

$$tan\theta = \frac{\omega_y}{\omega_z} \tag{209}$$

It is because there are two direction when ω is rotated around x axis to be on z-x plane. So, without any loss of generality, it is assumed as follows:

$$-\frac{\pi}{2} < \theta \leq \frac{\pi}{2} \tag{210}$$

Thus, $cos\theta \geq 0$.

$$\begin{aligned} tan\theta &= \frac{\omega_y}{\omega_z} \\ tan^2\theta &= \frac{\omega_y^2}{\omega_z^2} \\ cos^2\theta &= \frac{1}{1+tan^2\theta} = \frac{\omega_z^2}{\omega_z^2+\omega_y^2} \\ cos\theta &= \begin{cases} \frac{\omega_z}{\sqrt{\omega_z^2+\omega_y^2}} & \omega_z \geq 0 \\ \frac{-\omega_z}{\sqrt{\omega_z^2+\omega_y^2}} & \omega_z < 0 \end{cases} \end{aligned} \tag{211}$$

Same as θ, there are two direction of ϕ when ω is rotated around y-axis to be on z-axis. So, it is assumed as follows:

$$-\frac{\pi}{2} < \phi \leq \frac{\pi}{2} \tag{212}$$

Thus, $cos\phi \geq 0$.

$$
\begin{aligned}
tan\phi &= -\frac{1}{cos\theta}\frac{\omega_x\omega_z}{\omega_y^2+\omega_z^2} \\
tan^2\phi &= \frac{1}{cos^2\theta}\frac{\omega_x^2\omega_z^2}{(\omega_y^2+\omega_z^2)^2} = \frac{\omega_x^2}{\omega_y^2+\omega_z^2} \\
cos^2\phi &= \frac{1}{1+tan^2\phi} = \frac{\omega_y^2+\omega_z^2}{\omega_x^2+\omega_y^2+\omega_z^2} \\
cos\phi &= \frac{\sqrt{\omega_y^2+\omega_z^2}}{\sqrt{\omega_x^2+\omega_y^2+\omega_z^2}} = \frac{\sqrt{\omega_y^2+\omega_z^2}}{\alpha}
\end{aligned} \tag{213}
$$

i) $\omega_z \geq 0$

$$
\begin{aligned}
R_\phi R_\theta &= \begin{pmatrix} \frac{\sqrt{\omega_y^2+\omega_z^2}}{\sqrt{\omega_x^2+\omega_y^2+\omega_z^2}} & \frac{-\omega_x\omega_y}{\sqrt{\omega_x^2+\omega_y^2+\omega_z^2}\sqrt{\omega_z^2+\omega_y^2}} & \frac{-\omega_x\omega_z}{\sqrt{\omega_x^2+\omega_y^2+\omega_z^2}\sqrt{\omega_z^2+\omega_y^2}} \\ 0 & \frac{\omega_z}{\omega_z^w+\omega_y^2} & -\frac{\omega_y}{\omega_z^w+\omega_y^2} \\ \frac{\omega_x}{\sqrt{\omega_x^2+\omega_y^2+\omega_z^2}} & \frac{\omega_y}{\sqrt{\omega_x^2+\omega_y^2+\omega_z^2}} & \frac{\omega_z}{\sqrt{\omega_x^2+\omega_y^2+\omega_z^2}} \end{pmatrix} \\
&= \frac{1}{|\omega|}\begin{pmatrix} \sqrt{\omega_z^w+\omega_y^2} & -\frac{\omega_x\omega_y}{\sqrt{\omega_z^w+\omega_y^2}} & -\frac{\omega_x\omega_z}{\sqrt{\omega_z^w+\omega_y^2}} \\ 0 & |\omega|\frac{\omega_z}{\sqrt{\omega_z^w+\omega_y^2}} & |\omega|\frac{\omega_y}{\sqrt{\omega_z^w+\omega_y^2}} \\ \omega_x & \omega_y & \omega_z \end{pmatrix} \\
&= \frac{1}{|\omega|}\begin{pmatrix} |\omega_{yz}| & -\frac{\omega_x\omega_y}{|\omega_{yz}|} & -\frac{\omega_x\omega_z}{|\omega_{yz}|} \\ 0 & |\omega|\frac{\omega_z}{|\omega_{yz}|} & |\omega|\frac{\omega_y}{|\omega_{yz}|} \\ \omega_x & \omega_y & \omega_z \end{pmatrix}
\end{aligned} \tag{214}
$$

$$
\begin{aligned}
R =& R_\theta^{-1} R_\phi^{-1} R_\alpha R_\phi R_\theta \\
=& \frac{1}{|\omega|^2} \begin{pmatrix} |\omega_{yz}| & -\frac{\omega_x\omega_y}{|\omega_{yz}|} & -\frac{\omega_x\omega_z}{|\omega_{yz}|} \\ 0 & |\omega|\frac{\omega_z}{|\omega_{yz}|} & |\omega|\frac{\omega_y}{|\omega_{yz}|} \\ \omega_x & \omega_y & \omega_z \end{pmatrix} \begin{pmatrix} cos\alpha & -sin\alpha & 0 \\ sin\alpha & cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} |\omega_{yz}| & 0 & \omega_x \\ -\frac{\omega_x\omega_y}{|\omega_{yz}|} & |\omega|\frac{\omega_z}{|\omega_{yz}|} & \omega_y \\ -\frac{\omega_x\omega_z}{|\omega_{yz}|} & |\omega|\frac{\omega_y}{|\omega_{yz}|} & \omega_z \end{pmatrix} \\
=& \frac{1}{|\omega|^2} \begin{pmatrix} |\omega_{yz}|^2 cos\alpha + \omega_x^2 & \omega_x\omega_y(1-cos\alpha) - |\omega|\omega_z sin\alpha & \omega_x\omega_z(1-cos\alpha) + |\omega|\omega_y sin\alpha \\ \omega_x\omega_y(1-cos\alpha) + |\omega|\omega_z sin\alpha & |\omega_{xz}|^2 cos\alpha + \omega_y^2 & \omega_y\omega_z(1-cos\alpha) - |\omega|\omega_x sin\alpha \\ \omega_x\omega_z(1-cos\alpha) - |\omega|\omega_y sin\alpha & \omega_y\omega_z(1-cos\alpha) + |\omega|\omega_x sin\alpha & |\omega_{xy}|^2 cos\alpha + \omega_z^2 \end{pmatrix} \\
=& I + \frac{sin\alpha}{\alpha} \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix} + \frac{1}{|\omega|^2} \begin{pmatrix} -|\omega_{yz}|^2(1-cos\alpha) & \omega_x\omega_y(1-cos\alpha) & \omega_x\omega_z(1-cos\alpha) \\ \omega_x\omega_y(1-cos\alpha) & -|\omega_{xz}|^2(1-cos\alpha) & \omega_y\omega_z(1-cos\alpha) \\ \omega_x\omega_z(1-cos\alpha) & \omega_y\omega_z(1-cos\alpha) & -|\omega_{xy}|^2(1-cos\alpha) \end{pmatrix} \\
=& I + \frac{sin\alpha}{\alpha}[\omega]_\times + \frac{1-cos\alpha}{\alpha^2}[\omega]_\times^2
\end{aligned}
\tag{215}
$$

ii) $\omega_z \geq 0$

Although the sign of $cos\theta$ is changed in this case, the Rodrigues equation is derived similarly.

C.6 Infinitesimal Rotation

When the α is small, $sin\alpha$ is approximated to α, and $cos\alpha$ is approximated to 0. Therefore, the Rodrigues equation (207) is approximated to as follows:

$$R = I + [\omega]_\times \tag{216}$$

Therefore, the motion of a point δP can be written as follows:

$$\delta P = RP - P = [\omega]_\times P \tag{217}$$

D Factorization

D.1 Problem Statement

N image feature points are tracked in F frames sequential image. Those feature point locations are measured in image coordinate system. The problem is to estimate camera motion and the three-dimensional location of those feature points using the measurement of image feature points.

D.2 Principle

P_n^g is defined as nth image feature point location in global coordinate system. fth frame camera location and direction is represented by a translation $\mathbf{t_f}$ and a rotation R_f. P_{nf}^c is defined as nth image feature point in fth frame camera coordinate system. The relationship among the feature point in global coordinate system and in image coordinate system is expressed as follows:

$$P_{nf}^c = R_f \cdot (P_n^g - \mathbf{t}_f) \tag{218}$$

It is supposed that the global coordinate system is taken to hold the following equation:

$$\bar{P} = \sum_n P_n = \mathbf{0} \tag{219}$$

Then, the $\bar{P}_f$ is defined as follows:

$$\bar{P}_f = \sum_n P_{nf}^c = -R_f \cdot \mathbf{t}_f \tag{220}$$

In addition, $\tilde{P}_{nf}^c$ is defined as follows:

$$\tilde{P}_{nf}^c = P_{nf}^c - \bar{P}_f = R_f \cdot P_n^g \tag{221}$$

It is expressed using matrices. $\tilde{\mathbf{P}}^c$,$\mathbf{R}$ and $\mathbf{P}^g$ are defined as follows:

$$\tilde{\mathbf{P}}^c = \begin{pmatrix} \tilde{P}^c_{11} & \dots & \tilde{P}^c_{N1} \\ \vdots & \ddots & \vdots \\ \tilde{P}^c_{1F} & \dots & \tilde{P}^c_{NF} \end{pmatrix} \tag{222}$$

$$\mathbf{R} = \begin{pmatrix} R_1 \\ \vdots \\ R_F \end{pmatrix} \tag{223}$$

$$\mathbf{P}^g = \begin{pmatrix} P^g_1 & \dots & P^g_N \end{pmatrix} \tag{224}$$

Then, equation (221) is rewritten in matrices as follows:

$$\tilde{\mathbf{P}}^c = \mathbf{R} \cdot \mathbf{P}^g \tag{225}$$

Since the right term of equation (225) is the product of rank three matrices, $\mathbf{P}^c$ can be decomposed to the product of the camera rotation matrix $\mathbf{R}$ and the feature points location matrix $\mathbf{P}^g$ by singular value decomposition. The principle of factorization is the decomposition.

Suppose that the projection from the camera coordinate system to the image coordinate system is represented by orthographic projection, x and y coordinates in an image coordinate system are same as x and y coordinates in a camera coordinate system. Then, a matrix R_f is defined as the first two column of the matrix R_f. The nth image feature point position $\tilde{P}^i_{nf}$ measured from the average position of N image feature points in f frame image coordinate system is expressed as follows:

$$\tilde{P}^i_{nf} = R_f \cdot P^g_n \tag{226}$$

Similarly, $\tilde{\mathbf{P}}^i$ is defined as follows:

$$\tilde{\mathbf{P}}^i = \begin{pmatrix} \tilde{P}^i_{11} & \dots & \tilde{P}^c_{N1} \\ \vdots & \ddots & \vdots \\ \tilde{P}^i_{1F} & \dots & \tilde{P}^c_{NF} \end{pmatrix} \tag{227}$$

The $\mathbf{R}$ is redefined as R_f. Then, $\tilde{\mathbf{P}}^i$ is expressed as follows:

$$\tilde{\mathbf{P}}^i = \mathbf{R} \cdot \mathbf{P}^g \tag{228}$$

The factorization method solve the camera direction and feature point positions in global coordinate system by decomposing $\tilde{\mathbf{P}}^i$ into $\mathbf{R}$ and $\mathbf{P}^g$ using singular value decomposition. The location of camera coordinate system in the global coordinate system is not completely solved.

$\tilde{\mathbf{P}}^i$ is decomposed into $2F \times 3$ matrix $\hat{\mathbf{R}}$ and $3 \times N$ matrix $\hat{\mathbf{P}}$ using the singular value decomposition. However, in general, $\hat{R}$ does not satisfy the condition in which its $2f-1$th row $\mathbf{i}_f^T$ and $2f$th row $\mathbf{i}_f^T$ have to be mutually perpendicular unit vectors that represent x and y axis respectively in the fth frame camera coordinate system. Therefore, after the decomposition, Q is solved to satisfy the condition in which $\mathbf{R} = \hat{\mathbf{R}} \cdot Q$

Since the normal perpendicularity is invariant with rotation matrix, the Q is solved up to rotation matrices. When it is supposed that the direction of the global coordinate system is same as the direction of the camera coordinate system, the freedom of rotation is eliminated to determine a unique matrix Q. However, when the camera motion is only rotation along an axis, the Q have more freedom than rotation matrices.

In case that all the feature points are coplanar, the rank of $\tilde{\mathbf{P}}^i$ is two, therefore, $\tilde{\mathbf{P}}^i$ cannot be decomposed using singular value decomposition.

D.3 Another Definition

In some textbooks, same relationship showed in previous section is expressed in different way. As it is shown below, the columns of $\tilde{\mathbf{P}}^i$ and $\mathbf{R}$ are exchanged in the other expression.

$p_n^f = (x_n^f, y_n^f)$ is defined as the nth image feature point coordinate in fth frame image coordinate system. Then, a matrix M is defined as follows:

$$M = \left(\begin{array}{cccc} x_1^1 & x_2^1 & \dots & x_N^1 \\ \vdots & \vdots & \vdots & \vdots \\ x_1^F & x_2^F & \dots & x_N^F \\ \hline y_1^1 & y_2^1 & \dots & y_N^1 \\ \vdots & \vdots & \vdots & \vdots \\ y_1^F & y_2^F & \dots & y_N^F \end{array}\right) \tag{229}$$

This matrix M is called as measurement matrix.

$\left(\bar{x^f}, \bar{y^f}\right)$ is defined as the average position of all image feature points in f frame image, and $\tilde{M}$ is defined as the matrix in which elements are difference of each point position in M from the average position as follows:

$$\begin{pmatrix} \bar{x^f} \\ \bar{y^f} \end{pmatrix} = \begin{pmatrix} \frac{1}{N}\sum_{n=1}^{N} x_n^f \\ \frac{1}{N}\sum_{n=1}^{N} y_n^f \end{pmatrix} \tag{230}$$

$$\begin{pmatrix} \tilde{x_n^f} \\ \tilde{y_n^f} \end{pmatrix} = \begin{pmatrix} x_n^f - \bar{x^f} \\ y_n^f - \bar{y^f} \end{pmatrix} \tag{231}$$

$$\tilde{M} = \left(\begin{array}{cccc} \tilde{x}_1^1 & \tilde{x}_2^1 & \dots & \tilde{x}_N^1 \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{x}_1^F & \tilde{x}_2^F & \dots & \tilde{x}_N^F \\ \hline \tilde{y}_1^1 & \tilde{y}_2^1 & \dots & \tilde{y}_N^1 \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{y}_1^F & \tilde{y}_2^F & \dots & \tilde{y}_N^F \end{array}\right) \tag{232}$$

Suppose that the direction of x axis and y axis in fth frame camera coordinate are represented by unit vectors $\mathbf{i}^f, \mathbf{j}^f$ in the global coordinate system respectively, a matrix R is defined as follows:

$$R = \left(\begin{array}{c} \mathbf{i}^1 \\ \vdots \\ \mathbf{i}^F \\ \hline \mathbf{j}^1 \\ \vdots \\ \mathbf{j}^F \end{array}\right) \tag{233}$$

Then, the measurement matrix is expressed as follows:

$$M = R \cdot P \tag{234}$$

D.4 An Example

It is supposed that feature points are located as follows:

$$P_1 = (1, 0, 0)^T \tag{235}$$

$$P_2 = (0, 1, 0)^T \tag{236}$$

$$P_3 = (0, 0, 1)^T \tag{237}$$

$$P_4 = -P_1 - P_2 - P_3 = (-1, -1, -1)^T \tag{238}$$

x and y axis directions of each frame camera coordinate system are supposed to be as follows in the global coordinate system:

$$\mathbf{i^1} = (1, 0, 0)^T \qquad \mathbf{j^1} = (0, 1, 0)^T \tag{239}$$

$$\mathbf{i^2} = (-1, 0, 0)^T \qquad \mathbf{j^2} = (0, 1, 0)^T \tag{240}$$

$$\mathbf{i^3} = (0, 0, 1)^T \qquad \mathbf{j^3} = (0, 1, 0)^T \tag{241}$$

Suppose that the camera projection is represented by orthogonal projection model, the measurement matrix is expressed as follows:

$$\hat{M} = \begin{pmatrix} \mathbf{i}^{1T} \cdot P_1 & \mathbf{i}^{1T} \cdot P_2 & \mathbf{i}^{1T} \cdot P_3 & \mathbf{i}^{1T} \cdot P_4 \\ \mathbf{i}^{2T} \cdot P_1 & \mathbf{i}^{2T} \cdot P_2 & \mathbf{i}^{2T} \cdot P_3 & \mathbf{i}^{2T} \cdot P_4 \\ \mathbf{i}^{3T} \cdot P_1 & \mathbf{i}^{3T} \cdot P_2 & \mathbf{i}^{3T} \cdot P_3 & \mathbf{i}^{3T} \cdot P_4 \\ \hline \mathbf{j}^{1T} \cdot P_1 & \mathbf{j}^{1T} \cdot P_2 & \mathbf{j}^{1T} \cdot P_3 & \mathbf{j}^{1T} \cdot P_4 \\ \mathbf{j}^{2T} \cdot P_1 & \mathbf{j}^{2T} \cdot P_2 & \mathbf{j}^{2T} \cdot P_3 & \mathbf{j}^{2T} \cdot P_4 \\ \mathbf{j}^{3T} \cdot P_1 & \mathbf{j}^{3T} \cdot P_2 & \mathbf{j}^{3T} \cdot P_3 & \mathbf{j}^{3T} \cdot P_4 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & -1 \\ -1 & 0 & 0 & 1 \\ 0 & 0 & 1 & -1 \\ \hline 0 & 1 & 0 & -1 \\ 0 & 1 & 0 & -1 \\ 0 & 1 & 0 & -1 \end{pmatrix} \tag{242}$$

The M is decomposed using singular value decomposition as follows:

$$M = USV^T \tag{243}$$

$$U = \begin{pmatrix} -0.3791035 & -0.5529025 & 0.2248990 & 0.7071067 & 0.0000000 & 0.0000000 \\ 0.3791035 & 0.5529025 & -0.2248989 & 0.7071068 & 0.0000000 & 0.0000000 \\ -0.3280033 & -0.1542043 & -0.9320058 & 0.0000000 & 0.0000000 & 0.0000000 \\ -0.4490638 & 0.3487210 & 0.1003428 & 0.0000000 & -0.5773503 & -0.5773503 \\ -0.4490638 & 0.3487210 & 0.1003428 & 0.0000000 & 0.7886751 & -0.2113249 \\ -0.4490638 & 0.3487210 & 0.1003428 & 0.0000000 & -0.2113249 & 0.7886751 \end{pmatrix} \tag{244}$$

$$S = \begin{pmatrix} 2.901523 & 0.000000 & 0.000000 & 0.000000 \\ 0.000000 & 1.544918 & 0.000000 & 0.000000 \\ 0.000000 & 0.000000 & 1.092885 & 0.000000 \\ 0.000000 & 0.000000 & 0.000000 & 0.000000 \\ 0.000000 & 0.000000 & 0.000000 & 0.000000 \\ 0.000000 & 0.000000 & 0.000000 & 0.000000 \end{pmatrix} \tag{245}$$

$$V = \begin{pmatrix} -0.2613135 & -0.7157694 & 0.4115694 & 0.5000001 \\ -0.4643050 & 0.6771641 & 0.2754439 & 0.4999999 \\ -0.1130452 & -0.0998139 & -0.8527943 & 0.4999999 \\ 0.8386638 & 0.1384191 & 0.1657807 & 0.5000000 \end{pmatrix} \tag{246}$$

Since the diagonal $(1,1),(2,2)and(3,3)$ elements in S are non-zero, a matrix $\tilde{S}$ is defined as the top left 3×3 sub-matrix of S. $\tilde{U}$ is defined as 1,2 and 3 rows sub-matrix of U. $\tilde{V}$ is defined as 1,2 and 3 rows sub-matrix of V. Then, the matrix M is expressed as follows:

$$M = \tilde{U} \cdot \tilde{S} \cdot \tilde{V}^T \tag{247}$$

$\tilde{R}$ is defined as follows:

$$\tilde{R} = \tilde{U} \cdot \tilde{S}^{1/2} = \begin{pmatrix} -0.6457594 & -0.6872284 & 0.2351120 \\ 0.6457594 & 0.6872284 & -0.2351120 \\ -0.5587161 & -0.1916680 & -0.9743294 \\ -0.7649288 & 0.4334418 & 0.1048995 \\ -0.7649288 & 0.4334418 & 0.1048995 \\ -0.7649288 & 0.4334418 & 0.1048995 \end{pmatrix} \tag{248}$$

$\tilde{P}$ is defined as follows:

$$\tilde{P} = \tilde{S}^{1/2} \cdot \tilde{V}^T \begin{pmatrix} -0.4451176 & -0.7908905 & -0.1925596 & 1.4285677 \\ -0.8896635 & 0.8416794 & -0.1240636 & 0.1720476 \\ 0.4302594 & 0.2879522 & -0.8915207 & 0.1733091 \end{pmatrix} \quad (249)$$

Then, M is decomposed as $M = \tilde{R} \cdot \tilde{P}$, however, fth and $f+3$th column vector in $\tilde{R}$, which are suppoed to represent x and y axis of fth frame camera coordinate system respectively, are not perpendicular. In order to make them perpendicular, the R is defined as $R = \tilde{R} \cdot Q$, and the Q is determined to make fth and $f+3$th column vectors of R perpendicular.

Q is determined using the Newton method as follows:

$$Q = \begin{pmatrix} 0.9322650 & -0.0942472 & -0.1020902 \\ -0.0948945 & 1.2398700 & 0.0346459 \\ -0.1020046 & 0.0352958 & 0.9643888 \end{pmatrix} \quad (250)$$

Then, R is calculated as follows:

$$R = \tilde{R} \cdot Q = \begin{pmatrix} -0.5607872 & -0.7829144 & 0.2688554 \\ 0.5607872 & 0.7829144 & -0.2688554 \\ -0.4032972 & -0.2193757 & -0.8892335 \\ -0.7649479 & 0.6132063 & 0.1942726 \\ -0.7649479 & 0.6132063 & 0.1942726 \\ -0.7649479 & 0.6132063 & 0.1942726 \end{pmatrix} \quad (251)$$

P is also calculated using Q as follows:

$$P = Q^{-1} \cdot \tilde{P} = \begin{pmatrix} -0.5090860 & -0.7648207 & -0.3210116 & 1.5949183 \\ -0.7682569 & 0.6148545 & -0.0979502 & 0.2513526 \\ 0.4204182 & 0.1951860 & -0.9548100 & 0.3392059 \end{pmatrix} \quad (252)$$

Eventually, the measurement matrix M is decomposed as $M = R \cdot P$. Here, fth and $f+3$the column vectors in R represent each unit vector direction of x and y axis of f frame camera coordinate system in the global coordinate system. The nth row vector in P represents nth feature point position in the global coordinate system.

D.5 Consideration

The x and y axis of the first frame camera coordinate system are given as $\mathbf{i}^1 = (1, 0, 0)^T$ and $\mathbf{j}^1 = (0, 1, 0)^T$ at the beginning in the example. However, the obtained x and y axis are $\hat{\mathbf{i}}^1 = (-0.5607872, -0.7829144, 0.2688554)^T$ and $\hat{\mathbf{j}}^1 = (-0.7649479, 0.6132063, 0.1942726)^T$ In order to explain the reason why they are different, a matrix is defined as follows:

$$\phi = \begin{pmatrix} -0.5607872 & -0.7829144 & 0.2688554 \\ -0.7649479 & 0.6132063 & 0.1942726 \\ -0.4032972 & -0.2193757 & -0.8892335 \end{pmatrix} \tag{253}$$

The R_t and P_t are defined using given values as follows:

$$R_t = \begin{pmatrix} \mathbf{i}^{1T} \\ \mathbf{i}^{2T} \\ \mathbf{i}^{3T} \\ \mathbf{j}^{1T} \\ \mathbf{j}^{2T} \\ \mathbf{j}^{3T} \end{pmatrix} \tag{254}$$

$$P_t = \begin{pmatrix} P_1 & P_2 & P_3 & P_4 \end{pmatrix} \tag{255}$$

Then, the following equations are hold:

$$R = R_t \cdot \phi \tag{256}$$

$$P = \phi^{-1} \cdot P_t \tag{257}$$

That is, Q which satisfies that x and y axis are perpendicular is not unique. Suppose that Q' is defined as $Q' = Q \cdot \phi^{-1}$, and R and P is calculated as $R = \tilde{R} * Q'$ and $P = Q'^{-1}\tilde{P}$, R and P are obtained as $R = R_t$,$P = P_t$.

The freedom degree of ϕ is up to a rotation matrix generally because the matrix is a transformation matrix which preserves the perpendicularity between x and y axes of camera coordinate systems in multiple frames. However, since equations, $\mathbf{i}^1 = -\mathbf{i}^2$ and $\mathbf{j}^1 = \mathbf{j}^2$, are hold in the example, relationships, $\mathbf{i}^1 \perp \mathbf{j}^1$ and $\mathbf{i}^2 \perp \mathbf{j}^2$, are same condition. Moreover, an equation, $\mathbf{j}^3 = \mathbf{j}^1$, is hold. Therefore, when Q is determined, there are exist perpendicularity conditions only between $\mathbf{i}^1$ and $\mathbf{j}^1$ and between $\mathbf{i}^3$ and $\mathbf{j}^1$. In those condition, a matrix which is not a rotation matrix can be a transformation matrix which preserve perpendicularity between x and y aces of camera coordinate systems. If x and y axis of additional frame camera coordinate system is not placed on the plane of $\mathbf{i}^1 and \mathbf{j}^1$ and the plane of $\mathbf{i}^3, \mathbf{j}^1$, ϕ is a rotation matrix.

Eventually, Q is determined up to a rotation matrix ϕ. Therefore, it is supposed that axes of the first frame camera coordinate system are in same direction as axes of the global coordinate system. Then, the unique Q is obtained under the condition in which i^1 and j^1 are given as $\mathbf{i}^1 = (1, 0, 0)$, $\mathbf{i}^1 = (0, 1, 0)$.

When the number of feature points is less than three, $\tilde{P}_3$ is represented by a linear combination of $\tilde{P}_1$ and $\tilde{P}_2$ as $\tilde{P}_3 = -\tilde{P}_1 - \tilde{P}_2$. Since the rank of the matrix $\tilde{P}$ is two in this case, the rank of the measurement matrix $\tilde{M}$ is also two. Therefore, the decomposition into R and P cannot be executed, when the number of feature points is less than three. Even though the number of feature points is more than four, if those feature points are coplanar, the decomposition cannot be executed similarly. Therefore, the number of feature points has to be more than four and all those feature points must not be coplanar.

E Stabilization

E.1 Phase Correlation

$F_1(\omega)$ is defined as the Fourier transformation of a function $f_1(t)$, and $F_2(\omega)$ is defined as the Fourier transformation of a function $f_2(t)$.

$$F_1(\omega) = \int_{-\infty}^{\infty} f_1(t)e^{-i\omega t}dt \tag{258}$$

$$F_2(\omega) = \int_{-\infty}^{\infty} f_2(t)e^{-i\omega t}dt \tag{259}$$

The product of $F_1(\omega)$ and $F_2(\omega)$ is expressed as follows:

$$F_1(\omega)F_2(\omega) = \int_{-\infty}^{\infty} \left\{ \int_{-\infty}^{\infty} f_1(t')f_2(t-t')dt' \right\} e^{-i\omega t}dt \tag{260}$$

Therefore, the inverse Fourier transformation of $F_1(\omega)F_2(\omega)$ is expressed as follows:

$$\frac{1}{2\pi}\int_{\infty}^{\infty} F_1(\omega)F_2(\omega)e^{i\omega t}d\omega = \int_{-\infty}^{\infty} f_1(t')f_2(t-t')dt' \tag{261}$$

Suppose that $F(\omega)$ is the Fourier transformation of $f(t)$, $F(\omega)$ and conjugate of $F(\omega)$ can be expressed as follows:

$$\begin{aligned} F(\omega) =& A(\omega) - iB(\omega) \\ =& \int_{-\infty}^{\infty} f(t)cos(\omega t)dx - i\int_{-\infty}^{\infty} f(t)sin(\omega t)dx \\ =& \int_{-\infty}^{\infty} f(t)e^{-i\omega t} \end{aligned} \tag{262}$$

$$\begin{aligned} F^*(\omega) =& A(\omega) + iB(\omega) \\ =& \int_{-\infty}^{\infty} f(t)cos(\omega t)dx + i\int_{-\infty}^{\infty} f(t)sin(\omega t)dx \\ =& \int_{-\infty}^{\infty} f(t)e^{i\omega t} \\ =& \int_{-\infty}^{\infty} f(-t)e^{-i\omega t} \end{aligned} \tag{263}$$

Therefore, the inverse Fourier transformation of $F_1(\omega)F_2^*(\omega)$ is expressed as follows:

$$\frac{1}{2\pi}\int_{\infty}^{\infty} F_1(\omega)F_2^*(\omega)e^{i\omega t}d\omega = \int_{-\infty}^{\infty} f_1(t')f_2(t'-t)dt' \tag{264}$$

Suppose that $f_2(x, y) = f_1(x + u, y + v)$, the Fourier transformation of $f_2(x, y)$ is expressed as follows:

$$\begin{aligned}
F_2(\omega_x, \omega_y) &= \iint f_2(x, y)e^{-i(\omega_x x+\omega_y y)}dxdy \\
&= \iint f_1(x + u, y + v)e^{-i(\omega_x x+\omega_y y)}dxdy \\
&= \iint f_1(x', y')e^{-i(\omega_x(x'-u)+\omega_y(y'-v))}dx'dy' \\
&= e^{i(\omega_x u+\omega_y v)} \iint f_1(x', y')e^{-i(\omega_x x'+\omega_y y')}dx'dy' \\
&= e^{i(\omega_x u+\omega_y v)}F_1(\omega_x, \omega_y)
\end{aligned} \tag{265}$$

Therefore, the product of F_1 and conjugate of F_2 is express as follows:

$$\begin{aligned}
F(\omega_x, \omega_y) &= F_1(\omega_x, \omega_y) \cdot F_2^*(\omega_x, \omega_y) \\
&= e^{-i(\omega_x u+\omega_y v)}|F_1(\omega_x, \omega_y)|^2
\end{aligned} \tag{266}$$

Then, $F(\tilde{\omega_x}, \omega_y)$ is defined as follows:

$$F(\tilde{\omega_x}, \omega_y) = \frac{F(\omega_x, \omega_y)}{|F(\omega_x, \omega_y)|} = e^{-i(\omega_x u+\omega_y v)} \tag{267}$$

Therefore, the inverse Fourier transformation of $F(\tilde{\omega_x}, \omega_y)$, $f(x, y)$, is expressed as follows:

$$\begin{aligned}
f(x, y) &= \frac{1}{2\pi}\iint e^{-i(\omega_x u+\omega_y v)}e^{i(\omega_x x+\omega_y y)}d\omega_x d\omega_y \\
&= \frac{1}{2\pi}\iint e^{i\{\omega_x(x-u)+\omega_y(y-v)\}}d\omega_x d\omega_y
\end{aligned} \tag{268}$$

The real part of f(x,y) is expressed as follows:

$$Re\{f(x,y)\} = \frac{1}{2\pi}\iint cos\{\omega_x(x-u)+\omega_y(y-v)\}d\omega_x d\omega_y \tag{269}$$

Therefore, $Re\{f(x,y)\}$ have a maximum point at $x = u$, $y = v$.

Suppose $f_1(x,y)$ and $f_2(x,y)$ are two image intensity functions, the maximum point of phase correlation represent translation of image.

E.2 Log-Polar Coordinate

The log-polar coordinate is a two-dimensional coordinate system. One parameter is a logarithmic distance from the origin. The other parameter is an angle from a base line. When a point is expressed as (r,θ) in polar coordinate system, $(\log r,\theta)$ is the coordinate in log-polar coordinate system.

Suppose that $f(\log r,\theta)$ is a image intensity function in log-polar coordinate system, α times scaled and ϕ rotated image is represented as $f(\log\frac{r}{\alpha},\theta+\phi) = f(\log r - \log\alpha,\theta+\phi)$. Therefore, the scaling and rotation is represented as a displacement in log-polar coordinate system.

Fig.30 shows an example of log-polar coordinate representation of image.

E.3 Stabilization Algorithm

Stabilization is an image processing technique to remove the image motion caused by camera motion. The phase correlation is applied to consecutive two images in Cartesian and log-polar coordinate system in the stabilization algorithm. Since the scaling and rotation is represented as a displacement in log-polar coordinate, the maximum point of phase correlation indicates

Figure 30: a) Cartesian Coordinate and b) Log-Polar Coordiante Image

the scaling and rotation between images. When the peak of phase correlation is higher in Cartesian coordinate than that in log-polar coordinate, the translation in Cartesian coordinate is removed first. Then, the phase correlation is applied again in log-polar coordinate system to extract and remove scaling and rotation. Fig.31 shows an example result in case that image is translated. Fig.32 shows an example result in case that image is scaled and

Figure 31: a) Translated Image and b) Stabilized Image

rotated. In both example, Fig.30-a is the target image of stabilization.

Figure 32: a) Scaled and Rotated Image and b) Stabilized Image

Theoretically, the phase correlation works to find displacement only in each aces, in other words, it works only for pure translation in image when it is applied in Cartesian coordinate system, or it works only for combination of scaling and rotation along z-axis in log-polar coordinate system. For other motion, such as combination of translation and rotation, the phase correlation is not guaranteed to have a peak to indicate camera motion. However, when the motion is close to a pure translation or close to a combination of scaling and rotation, the highest peak between two coordinate systems indicates the motion.

References

[1] Koichiro Deguchi. *Robot Vision no Kiso (in Japanese)*. Corona, 2000.

[2] David Demirdjian and Radu Horaud. Motion-egomotion discrimination and motion segmentation from image-pair streams. *Computer Vision and Image Understanding*, 78:53–68, 2000.

[3] Cornelia Fermuller and Yiannis Aloimonos. On the geometry of visual correspondence. Technical report, Computer Vision Laboratory, Center for Automation Research, University of Maryland, July 1994.

[4] Cornelia Fermuller and Yiannis Aloimonos. Observability of 3d motion. *International Journal of Computer Vision*, 37(1):43–63, 2000.

[5] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in computer vision*. CAMBRIDGE, 2000.

[6] Richard I. Hartley. Kruppa's equations derived from the fundamental matrix. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2), February 1997.

[7] Thomas S. Huang and Arun N. Netravali. Motion and structure from feature correspondences: A review. *Proceedings of the IEEE*, 82(2):252–268, February 1994.

[8] Nakatani. Active camera niyoru gazou tougou ni kansuru kenkyuu (in japanese). Master's thesis, Department of Mechanical and Environmential Informatics, Tokyo Institute of Technology, 2000.

[9] Abhijit S. Ogale and Yiannis Aloimonos. Shape and the stereo correspondence problem. *under review*, 2004.

[10] Abhijit S. Ogale, Cornelia Fermuller, and Yiannis Aloimonos. The motion segmentation hierarchy. *under review*, 2004.

[11] John Oliensis. Exact two-image structure from motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(12):1618–1633, May 2002.

[12] John Oliensis. The least-squares error for structure from infinitesimal motion. *European Conference on Computer Vision*, May 2004.

[13] Amnon Shashua. Projective depth: A geometric invariant for 3d reconstruction from two perspective/orthographic views and for visual recognition. *IEEE International Conference on Cputer Vision*, pages 583–590, May 1993.

[14] Emanuele Trucco and Alessandro Verri. *INTRODUCTORY TECHNIQUES for 3-D COMPUTER VISION*. PRENTICE HALL, 1998.

[15] Roger. Y. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal fo Robotics and Automation*, RA-3(4):323–344,, August 1987.

[16] Gang Xu and Sabro Tsuji. *San Jigen Vision (in Japanese)*. Kyoritu, 1998.

[17] Zengyou Zhang. A flexible new technique for camera calibration. Technical report, Microsoft Research Microsoft Corporation, December 1998.

[18] Zengyou Zhang, Quang-Tuan Luong, and Olivier Faugeras. Motion of an uncalibrated stereo rig: Self-calibration and metric reconstruction. *IEEE Transactions on Robotics and Automation*, 12(1):103–113, February 1996.