Ground Floor Detection and Ego-Motion Estimation
for Visual Navigation of Mobile Robots

by
Young-Geun Kim

Thesis Advisor: Hakil Kim

A THESIS
Submitted to the faculty of
INHA UNIVERSITY
in partial fulfillment of the requirements
for the degree of
MASTER OF SCIENCE
The Graduate School of
Information Technology and Telecommunication
February 2004
Ground Floor Detection and Ego-Motion Estimation
for Visual Navigation of Mobile Robots

by
Young-Geun Kim

A thesis submitted to the faculty of Inha University in partial fulfillment of the requirements for the degree of Master of Science in the Graduate School of Information Technology and Telecommunication

Incheon, Korea
February 2004

Approved by Prof. Hakil Kim

Major Advisor
This certifies that the thesis of Young Geun Kim is approved.

Chair ______________________

Referee ______________________

Referee ______________________

The Graduate School of
Information Technology and Telecommunication
Inha University
February 2004
ABSTRACT

Ground Floor Detection and Ego-Motion Estimation for Visual Navigation of Mobile Robots

Young Geun Kim
The Graduate school of
Information Technology and Telecommunication
Inha University

This paper addresses two coupled problems in visual navigation for mobile robots operating in unknown structured environments: local map building and incremental localization. Many traditional vision based methods can be divided into two major classes: First, stereo-based methods produce a dense disparity map with two or more images and construct the map into a 3D map. Depth information obtained from the 3D map may provide 3D visual cues such as paths or obstacles for navigation but these methods may fail when particular types of features are not supported in images. Secondly, motion-based methods compute optical flows in image sequences. These techniques are appropriate for detecting interesting objects of dominant motions but do not work in some types of scenes such as walls.

For mobile robots in structured environments, the ground floor is a very interesting object because it represents a local map of movable paths except for other static or dynamic objects. In spite of the merit, it is not easy to be detected by image properties, stereo-based or motion-based methods. Unlike these approaches, the geometric fact that other static or dynamic
objects are placed on the ground floor perpendicularly can provide an effective clue to separate the ground floor from scene images.

The purpose of this thesis is to propose ground floor detection and ego-motion estimation algorithms to solve the two coupled problems. In order to detect the ground floor a geometric fact observed in the scene and some assumptions about images are exploited so that a plane normal can be an effective clue to separate the ground floor from the scene. In order to compute plane normals in images, two methods are proposed and combined together with a designed iterative refinement process so that the ground floor can be detected as accurate as possible although mismatched point correspondences are obtained. In order to find camera ego-motion parameters, a plane, such as the ground floor, is used because the ego-motion model does not carry 3D information any more if plane information is given. Given the image two methods are developed which are the inverse Jacobian method computing a least squared estimate derived from the iterative Newton-Raphson formula based on image Jacobian and the image Gradient method computing an optiminimal ego-motion parameters robustly so that it minimizes computation errors although inaccurate and noisy image motion vector are obtained.

The preliminary experiments use synthetic data and real data to verify that the proposed algorithms perform ground floor detection and camera ego-motion estimation correctly and also to show the appropriateness and the effectiveness of the algorithms for visual navigation of mobile robots.

Key words: Ground floor detection, plane normal computation, image motion estimation, ego-motion estimation, visual navigation, mobile robot.
본 논문은 비전 기반의 자율 주행 로봇이 미지의 구조화된 환경에서 동작할 때 나타나는 두 가지 결합된 문제들, 국부 환경지도 작성과 점진적인 자기 위치 인식문제를 고려한다. 이러한 문제를 해결하기 위한 전형적인 비전 기반의 접근방법들은 크게 두 가지, 스테레오 기반의 접근 방법과 모션 기반의 접근 방법으로 분류될 수 있다. 국부 환경지도 작성은 구조화된 환경의 영상들에서의 특정한 형태의 특징들이 잘 나타나지 않는 경우에는 조밀한 차영상을 얻을 수 없기 때문에, 복원된 3차원 지도의 정확성이 낮아지고 깊이 정보의 신뢰도도 떨어지게 된다. 국부 환경지도 작성을 위한 모션 기반의 접근 방법들은 연속된 영상으로부터 움직임 정보를 추출하고 눈에 띄는 움직임을 갖는 물체들을 검출한다. 이러한 방법들은 전방에 나타내는 동적 혹은 정적의 장애물을 인식하는 용도로는 적합할 수 있지만, 벽과 같이 움직임이 두드러지지 않은 물체들을 검출하기는 어렵다. 따라서, 이러한 방법들은 영상에서 특정한 형태의 특징이 잘 나타나지 않는 구조화된 환경에는 적합하지 않다.

구조화된 환경에서 동작하는 로봇에게 바닥면은 매우 흥미로운 대상이다. 그 이유는 바닥면이 정적 혹은 동적 물체를 제외한 주행 가능한 경로를 나타내기 때문이다. 이러한 장점에도 불구하고 바닥면 영상에는 모서리, 에지, 색상 또는 텍스처와 같이 특정한 공동된 특징들이 나타나지 않기 때문에, 영상적 특성이나 스테레오 방법 혹은 모션 기반의 방법들로는 직접 바닥면을 검출하기가 쉽지 않다. 이러한 접근 방법들과는 달리 구조화된 환경에서 관찰될 수 있는 정적 혹은 동적 물체들이 바닥면과 수직하게 놓여진다는 기하학적 사실은 영상에서 바닥면 만을 분리해주거나 또는 주행 가능성을 갖는 효과적인 단서를 제공할 수 있다.
본 논문의 목적은 두 가지 문제를 동시에 해결하기 위해 바닥면을 검출해서 사용하는 새로운 방법을 제안하는 것이다. 바닥면 검출을 위해 정적 혹은 동적 물체들이 바닥면에 수직하게 놓여진다는 기하학적 사실을 활용한다. 그리고 이러한 사실을 제안되는 알고리즘에 적용하기 위해 영상이 작은 영역으로 분할되고 각 영역들은 공간상에서 하나의 평면에 해당되며 적어도 3개 이상의 점들로 구성되어 하나의 평면을 정의할 수 있다고 가정한다. 이러한 기하학적 사실과 가정들은 평면의 수직 벡터가 환경 영상에서 바닥면을 분리할 수 있는 효과적인 단서가 될 수 있게 한다.

제안된 바닥면 검출 방법은 연속된 영상에서 보여지는 영상의 움직임을 검출하고 영상을 작은 영역들로 분할한다. 그리고 각 영역들에 대해 평면의 수직 벡터를 계산한 후 반복적인 정제과정을 거쳐 점진적으로 바닥면 영상을 검출한다. 알고리즘 개발을 위해 평면의 수직 벡터를 계산하는 두 가지 방법을 제안한다. 첫 번째 방법은 공간상의 평면에 의해 야기되는 투영 기하학 관계식을 유도하여 3개의 동일점들로부터 수직 벡터를 직접 계산해내는 방법이다. 두 번째 방법은 잘못 정합된 동일점들이 존재하더라도 계산시 오류를 최소화시킬 수 있도록 최소의 최적 해를 구하는 방법이다. 이 방법은 영상 분할 기법을 기반으로 하는 반복적인 정제과정과 혼합되어 잘못된 정합점들이 존재하더라도 강연하고 정확하게 바닥면 영상만을 분리할 수 있게 한다.

제안된 자기 위치인식 방법은 바닥면과 같은 공간상의 하나의 평면을 사용한다. 그 이유는 만약 하나의 평면에 대한 정보가 제공된다면 공간상의 카메라의 운동과 영상의 모션들 사이의 관계에는 더 이상 3차원 공간정보를 포함하지 않게 된다. 알고리즘 개발을 위해 카메라의 자기 운동 파라미터를 구하는 두 가지 방법들이 제안된다. 첫 번째 방법은 영상의 모션과 카메라의 공간상에서의 운동과의 관계를 나타내는 영상 자코비안을 유도하고, 반복적으로 해를 구하는 Newton-Raphson 공식을 도입하여 카메라의 정확한 자기운동 파라미터의 최소 자승 해를 구하는 영상의 역 자코비안에 기반한 방법이다. 두 번째 방법은 잘못 정합된 동일점들이 존재하더라도 계산상의 오류를 최소화시킬 수 있도록 최소의 최적 해를 구하는 방법으로서 영상
Gradient에 기반한 방법이다. 이 방법은 첫 번째 제안된 알고리즘에 포함되어 해를 구하는 데 있어서 빠른 수렴속도를 보인다.

제안된 알고리즘들의 정당성을 입증하기 위해 인공적으로 생성된 데이터와 실제 영상을 사용하여 실험들을 수행한다. 실험 결과를 통해 비전 기반의 자율 주행로봇에 대해 제안된 알고리즘들의 적합성과 효율성을 보인다.
## CONTENTS

ABSTRACT ................................................................................................................. i

CONTENTS ................................................................................................................ iii

LIST OF FIGURES ...................................................................................................... viii

LIST OF ALGORITHMS ............................................................................................ x

NOMENCLATURE ...................................................................................................... xi

1. INTRODUCTION .................................................................................................. 1
   1.1. Motivation ........................................................................................................ 1
   1.2. Research Goal .................................................................................................. 2
   1.3. Outline of the Thesis ....................................................................................... 5

2. PRELIMINARY ....................................................................................................... 6
   2.1. Camera Model ................................................................................................. 6
      2.1.1 Normalized Camera Model ....................................................................... 6
      2.1.2 General Camera Model ........................................................................... 8
      2.1.3 Camera Motion in 3-space ....................................................................... 11
   2.2. Image Motion Estimation .............................................................................. 12
      2.2.1 Image Brightness Constancy Equation ................................................. 14

3. GROUND FLOOR DETECTION ........................................................................... 16
   3.1. Planar Homography ....................................................................................... 16
   3.2. Plane Normal Computation ........................................................................... 18
      3.2.1 Case I: Three Image Point Correspondences ........................................ 21
      3.2.2 Case II: Three or More Noisy Point Correspondences .................... 25
   3.3. Layered Ground Floor Segmentation .......................................................... 27
      3.3.1 Optical Flow Estimation ......................................................................... 29
      3.3.2 Iterative Refinement Process .................................................................. 30
      3.3.3 Layered Image Representation ............................................................... 31
   3.4. Experimental Results ..................................................................................... 32
      3.4.1 Image Motion Estimation ......................................................................... 33
LIST OF FIGURES

Figure 1.1. A sample image of an indoor scene.......................................................3
Figure 2.1. Projection onto the normalized image plane $\pi$ at $f = 1$..............7
Figure 2.2. Projection onto an image plane $I$ with $f \neq 1$.............................8
Figure 2.3. Two normalized cameras in 3-space..............................................11
Figure 2.4. Difference between the image motion and the optical flow..........13
Figure 3.1. Planar homography induced by a plane......................................16
Figure 3.2. The concept of a homography.....................................................17
Figure 3.3. Central projection.................................................................18
Figure 3.4. Planar homography between two views.....................................19
Figure 3.5. A scene plane normal with 3 image points...............................22
Figure 3.6. Geometric meaning of $x'_i \times m$.............................................24
Figure 3.7. A scene plane normal with $n \geq 3$ noisy image points..............25
Figure 3.8. Framework of the layered ground floor segmentation..............28
Figure 3.9. Multi-scale coarse-to-fine estimation......................................29
Figure 3.10. Reference images for two image sequences..........................32
Figure 3.11. Image motion estimation.........................................................33
Figure 3.12. Image segmentation..............................................................34
Figure 3.13. Plane normal computation......................................................35
Figure 3.14. Iterative refinement process....................................................36
Figure 3.15. Intermediate processes for the case of no obstacles...............37
Figure 3.16. Intermediate processes for the case of static obstacles............38
Figure 3.17. The foreground layers.........................................................39
Figure 3.18. Ground truth layers vs. estimated layers..............................40
Figure 4.1. Projection of 3D motion of onto the normalized image.............43
Figure 4.2. Camera ego-motion estimation using ground floor detection...50
Figure 4.3. Bias and sensitivities on translational, angular velocities and
image transfer error...............................................................................70
Figure 4.4. Experimental results for pure translation............................71
Figure 4.5. Experimental results for pure rotation..........................72
Figure 4.6. Experimental results for general motion........................73
LIST OF ALGORITHMS

Algorithm 4.1. Summary of the inverse image Jacobian based algorithm. .58
Algorithm 4.2. Summary of the image gradient based algorithm. ............... 64
NOMENCLATURE

To enhance the readability the notations used throughout the thesis are summarized here. For matrices and vectors bold face fonts are used. Scalar values will be represented by italic fonts.

\{C\} : camera coordinate systems
\{I\} : image coordinate systems
\(\pi\) : normalized image plane
\(f\) : camera focal length
\(X\) : scene point in 3-space (3-vector)
\(\tilde{X}\) : homogeneous coordinates of \(X\) (4-vector)
\(x\) : image point of \(X\) in normalized image coordinates (3-vector)
\(proj\_p()\) : projection operator onto a plane \(p\)
\(P\) : normalized camera projection matrix (3\(\times\)4 matrix)
\(p\) : image of \(X\) in pixel coordinates (2-vector)
\(\tilde{p}\) : homogenous coordinates of \(p\) (3-vector)
\(x\), \(x'_i\) : image points in the 1\(st\) and 2\(nd\) views, respectively (3-vector)
\(l_i\), \(l'_i\) : epipolar lines for \(x_i\), \(x'_i\) in the 1\(st\) and 2\(nd\) views (3-vector)
e, \(e'\) : epipoles of the 1\(st\) and 2\(nd\) views (3-vector)
c : camera center (3-vector)
\(K\) : camera calibration matrix (3\(\times\)3 matrix)
\((p_x, p_y)\) : principle point in image coordinates
\(m_x, m_y\) : the number of pixels per unit distance in image coordinates
\(T\) : homogeneous transformation matrix (4\(\times\)4 matrix)
\(\nabla E\) : spatial image gradient (2-vector)
\(E_t\) : temporal image difference
\(H\) : homography matrix (3\(\times\)3 matrix)
\(M, m\) : sub-matrices of the camera projection matrix
\(\Pi\) : scene plane in 3-space (4-vector)
v: plane normal (3-vector)
v': optimum plane normal
v_i: estimate of the plane normal of a region \( R_i \)
\( \mathbf{v}_G \): plane normal of the ground floor
\( \theta \): angle difference
\( L_0, L_1 \): the background and foreground layers
\( \alpha_i \): alpha map defining transparency of \( L_i \)
\( \dot{\mathbf{X}} \): linear velocity of a scene point \( \mathbf{X} \) (3-vector)
\( (\mathbf{X})_z \): \( z \)-axis component of the linear velocity \( \dot{\mathbf{X}} \)
\( \mathbf{t} \): translational velocity in 3-space (3-vector)
\( \omega \): angular velocity in 3-space (3-vector)
\( \ddot{\mathbf{x}} \): image motion vector (3-vector)
\( \dot{\mathbf{x}} \): translational component of \( \ddot{\mathbf{x}} \)
\( \dot{\mathbf{x}}_r \): rotational component of \( \ddot{\mathbf{x}} \)
\( \mathbf{J}_i(x) \): image Jacobian matrix at an image point \( x \) (3x6 matrix)
\( \mathbf{J}_{i, \mathbf{t}}(x) \): contribution of the relative translation velocity \( \mathbf{t} \) (3x3 matrix)
\( \mathbf{J}_{i, \omega}(x) \): contribution of the relative angular velocity \( \omega \) (3x3 matrix)
\( \Phi \): camera ego-motion vector (6-vector)
\( \Phi^* \): desired camera ego-motion vector
\( \Phi_k \): current approximation of \( \Phi^* \)
\( \delta \Phi \): differential camera ego-motion vector (6-vector)
\( \delta \Phi_k \): current approximation of the differential camera ego-motion vector
\( \mathbf{k} \): the unit vector along to the \( z \)-axis (3-vector)
\( \mathbf{J} \): accumulated image Jacobian matrix for \( N \) image points
\( 3 \times N \times 6 \) matrix
\( \mathbf{J}_k \): current approximation of the accumulated image Jacobian matrix
\( \mathbf{v} \): accumulated image motion vectors for \( N \) image points (3x3 vectors)
\( \delta \mathbf{v}_k \): current approximation of the accumulated image motion vectors
\( 3 \times N \) vectors
\( \delta \mathbf{x}_{ik} \): current approximation of the differential image motion vector
\( \mathbf{H}_k \): current approximation of the planar homography matrix (3x3 matrix)
$P'_c$: current approximation of the camera projection matrix ($3\times4$ matrix)
$R$: rotation matrix ($3\times3$ matrix)
$R_c$: current approximation of the rotation matrix
$\delta R$: differential rotation matrix ($3\times3$ matrix)
$\delta R_c$: current approximation of the differential rotation matrix
$t_i$: current approximation of the translation vector (3-vector)
d: differential translational velocity (3-vector)
d_c: current approximation of the differential translational velocity
$\delta$: differential angular velocity (3-vector)
$\delta_c$: current approximation of the differential angular velocity
$\varepsilon$: an error function
$\nabla I'(x_i)$: spatial image gradient vector (3-vector)
$\nabla I'(x_{i,c})$: current approximation of the spatial image gradient vector
$\delta I(x_i)$: temporal image difference
$\delta I(x_{i,c})$: current approximation of the temporal image difference
$H_I$: image Hessian matrix ($6\times6$ matrix)
$H_{I,c}$: current approximation of the image Hessian matrix
$m_i$: image mismatch vector (6-vector)
$m_{i,c}$: current approximation of the image mismatch vector
-- macro 있음

-- 까지: 지우지 말 것

1
CHAPTER 1

INTRODUCTION

1.1. Motivation

In visual navigation, the purpose of local map building is to solve the problem of “Where should I go?” and the coupled problem is localization often mentioned by the question of “Where am I?” Many traditional vision based methods can be classified into two major classes: First, stereo-based methods produce sparse or dense disparity maps with two or more images and they are constructed into 3D maps. Depth information obtained from 3D maps may provide 3D visual cues such as paths or obstacles for navigation [4,23]. And these methods are also used in incremental localization by detecting specific features such as corners or edges, called natural landmarks and tracking them in consecutive images. But they may fail when particular types of features are not supported in images. In this case, the accuracy of reconstructed 3D maps is decreased and the confidence of the map is not also guaranteed. Secondly, motion-based methods compute optical flows called the image motion field in consecutive images. Motion information is used for detecting interesting objects of dominant motions [19,21] and imitating the centering behaviors of honeybees within walls [18] and also time-to-contact estimation for navigation [3]. These methods are appropriate for detecting other static or moving objects but they do not work in some types of scenes such as walls having no dominant motions.

For mobile robots in structured environments, the ground floor is a very interesting object because it represents a local map of movable paths
except for other static or dynamic objects. In spite of the merit, it is not easy to be detected by image properties because common types of features, such as corners, edges, colors or textures, are not supported in ground floor images. In stereo approaches it cannot be found directly from 3D maps because of different depth. In motion approaches it is also difficult because of the motion is not dominant against the surroundings. Unlike these approaches, a geometric fact that other static or dynamic objects are placed on the ground floor perpendicularly can provide a clue to find the ground floor in images.

1.2. Research Goal

This paper considers visual navigation in structured environments such as indoor scenes and focus on a method to detect obstacles and paths within images simultaneously. Figure 1.1 illustrates the concept with an image of an indoor scene that contains static objects such as walls, desks and a book on the floor. The key idea in the figure is to detect static objects on the ground floor as obstacles (Figure 1.1b) and to detect the ground floor as a movable path that robots can navigate (Figure 1.1c).

In order to detect the ground floor in images, this paper exploits the geometric fact that other static or dynamic objects are placed on the ground floor perpendicularly. In order to use the fact in the proposed algorithm to detect the ground floor, it is assumed that an image consists of small patches and each small patch corresponds to a plane in 3-space so that it can define a plane with at least three image points. It is because at least three point correspondences in two or more images can define a plane in 3-space [7]. Such the geometric fact and assumptions provide the following advantages: (i) Plane normals can be an effective clue to separate the ground floor from the scene; (ii) To compute a plane normal for all image patches can increase the precision of the detected ground floor image. But
this approach contains the following problems: The first, known as correspondence problem, is determining which point in the first image corresponds to which point in the second image [22] and the second determining the shape and size of a patch so that a patch corresponds to a flat surface in 3-space. In order to solve the problems and satisfy the above geometric fact and assumptions, the following methods are proposed.

1. Compute image motion field from consecutive images to obtain dense image point correspondences instead of using stereo images providing only sparse point correspondences because a patch should be have at least three image point correspondences although it is small.
2. Adopt multi-scale coarse-to-fine estimation algorithm, such as Lucas-Kanade estimation algorithm, so that image point correspondences should be accurate as possible.

(a) an indoor scene    (b) obstacles
(c) the ground floor

Figure 1.1. A sample image of an indoor scene.
3. Split an image into sub-regions as small as possible by using image splitting techniques based on color homogeneity so that each region is close to a plane in 3-space.

4. Derive an optimimal estimate of the plane normal for a patch so that computation errors are minimized although mismatched image point correspondences are obtained.

5. Design an iterative refinement process based on region growing and merging to detect and segment the only ground floor within an image.

6. Represent the segmented image with two layers in a proper form for visual navigation.

General ego-motion model is represented by a function of image motion vectors and 3D information, but it cannot carry 3D information any more if a plane is known, such as the ground floor. In order to develop ego-motion estimation algorithms, this paper uses plane information provided in the ground floor detection algorithm. Given the ground floor images and the image motion vectors on the ground floor the ego-motion model can be computed by at least three image motion vectors. In this paper, the following methods are proposed.

1. The first proposed algorithm is the inverse Jacobian method which is derived from the iterative Newton-Raphson formula based on image Jacobian and a least squared solution of the ego-motion parameters is computed.

2. The second proposed algorithm is the image Gradient method computing an optimimal weighted least squared estimate to determine the ego-motion parameters robustly so that it minimizes computation errors although inaccurate and noisy image motion vector are obtained.
1.3. Outline of the Thesis

The remainder of the paper is organized as follows: Chapter 2 introduces some terminologies and basic concepts for developing the proposed ground floor detection and ego-motion estimation algorithms. Chapter 3 presents the proposed ground floor detection algorithm in more detail, shows the experimental results with two real scenes and compares them with the ground truth data. Chapter 4 presents the proposed ego-motion estimation algorithm, shows the simulation and real experimental results. Chapter 5 concludes the thesis with the validity for visual navigation of mobile robots, and the effectiveness of the proposed algorithms.
CHAPTER 2

PRELIMINARY

This chapter introduces some terminologies and basic concepts for developing the ground floor detection and ego-motion estimation algorithms. First, virtual image plane and normalized camera model is presented. Second, basic concepts for image motion estimation are introduced briefly.

2.1. Camera Model

This section describes the mathematical model of projective cameras, which represents a mapping between the 3D scene space and a 2D image plane. It can be considered as the central projection of points in space onto a plane. Although the model has been derived by a number of authors [2,7,22,25], it is rewritten in an appropriate form to develop the proposed algorithms, which is considered in the normalized image plane.

2.1.1 Normalized Camera Model

The basic pinhole camera model is considered as the central projection of points in space onto a plane. Figure 2.1 shows a virtual image plane $\pi$ at a focal length $f = 1$ in the camera coordinate system $\{C\}$. Let $X = (X,Y,Z)^T$ be a scene point in 3-space and $x = (x,y,l)^T$ its corresponding point in the virtual image plane $\pi$. Then the central projection of a scene point $X$ in the virtual image plane is defined by
where the term \( \text{proj}_p(.) \) denotes a projection operator which project a scene point onto a plane \( p \) and the image point \( x \) in the image plane \( \pi \) is given by

\[
\begin{bmatrix}
x \\
y \\
1 
\end{bmatrix} = \frac{1}{Z} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}.
\]

(2.2)

If the world and image points are represented by using homogeneous coordinates, then the central projection is expressed as a matrix-vector form and the above equation may be written as

\[
\begin{bmatrix}
sx \\ sy \\ s 
\end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 
\end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 
\end{bmatrix}
\]

(2.3)

where \( s \) is a scale factor between the homogeneous coordinates and non-homogeneous coordinates. If \( \tilde{X} = (X,Y,Z,1)^T \) represents the homogeneous coordinates of \( X \), the above equation becomes a compact form as

\[
X = \frac{1}{f} \begin{bmatrix} X \\ Y \\ Z \\ 1 
\end{bmatrix} = \frac{1}{f} \tilde{X}
\]

Figure 2.1. Projection onto the normalized image plane \( \pi \) at \( f = 1 \).
where the matrix \( P = [I | 0] \) is a camera projection matrix with respect to the virtual image plane \( \pi \) at a focal length \( f = 1 \). This is called the \textit{normalized camera projection matrix} because it does not carry camera information.

The center of projection is called the \textit{camera center}. The line from the camera center perpendicular to the image plane is called the \textit{principle axis} and the point where the principle axis meets the image plane is called the \textit{principle point}.

### 2.1.2 General Camera Model

Now consider a real image plane not a virtual image plane. Figure 2.2 shows an image point \( p = (u, v)^T \) that is projection of a scene point \( X \) onto a real image plane \( I \) at a focal length \( f \neq 1 \).

By considering the focal length, Eq. (2.2) is written as

\[
sx = P\bar{X}
\]
\[
\begin{pmatrix}
  u \\
  v \\
\end{pmatrix} = \frac{f}{Z} \begin{pmatrix}
  X \\
  Y \\
\end{pmatrix}
\] 
(2.5)

and Eq. (2.3) becomes

\[
\begin{pmatrix}
  su \\
  sv \\
  s \\
\end{pmatrix} = \begin{bmatrix}
  f & 0 & 0 & 0 \\
  0 & f & 0 & 0 \\
  0 & 0 & 1 & 0 \\
\end{bmatrix} \begin{pmatrix}
  X \\
  Y \\
  Z \\
  1 \\
\end{pmatrix}.
\] 
(2.6)

If the image point \( \mathbf{p} \) is represented by the homogeneous coordinate \( \tilde{\mathbf{p}} = (u, v, 1)^\top \), then Eq. (2.4) has the concise form

\[
SP = diag(f, f, 1)P\tilde{X}.
\] 
(2.7)

In the above equation, the focal length is represented as a distinct term from the normalized camera projection matrix because the focal length has different values according to different camera.

In the case of CCD cameras, there are additional parameters related to the specification of cameras. The pinhole camera model just derived assumes that the origin of the image coordinates is located at the principle point, but the image point \( \mathbf{p} \) has an offset corresponding to the principle point. If \( (p_x, p_y) \) denotes the position of the principle point in the image coordinate, Eq. (2.6) becomes

\[
\begin{pmatrix}
  su \\
  sv \\
  s \\
\end{pmatrix} = \begin{bmatrix}
  f & 0 & p_x & 0 \\
  0 & f & p_y & 0 \\
  0 & 0 & 1 & 0 \\
\end{bmatrix} \begin{pmatrix}
  X \\
  Y \\
  Z \\
  1 \\
\end{pmatrix}.
\] 
(2.8)

The image coordinates is measured in pixel unit. Thus conversion factors between pixel and world coordinates must be considered. If the
number of pixels per unit distance in image coordinates are \( m_x \) and \( m_y \) in the \( x \) and \( y \) directions, then the transformation from world coordinates to pixel coordinates is obtained by multiplying \( \text{diag}(m_x, m_y, 1) \) on the left of the above equation:

\[
\begin{pmatrix}
    s u \\
    s v \\
    s
\end{pmatrix} =
\begin{pmatrix}
    f m_x & 0 & p_x m_x & 0 \\
    0 & f m_y & p_y m_y & 0 \\
    0 & 0 & 1 & 0
\end{pmatrix}
\begin{pmatrix}
    X \\
    Y \\
    Z \\
    1
\end{pmatrix}
\]

(2.9)

and the above equation is simply

\[
\tilde{s}\tilde{p} = K\tilde{X}
\]

(2.10)

where

\[
K =
\begin{pmatrix}
    f m_x & 0 & p_x m_x & 0 \\
    0 & f m_y & p_y m_y & 0 \\
    0 & 0 & 1 & 0
\end{pmatrix} =
\begin{pmatrix}
    \alpha_x & 0 & x_0 & 0 \\
    0 & \alpha_y & y_0 & 0 \\
    0 & 0 & 1 & 0
\end{pmatrix}
\]

(2.11)

The term \( K \) is a matrix containing internal parameters depending on camera specifications, called the \textit{camera calibration matrix} and the terms \( \alpha_x, \alpha_y, x_0 \) and \( y_0 \) are represented in pixel coordinates.

The conversion from image coordinates to the normalized image coordinates is given by

\[
x = K^{-1}\tilde{p}.
\]

(2.12)

The above equation can be useful to eliminate the effect of camera internal parameters when detecting the ground floor and estimating the ego-motion.
2.1.3 Camera Motion in 3-space

Camera motion (translation and rotation) in 3-space changes the form of the camera projection matrix and it can be obtained by using two cameras with different positions and orientations. Figure 2.3 shows two normalized cameras. Suppose that two frames \{0\} and \{1\} are the coordinates of the first and second cameras with the normalized camera projection matrices \( P_0 = [I \mid 0] \) and \( P_1 = [I \mid 0] \), respectively. If the position of a scene point \( X \) is represented by \( ^0X \) and \( ^1X \) with respect to each frame \{0\} and \{1\}, the corresponding image points are given by

\[
\begin{align*}
^0sX_0 &= P_0 \cdot ^0X = [I \mid 0] \cdot ^0X, \\
^1sX_1 &= P_1 \cdot ^1X = [I \mid 0] \cdot ^1X.
\end{align*}
\]

The relation between \( ^0X \) and \( ^1X \) is represented by a 4×4 homogeneous transformation matrix consisting of rotation and translation between two coordinates:

![Figure 2.3. Two normalized cameras in 3-space.](image-url)
\[ ^0\tilde{X} = ^0T_1^1\tilde{X} = \begin{bmatrix} ^0R_1 & ^0t_1 \\ 0^T & 1 \end{bmatrix}^1\tilde{X} \]  

(2.14)

where \(^0T_1\) is \(4\times4\) homogeneous transformation matrix which represents the second frame \(\{1\}\) with respect to the first frame \(\{0\}\), the matrix \(^0R_1\) represents the orientation of the second frame \(\{1\}\) with respect to the first frame \(\{0\}\) and \(^0t_1\) is the position of the second frame \(\{1\}\) expressed in the first frame \(\{0\}\). By substituting the above equation in Eq. (2.13), the central projection of \(^0X\) in the first frame \(\{0\}\) with respect to the second frame \(\{1\}\) is

\[
^sx_i = [I \mid 0](^0T_1)^{-1} ^0\tilde{X} \\
= [(^0R_1)^T \mid -(^0R_1)^T \ 0^T_1] \ 0\tilde{X} \\
(2.15)
\]

Since the above equation is represented in the same coordinates \(\{0\}\), the subscript can be removed. The 3D motion of a camera may be written with the above equation:

\[
^sx_i = P_i \ 0\tilde{X} = [R^T \mid -R^T t] \ 0\tilde{X} \\
(2.16)
\]

where the matrix \(P_i\) is the normalized camera projection matrix and the point \(x_i\) is the normalized image point of a scene point \(X\) when a camera is rotated and translated by \(R\) and \(t\).

### 2.2. Image Motion Estimation

**Image motion field** is defined as the projection of the 3D velocity field on the image plane in view of the view geometry and it is also defined as the 2D vector field of the image correspondences in image sequences, induced
by the relative motion between the camera and the scene. Image motion estimation is to determine the motion field observed in the image plane, but the image motion field cannot really be observed. Instead, the spatial and temporal variations of the image brightness can be estimated, called the \textit{optical flow field}. The optical flow field is an approximation of the image motion field, but they are not same. Figure 2.4 shows the difference between them. Consider a smooth, lambertian and uniform sphere rotating around a diameter in front of a camera as shown in Figure 2.4a. In this case, the image motion field is not zero because the points on the sphere are moving, but the optical flow field is zero because there are not any moving patterns in the images. Consider a still, smooth, specular and uniform sphere in front of a camera and a moving light source as shown in Figure 2.4b. In this case, the image motion field is zero since the points on the sphere are not moving, but the optical flow field is not zero since there are moving patterns in the images.

In order to estimate the optical flow field, it is basically assumed that

![Figure 2.4. Difference between the image motion and the optical flow.](image)
the brightness of moving objects remain constant, called the *image brightness constancy assumption* [22]. And methods to obtain the optical flow are classified into gradient-based methods and filter-based methods. Gradients-based methods compute optical flow from spatial and temporal derivatives of image intensity [9,12,15] Filter based methods computes optical flow in the frequency domain [26,8,6,5]. For obtaining accurate optical flow fields, multi-scale, coarse-to-fine, refinement techniques are also used with them.

### 2.2.1 Image Brightness Constancy Equation

Image brightness constancy equation is a fundamental equation to obtain optical flow in the gradient-based methods. Consider a moving point in 3-space. Let $\mathbf{P} = (X(t), Y(t), Z(t))^T$ be a moving scene point at time $t$ and $\mathbf{p} = (x(t), y(t))^T$ be its image point at time $t$. If $E(x(t), y(t), t)$ represents the brightness at $\mathbf{p}$ at time $t$, the brightness constancy assumption is written as

$$\frac{dE}{dt} = 0 \quad (2.17)$$

and the total temporal derivative may be written by the chain rule of differentiation:

$$\frac{dE}{dt} = \frac{\partial E}{\partial x} \frac{dx}{dt} + \frac{\partial E}{\partial y} \frac{dy}{dt} + \frac{\partial E}{\partial t} = 0 \quad (2.18)$$

Denote

$$\nabla E = \left[ \frac{\partial E}{\partial x} \quad \frac{\partial E}{\partial y} \right]^T \quad (2.19)$$
is the spatial image gradient,

$$\mathbf{v} = \begin{bmatrix} \frac{dx}{dt} & \frac{dy}{dt} \end{bmatrix}^T$$  \hspace{1cm} (2.20)

is the optical flow and

$$E_t = \frac{\partial E}{\partial t}$$  \hspace{1cm} (2.21)

is the temporal image difference. Then Eq. (2.18) becomes a compact form

$$\nabla E^T \mathbf{v} + E_t = 0.$$  \hspace{1cm} (2.22)

Methods to obtain the optical flow $\mathbf{v}$ can be divided two major classes: differential techniques and matching techniques. Differential techniques use least squared or weight least squared estimations derived from the brightness constancy equation based on image brightness constancy assumption and these methods assumes that the optical flow $\mathbf{v}$ is well approximated by a constant vector within any small region of the image plane [22]. Matching techniques use correlation or block-matching methods in which each small patch of the first frame is compared with nearby patches in the next frame [12]. In matching techniques, two key components must be considered: accuracy and robustness. Accuracy is related to the local sub-pixel accuracy and robustness is related to sensitivity with respect to light change, size of image motion, etc. Small window is preferable at occluding areas and it doesn’t smooth out image values but it cannot handle large motions. Thus it is required a tradeoff between local accuracy and robustness when selecting window size. As a solution to the problem, a pyramidal implementation can handle large motions with small window.
CHAPTER 3

GROUND FLOOR DETECTION

This chapter proposes the ground floor detection algorithm estimating plane normals based on planar homography, shows the experimental results with two real scenes and compares them with the ground truth data.

3.1. Planar Homography

Consider a plane in a scene and suppose the plane is projected into two views. In this case, the relation between the images projected from the scene plane is a projective relation, called *planar homography* induced by a plane [7]. Figure 3.1 illustrates the concept with two images. In the figure the three regions between two images correspond three different planes in

![Figure 3.1. Planar homography induced by a plane.](image-url)
3-space and each region has own planar homography induced by its corresponding scene plane.

A homography is also called a projectivity, or a collineation, or a projective transformation [7]. A homography shown in Figure 3.2 is defined by an invertible mapping \( h \) from a 2D projective space \( \mathbb{P}^2 \) to itself such that three points \( x_1, x_2 \) and \( x_3 \) lie on a line \( l \) if and only if \( h(x_1), h(x_2) \) and \( h(x_3) \) lie on another line \( l' \). In other words, a mapping \( h \) is a homography if and only if there exists a non-singular \( 3 \times 3 \) matrix \( H \) such that for any point in a 2D projective space \( \mathbb{P}^2 \) represented by a vector \( x \), it is true that \( h(x) = Hx \).

Thus any invertible linear transformation of homogeneous coordinates is a homography and a planar homography is a linear transformation on homogeneous 3-vectors represented by a non-singular \( 3 \times 3 \) matrix:

\[
\begin{bmatrix}
  x' \\
  y' \\
  1
\end{bmatrix} =
\begin{bmatrix}
  h_1 & h_2 & h_3 \\
  h_4 & h_5 & h_6 \\
  h_7 & h_8 & 1
\end{bmatrix}
\begin{bmatrix}
  x \\
  y \\
  1
\end{bmatrix}.
\]

This planar projective transformation is simply a linear transformation of \( \mathbb{R}^3 - (0,0,0) \) and it has 8 degree of freedom. While, shown in Figure 3.3,
consecutive image planes has central projection mapping points to points and also lines to lines and it is called a *central projectivity* or a *perspectivity*. It has 6 degree of freedom. Thus a planar projective transformation can be specified by four point correspondences, or by two corresponding lines. While a perspectivity can be specified by three image point correspondences, or by one image point correspondence and one corresponding line.

### 3.2. Plane Normal Computation

Ground floor is a very interesting object for mobile robot navigation in structure environments, which represents movable paths except for other static or dynamic objects. Since such objects are on the ground floor perpendicularly, plane normals can be a clue to separate the ground floor from the scene.

Consider a plane in a scene and suppose the plane is projected onto two views. Then the relation between two images which is projection of the

![Figure 3.3. Central projection.](image-url)
scene plane is a planar homography induced by a plane. Figure 3.4 shows a scene point \( \mathbf{X} \) on a scene plane \( \Pi \) in 3-space and its two normalized image points \( \mathbf{x} \) and \( \mathbf{x}' \) which are projected onto the normalized image planes \( \pi \) and \( \pi' \). Let \( \mathbf{P} = [\mathbf{I} \mid 0] \) and \( \mathbf{P}' = [\mathbf{M} \mid \mathbf{m}] \) be the first and second normalized camera projection matrices. Using the normalized pinhole camera model, the image points \( \mathbf{x} \) and \( \mathbf{x}' \) are

\[
\begin{align*}
sx &= \mathbf{P}\tilde{\mathbf{X}} = [\mathbf{I} \mid 0]\tilde{\mathbf{X}} \\
\mathbf{s}x' &= \mathbf{P}'\tilde{\mathbf{X}} = [\mathbf{M} \mid \mathbf{m}]\tilde{\mathbf{X}}
\end{align*}
\]

(3.2)

where the matrices \( \mathbf{M} \) and \( \mathbf{m} \) are sub-matrices of the second camera projection matrix in a canonical form.

For the first view \( \pi \), the scene point \( \mathbf{X} \) is on the ray passing through its image point \( \mathbf{x} \) and the camera center \( \mathbf{c} \):

\[
\mathbf{X}(\lambda) = (1 - \lambda)\mathbf{c} + \lambda\mathbf{x}
\]

(3.3)
Since the location of the first camera center \( c \) in the first view is

\[
P \tilde{c} = [I | 0] \begin{bmatrix} c \\ 1 \end{bmatrix} = 0
\]  

(3.4)

the camera center \( c \) is given by

\[
c = 0.
\]  

(3.5)

From Eq. (3.3) the scene point \( X \) becomes

\[
X(\lambda) = (1 - \lambda)c + \lambda x = \lambda x
\]  

(3.6)

and it can be written in homogeneous coordinates

\[
\tilde{X} = \begin{bmatrix} x \\ 1/\lambda \end{bmatrix}.
\]  

(3.7)

If the scene plane \( \Pi \) is represented by \( \Pi = [n^T \ | \ d]^T \) in 3-space, the scene point \( X \) on the scene plane \( \Pi \) satisfies

\[
\Pi^T \tilde{X} = (v^T \ 1) \begin{bmatrix} x \\ 1/\lambda \end{bmatrix} = 0
\]  

(3.8)

where the vector \( v \) is the plane normal of the scene plane \( \Pi \) and it is parameterized by \( n/d \). Thus the scalar value is given by

\[
\lambda = \frac{1}{-v^T x}
\]  

(3.9)

and the scene point \( X \) may be rewritten as

\[
\tilde{X} = \begin{bmatrix} x \\ -v^T x \end{bmatrix}.
\]  

(3.10)
For the second view \( \pi' \), the normalized image point \( x' \) is obtained from Eq. (3.2) and (3.10):

\[
x' = P'\tilde{X} = [M | m] \begin{pmatrix} x \\ -v^T x \end{pmatrix} = (M - mv^T)x
\]

(3.11)

Since the relation between two image points \( x \) and \( x' \) is a planar homography as in Eq. (3.1) and the homography matrix is defined up to scale, the 3×3 homography matrix \( H \) becomes

\[
H = M - mv^T.
\]

(3.12)

The above equation represents that the homography matrix can be specified with the matrix \( M \) and two vectors \( m, v \).

### 3.2.1 Case I: Three Image Point Correspondences

A scene in 3-space can be specified by three image point correspondences. Refer to Figure 3.5. Suppose that three image point correspondences are given between two views. Then the homography induced by the plane of the three image points is

\[
x_i \leftrightarrow x'_i = Hx_i \quad \text{for} \quad i = 1, \ldots, 3 \quad \text{where} \quad x_i \in \pi, x'_i \in \pi'.
\]

(3.13)

Since the left and right terms \( x'_i \) and \( Hx_i \) are parallel their cross product should be zero:

\[
x'_i \times Hx_i = x'_i \times (M - mv^T)x_i = 0.
\]

(3.14)
In order to compute the plane normal, making a linear equation with respect to $v$ yields

$$(x_i' \times m)(v^T x_i) = x_i' \times Mx_i \quad (3.15)$$

and it is written in a compact form as

$$x_i^T v = b_i \quad (3.16)$$

where

$$b_i = \frac{(x_i' \times m)^T (x_i' \times Mx_i)}{(x_i' \times m)^T (x_i' \times m)}. \quad (3.17)$$

Summing up the above equation for three image point correspondences yields a linear equation with respect to the plane normal:

Figure 3.5. A scene plane normal with 3 image points.
Thus the plane normal $\mathbf{v}$ can be computed directly from three image point correspondences.

Note that if the matrix $\mathbf{A}$ is not of full rank, a plane normal cannot be obtained because three image points $\mathbf{x}_i$ are collinear. The accuracy of the plane normal $\mathbf{v}$ depends on the accuracy of the three image point correspondences.

Two image points $\mathbf{x}_i$ and $\mathbf{x}_i'$ for two views are assumed to be in the normalized image coordinates because the camera projection matrices are $\mathbf{P} = [\mathbf{I} | \mathbf{0}]$ and $\mathbf{P}' = [\mathbf{M} | \mathbf{m}]$. But it is necessary to consider their pixel coordinates for plane normal computation. If two image points $\mathbf{p}_i$ and $\mathbf{p}_i'$ are represented in pixel coordinates, their normalized image coordinates $\mathbf{x}_i$ and $\mathbf{x}_i'$ are obtained by using camera calibration matrices $\mathbf{K}$ and $\mathbf{K}'$:

$$
\begin{align*}
\mathbf{x}_i' &= \mathbf{x}_i - \mathbf{b}_1 \\
\mathbf{x}_i' &= \mathbf{x}_i - \mathbf{b}_2 \\
\mathbf{x}_i' &= \mathbf{x}_i - \mathbf{b}_3
\end{align*}
\quad \Leftrightarrow \quad \mathbf{A} \mathbf{v} = \mathbf{b}.
\qquad \text{(3.18)}
$$

Geometric meaning of $\mathbf{x}_i' \times \mathbf{m}$

Let $\mathbf{e}$ and $\mathbf{e}'$ be the epipoles of the first and second views as shown in Figure 3.6. The epipole $\mathbf{e}'$ of the second view is defined by an image point that is projection of the first camera center $\mathbf{c}$ onto the second view:

$$
\mathbf{e}' = \mathbf{P}' \mathbf{c}.
\qquad \text{(3.20)}
$$
Since the normalized projection matrix $P'$ of the second camera is $P'=[M|m]$ and the center of the first camera is zero from Eq. (3.5), the above equation becomes

$$e'=m.$$

The above equation means that the right most column vector $m$ of the second camera projection matrix corresponds to the epipole $e'$ of the second camera. Thus the term $x'_i \times m$ may be written as

$$x'_i \times m = x'_i \times e' = l'_i$$

where the line $l'_i$ is the epipolar line of the image point $x'_i$ in the second view. Eq. (3.17) is rewritten as

$$b_i = (x'_i \times Mx_i)^T l'_i / \|l'_i\|.$$ (3.23)

![Figure 3.6. Geometric meaning of $x'_i \times m$.](image-url)
3.2.2 Case II: Three or More Noisy Point Correspondences

The accuracy of the plane normal $\mathbf{v}$ depends on the accuracy of computing the corresponding image point in another image for a point in the reference image. Although the multi-scale coarse-to-fine estimation algorithm produces accurate and dense image point correspondences, less-matched image points may exist in case of large motion vectors and mismatched image points may also exist in case of the apparent brightness changes are not observed in images. For these cases it is necessary to compute an optimimal plane normal which minimizes computation error.

Assume that a region $R$ is the sub-image that is projection of a scene plane in 3-space as shown in Figure 3.7. An appropriate error function is the sum of squared difference of the residue for a plane normal $\mathbf{v}$ in Eq.(3.16) for all pixels in that region $R$.

![Figure 3.7. A scene plane normal with $n \geq 3$ noisy image points.](image-url)
\[ e(v) = \sum_{x_i \in R} \left( x_i^T v - b_i \right)^2. \]  

(3.24)

At the optimimum plane normal \( v^* \), the partial derivatives of \( e \) with respect to \( v \) should be zero:

\[ \frac{\partial e(v)}{\partial v} \bigg|_{v=v^*} = 0. \]  

(3.25)

After expansion of the derivatives, Eq. (3.24) becomes

\[ \frac{1}{2} \left( \frac{\partial e(v)}{\partial v} \right) = \sum_{x_i \in R} (x_i x_i^T) v - \sum_{x_i \in R} (x_i b_i). \]  

(3.26)

Denote

\[ G = \sum_{x_i \in R} (x_i x_i^T) \]  

(3.27)

\[ b = \sum_{x_i \in R} (x_i b_i). \]  

(3.28)

Then Eq. (3.25) is written as

\[ \left. \frac{1}{2} \left( \frac{\partial e(v)}{\partial v} \right) \right|_{v=v^*} = Gv^* - b = 0. \]  

(3.29)

Therefore the optimimum plane normal \( v^* \) is given by

\[ v^* = G^{-1} b. \]  

(3.30)

Note that the optimal estimate of a plane normal which minimizes the error is a least squared solution and it is valid when the matrix \( G \) is invertible.
3.3. Layered Ground Floor Segmentation

This section describes the framework to segment the ground floor in image sequences. As mentioned earlier, it is exploited the fact that other static or moving objects are on the ground floor and they are perpendicular to the ground. Also it is assumed that an image consists of small patches (or regions) and each corresponds to a plane in 3-space and has at least three image points so that it can define a plane. It is because at least three point correspondences in two or more images can define a plane in 3-space. Such a fact and assumptions provide the following advantages: (i) Plane normals can be an effective clue to separate the ground floor from the scene; (ii) To compute a plane normal for all image points in a patch increases its computational accuracy.

But this approach contains the following problems: The first, known as correspondence problem, is determining which point in the first image corresponds to which point in the second image [23]. The second problem is determining the shape and size of a patch so that a patch corresponds to a flat region in 3-space.

In order to solve the problems and satisfy the above fact and assumptions, the following method is suggested: (i) A patch should have at least three point correspondences although it is small. Thus image motion field is computed to produce dense point correspondences instead of using stereo images providing only sparse point correspondences; (ii) Point correspondences should be accurate as possible to provide accurate plane normals. Thus the multi-scale coarse-to-fine estimation is adopted, such as Lucas-Kanade optical flow estimation [12]; (iii) Each patch should be closest to a plane in 3-space. Thus an image is split into sub-regions as small as possible by using image splitting techniques based on color homogeneity [17]; (iv) Estimated plane normals should be optimimal with respect to point correspondence errors. Thus the optimal estimate of the plane normal, which is just derived in the previous section, is used so that the estimation...
error is minimized although mismatched image point correspondences are obtained, (v) The patches corresponding to the ground floor should be refined within images. Thus an iterative refinement process is designed to detect and segment the ground floor by using region growing and merging techniques, (vi) The ground floor image should be represented in a proper form for visual navigation. Thus the segmented images is represented with two layers [24,10].

The proposed algorithm consists of three stages as shown in Figure 3.8:

1. Optical flow estimation and image segmentation: Compute optical flow to obtain accurate and dense image point correspondences in consecutive images by using multi-scale coarse-to-fine estimation. Split images into small regions.

2. Iterative refinement process: Select a seed region and grow it to connected regions using a Queue structure. For each region, estimate the plane normal and merge it into the ground floor if it is close to the ground plane normal.

3. Two-layered representation: Decompose the image into the foreground layer and the background layer.

Figure 3.8. Framework of the layered ground floor segmentation.
3.3.1 Optical Flow Estimation

The performance of detecting the ground floor is mainly affected by both accuracy and density of image point correspondences. In stereo approaches accurate point correspondences can be only obtained in the particular types of features such as corners, edges, etc. Since large disparity may produce many mismatched points it is not guaranteed that patches should be have at least three image point correspondence although it is small. In motion approaches, accurate and dense point correspondences can be obtained with multi-scale coarse-to-fine estimation algorithms based on gradient approaches [12,1]. Traditional algorithms produce only motion vectors, known as optical flows, which represent the directions to the corresponding points in the next image. But the vector can represent the location of the corresponding points in the next image using multi-scale coarse-to-fine estimation algorithms because they can handle large motions. Figure 3.9 shows the multi-scale coarse-to-fine motion estimation algorithm based on the Gaussian pyramid using the Lucas-Kanade equation [12].

![Gaussian pyramid diagram](image)

**Figure 3.9.** Multi-scale coarse-to-fine estimation.
3.3.2 Iterative Refinement Process

In the iterative refinement process, region growing and merging rules are used with a classifier, that is, angle difference between an estimated plane normal of a region and the plane normal of the ground floor. The classifier decides whether a plane normal is similar to the ground plane normal:

\[
\theta(v_i, v_G) = \cos^{-1} \left( \frac{v_i^T v_G}{\|v_i\| \|v_G\|} \right)
\]  (3.31)

where \( v_i \) is the estimate of the plane normal for a region \( R_i \) and \( v_G \) is the plane normal of the ground floor.

The merging rule is that a region \( R_i \) is merged with the ground floor region \( R_G \) if the angle difference is smaller than a threshold value:

\[
R_G = R_i \cup R_G \quad \text{if} \quad \theta(v_i, v_G) < \tau
\]  (3.32)

where \( \tau \) is a threshold for angle difference.

In the growing rule, when a region \( R_i \) is merged into the ground floor region \( R_G \), new regions connected with \( R_i \) are found. This process is repeated using a Queue structure until it converges. After the iterative refinement using region growing and merging via classification, an image is divided into the ground floor and the rest.
3.3.3 Layered Image Representation

In order to represent the segmented images in a proper form for visual navigation, we decompose the image into two layers, described by [10]. The first layer is the foreground layer representing the ground floor as a path that robots can navigate and the second layer is the background layer representing other static or moving objects as obstacles:

\[ I(x, y) = L_0(x, y)(1 - \alpha_1(x, y)) + L_1(x, y)\alpha_1(x, y) \]  (3.33)

where \( L_0 \) and \( L_1 \) are background and foreground layers, respectively and \( \alpha_1 \) is an alpha map defining transparency of \( L_1 \).
3.4. Experimental Results

This section describes the experimental results for two image sequences without obstacles and with static obstacles, respectively, as shown in Figure 3.10. Each image corresponds to the first frame of two image sequences when a robot moves forward on the ground floor.

(a) no obstacles

(b) static obstacles

Figure 3.10. Reference images for two image sequences.
3.4.1 Image Motion Estimation

Figure 3.11 shows the motion field that was estimated for all pixels, but appeared at every 5 pixels. For image motion estimation three level Gaussian pyramids and coarse-to-fine algorithms were used. Note that any kind of dominant motion does not appear and wrong motions appeared in the figures.

(a) no obstacles

(b) static obstacles

Figure 3.11. Image motion estimation.
3.4.2 Image Segmentation

Figure 3.12 shows the image segmentation result in which the image was split into a number of regions as small as possible to satisfy our assumptions. Almost 450 regions were detected and each region was represented by a non-overlapped unique color for visualization. Note that the ground floor image is split into several regions.

Figure 3.12. Image segmentation.
3.4.3 Plane Normal Computation

In order to show that the plane normal is an effective measure to detect the ground floor, Figure 3.13 shows the result of plane normal computation without iterative refinement process. The right color indices represent the angle between the ground floor and a color region. The blue colors indicate the corresponding regions are close to the ground floor and the red colors indicate the corresponding regions are perpendicular to the ground floor.

![Figure 3.13. Plane normal computation.](image)

(a) no obstacles

(b) static obstacles

Figure 3.13. Plane normal computation.
3.4.4 Iterative Refinement Process

Figure 3.14 shows the result of iterative refinement process to detect the only ground floor image. The region growing and merging rules were used with a classifier, that is, angle difference between an estimated plane normal of a region and the plane normal of the ground floor. Figure 3.15 and Figure 3.16 shows intermediate processes to detect the ground floor.

(a) no obstacles

(b) static obstacles

Figure 3.14. Iterative refinement process.
Figure 3.15. Intermediate processes for the case of no obstacles.
Figure 3.16. Intermediate processes for the case of static obstacles.
3.4.5 Layered Image Representation

Figure 3.17 shows the foreground layers of layered image representation, which correspond to the ground floor. Note that the only ground floor was segmented except for the other static objects perpendicular to the ground in images.

(a) no obstacles

(b) static obstacles

Figure 3.17. The foreground layers.
3.4.6 Comparison

Figure 3.18 shows the ground truth layers and the estimated layers generated by the proposed algorithm. The ground truth layers were produced by using image processing packages such as Adobe Photoshop or Paint Shop Pro. As we can see, the estimated foreground layer contained the ground floor and the estimated background layer contained the other static objects such as walls, desks, a chair, etc. and an static obstacle on the floor. The wrong parts in each layer were caused by mismatched image point correspondences in motion estimation and the missing parts were due to missing image point correspondences.

![Figure 3.18. Ground truth layers vs. estimated layers.](image)

(a) no obstacles  (b) static obstacles
3.5. Summary

The proposed ground floor detection algorithm exploited the geometric fact and the assumptions as mentioned in chapter 1. In order to specify a plane in 3-space, planar homography induced by a plane was considered and two methods to compute plane normal were derived, direct estimation with three image point correspondences and optiminimal estimation with noisy three or more image point correspondences. As a result a robust plane normal estimation algorithm that minimizes the computation error was developed although mismatched image point correspondences were obtained.

In order to show the validity of the proposed algorithm to detect the ground floor, the multi-scale coarse-to-fine estimation method was adopted to obtain accurate and dense point correspondences, the iterative refinement process was designed to segment the ground floor from the scene and the layered image representation was used for describing the ground floor image in a proper form for visual navigation. As a result, the proposed algorithm detected the only ground floor although mismatched image point correspondences were obtained in real experiments.
CHAPTER 4

CAMERA EGO-MOTION ESTIMATION

This chapter proposes two ego-motion estimation algorithms induced by a plane in 3-space and verifies the performance by experiments on synthetic data and real scenes.

4.1. Image Motion Model

This section describes a mathematical model of the image motion field which represents image changes observed in time-varying image sequences caused by the relative 3D motion between a scene and the camera. An image sequence is a series of images acquired at discrete time instants with fixed time intervals. And the relative 3D motion between a scene and a camera is caused by a moving camera in a static scene, or moving objects in front of a fixed camera, or both the camera and objects having different motions. Thus, the mathematical model of the image motion field represents a mapping between the 3D motion of a camera (or a robot) relative to a scene, and it represents the image changes observed in the imaging sensor plane. Also it can be thought of as the projection of 3D motion on the image plane.

Figure 4.1 shows the projection of the 3D motion of a scene point to a normalized image plane. Let \( \mathbf{X} = [X, Y, Z]^T \) be a scene point in 3-space. The linear velocity of the point \( \mathbf{X} \) is a function of the translational and rotational velocities of the point relative to the camera:

\[
\dot{\mathbf{X}} = \mathbf{t} + \mathbf{\omega} \times \mathbf{X} \quad (4.1)
\]
Figure 4.1. Projection of 3D motion of onto the normalized image.

where $t = [t_x, t_y, t_z]^T$ is the translational velocity of the point $X$ with respect to the camera and $\omega = [\omega_x, \omega_y, \omega_z]^T$ is the angular velocity of the point.

The normalized pinhole camera projects the scene point $X$ onto the normalized image plane at the focal length $f = 1$ as in Eq. (2.2):

$$x = \frac{1}{Z}X$$

(4.2)

where $x = [x, y, 1]^T$ denotes the image of the scene point $X$. Taking time derivative of the image point $x$ in the above equation leads to the image motion vector of the point $X$:

$$\dot{x} = \frac{1}{Z^2}(\dot{X}Z - X\dot{Z}).$$

(4.3)

By substituting Eq. (4.1) in the above equation and writing $\dot{Z}$ as $\dot{Z} = (\dot{X})_z$ to represent the $z$-axis component of the linear velocity $\dot{X}$, the image motion vector $\dot{x}$ becomes
\[
\dot{\mathbf{x}} = \frac{1}{Z} \mathbf{X} - \frac{1}{Z^2} \mathbf{X} \dot{\mathbf{X}}_z
\]

\[
= \frac{1}{Z} (\mathbf{t} + \omega \times \mathbf{X}) - \frac{1}{Z^2} \mathbf{X} (\mathbf{t} + \omega \times \mathbf{X})_z
\]

(4.4)

and by writing \( \mathbf{X} = Z \mathbf{x} \) to eliminate \( \mathbf{X} \), the mathematical model of the image motion vector can be rewritten as

\[
\dot{\mathbf{x}} = \frac{1}{Z} \mathbf{t} - \frac{1}{Z} \mathbf{x}(t)_z + \omega \times \mathbf{x} - \mathbf{x} (\omega \times \mathbf{x})_z
\]

(4.5)

where both \( (\mathbf{t})_z = t_z \) and \( (\omega \times \mathbf{x})_z = (\omega_x y - \omega_y x) \) are scalars. The image motion vector is called the image motion field.

Eq. (4.5) discloses two important properties of the image motion field: First, the motion field is a function of the translational and angular velocities of the camera with respect to the scene and depth. And it is represented as the sum of two components, one of which depends on the translational velocity and the other depends on the rotational velocity. Thus the image motion field may be written as

\[
\dot{\mathbf{x}} = \dot{\mathbf{x}}_t + \dot{\mathbf{x}}_r
\]

(4.6)

where

\[
\dot{\mathbf{x}}_t = \frac{1}{Z} \mathbf{t} - \frac{1}{Z} \mathbf{x}(t)_z
\]

(4.7)

is the translational component and

\[
\dot{\mathbf{x}}_r = \omega \times \mathbf{x} - \mathbf{x} (\omega \times \mathbf{x})_z
\]

(4.8)

is the rotational component. Second, the rotational component does not depend on depth information [22].
4.2. Derivation of Image Jacobian Matrix

The relationship between the 3D motion of a scene point relative to the camera and the image motion field observed in an imaging sensor plane is represented by a matrix, called the image Jacobian, which transforms the relative 3D motion to the image motion field. The image Jacobian matrix is one of the important terms for image motion analysis, which is useful for estimating the 3D motion of the camera or visual servoing.

The image Jacobian matrix can be represented in a matrix-vector form from the mathematical model of the image motion field which is just derived. Let $J_f(x)$ be the image Jacobian matrix at an image point $x$ and $\Phi$ be a vector representing the relative motion of the scene point with respect to the camera. Then the image motion field of the image point is a function of the image Jacobian matrix and the relative 3D motion of the scene point:

$$\dot{x} = J_f(x)\Phi$$

(4.9)

where the vector $\Phi = [t^T \omega^T]^T$ contains the relative translational and angular velocities of the scene point $X$ with respect to the camera. The image Jacobian matrix can be partitioned into two sub-matrices representing the contribution of each velocity $t$ and $\omega$ as

$$\dot{x} = \begin{bmatrix} J_{f,t}(x) & J_{f,\omega}(x) \end{bmatrix} \begin{bmatrix} t \\ \omega \end{bmatrix}.$$  

(4.10)

The term $J_{f,t}(x)$ represents the contribution of the relative translational velocity $t$ of the scene point $X$ with respect to the camera to the image motion at an image point $x$ and the term $J_{f,\omega}(x)$ represents the contribution of the relative angular velocity $\omega$ between them to the image motion.
The contribution of each velocity to the image motion can be computed by setting the other component to zero in Eq. (4.5) and (4.10). The contribution \( J_{t,t}(x) \) of the translational velocity \( t \) to the image motion can be computed by setting \( \omega = 0 \):

\[
\dot{x} = \frac{1}{Z} t - \frac{1}{Z} x(t)z.
\]

(4.11)

Let \( k \) be the unit vector along the \( z \)-axis. The contribution \( J_{t,t}(x) \) of the translational velocity \( t \) can be obtained in a matrix-vector form as

\[
J_{t,t}(x)t = \frac{1}{Z}(t - x(t \cdot k))
\]

(4.12)

\[
= \frac{1}{Z}(t - xk^\top t). \quad (4.12)
\]

Then the contribution becomes

\[
J_{t,t}(x) = \frac{1}{Z}(I - xk^\top). \quad (4.13)
\]

Equivalently, the contribution \( J_{t,\omega}(x) \) of the angular velocity \( \omega \) to the image motion can be computed by setting \( t = 0 \):

\[
\dot{x} = \omega \times x - x(\omega \times x) \cdot k
\]

(4.14)

\[
= J_{t,\omega}(x)\omega.
\]

Since the term \((\omega \times x) \cdot k = (x \cdot k) \cdot \omega\) by the relationship between cross and dot products,

\[
J_{t,\omega}(x)\omega = -x \times \omega - x(x \times k)^\top \omega
\]
where the term \([\cdot]\) represents the matrix form of the cross product. Then the contribution becomes

\[
J_{I,\omega}(x) = \left( I - x_k^T \right) x_k^T .
\]  (4.16)

Thus the mathematical model of the image motion field can be written as

\[
\dot{x} = J_I(x)\Phi = \begin{bmatrix} J_{I,t}(x) & J_{I,\omega}(x) \end{bmatrix} \begin{bmatrix} t \\ \omega \end{bmatrix} .
\]  (4.17)

\[
= \left( I - x_k^T \right) \begin{bmatrix} \frac{1}{Z} I \\ x_k^T \end{bmatrix} \begin{bmatrix} t \\ \omega \end{bmatrix}
\]

**4.3. Ego-Motion Model on the Ground Floor**

The image motion field of an image point \( x \) in Eq. (4.17) is a function of the translational and angular velocities of the scene point \( X \) with respect to the camera and the depth \( Z \) between the scene point and the camera:

\[
\dot{x} = f(t, \omega, Z) .
\]  (4.18)

If the depth \( Z \) is known the image motion field becomes a function of the only relative motions \( t \) and \( \omega \):

\[
\dot{x} = f(t, \omega) \]

(4.19)
and the relative 3D motion with respect to the camera can be recovered from the image motion field:

\[(\mathbf{t}, \mathbf{\omega}) = f^{-1}(\mathbf{\hat{x}}).\]  \hspace{1cm} (4.20)

The ground floor detection algorithm was proposed in the previous chapter, which can detect and segment the ground floor region using image motion fields in consecutive images. Since the distance from the camera to the ground floor is already known, the relative 3D motion with respect to the camera can be recovered from the image motion fields on the ground floor region. Suppose that \(\mathbf{\Pi}_G = [\mathbf{n}^T \ d]^T\) represents the ground floor in the camera coordinate. A scene point \(\mathbf{X}\) on the ground floor \(\mathbf{\Pi}_g\) satisfies

\[\mathbf{\Pi}_G^T \mathbf{\hat{X}} = \mathbf{n}^T \mathbf{X} + d = 0\]  \hspace{1cm} (4.21)

where the term \(d\) represents the negative distance from the camera to the ground floor. Writing \(\mathbf{X}\) as \(\mathbf{X} = \mathbf{Z} \mathbf{x}\) from the basic pinhole camera model in Eq. (4.2) to get an expression with respect to the depth yields

\[\mathbf{Z} = -\frac{d}{\mathbf{n}^T \mathbf{x}}.\]  \hspace{1cm} (4.22)

In order to eliminate \(\mathbf{Z}\) in Eq. (4.13), substituting the above equation in Eq. (4.13) yields

\[\mathbf{J}_{\mathbf{t},\mathbf{\omega}}(\mathbf{x}) = \frac{1}{Z} \left( \mathbf{I} - \mathbf{x} \mathbf{k}^T \right) \]
\[= -\frac{n^T \mathbf{x}}{d} \left( \mathbf{I} - \mathbf{x} \mathbf{k}^T \right).\]  \hspace{1cm} (4.23)

Thus the final image motion field model becomes a function of the only translational and angular velocities \(\mathbf{t}\) and \(\mathbf{\omega}\):
\[
\dot{x} = (I - xk^T) \left[ -\frac{n^T x}{d} I \right] \begin{bmatrix} [x]_r \end{bmatrix} \begin{bmatrix} t \end{bmatrix} \quad (4.24)
\]

for \( x = \text{proj}_x(X) \) where \( X \in \Pi_g \).

The above equation represents the relationship between the relative translational and angular velocities of a scene point on the ground floor and the image motion field of the corresponding image point.

If a scene point is on the ground floor, then the relative 3D motion between the ground floor and the camera is caused by the camera motion. Thus the above equation has to be modified so that the vectors \( t \) and \( \omega \) represent the translational and angular velocities of the camera relative to the scene:

\[
\dot{x} = (I - xk^T) \left[ -\frac{n^T x}{d} I \right] \begin{bmatrix} [x]_r \end{bmatrix} \begin{bmatrix} t \end{bmatrix} \quad (4.25)
\]

The above equation is called the *camera ego-motion model* on the ground floor, which is one of the most important terms for camera motion analysis and control. It represents the relation between image motion and the 3D motion of the camera relative to a scene and provides a method to recover the relative 3D motion of the camera from image motion fields without depth information. Then the image Jacobian matrix may be written as

\[
J_f(x) = (I - xk^T) \begin{bmatrix} \begin{bmatrix} v^T \end{bmatrix} I \end{bmatrix} \begin{bmatrix} [x]_r \end{bmatrix}. \quad (4.26)
\]

It transforms the camera motion to the image motion field on the ground floor, which is called the image Jacobian matrix induced by the ground floor \( \Pi_g \) and the vector \( v \) is the plane normal of the ground floor, which is parameterized by \( n/d \).
4.4. Ego-Motion Estimation Algorithms

Ego motion estimation is to determine the camera motion parameters given the image motion field. The relation between the image motion observed in the imaging sensor plane and the camera motion in 3-space is represented by the image Jacobian matrix. If the ground floor image is given the image Jacobian matrix does not carry depth information as in Eq. (4.26).

This section proposes two algorithms to find the camera ego-motion parameters using the ground floor image provided in the ground floor detection algorithm which is derived in the previous chapter. The first is an inverse image Jacobian method which is to determine the camera ego-motion parameters using the derived image Jacobian matrix. It is derived from the Newton-Raphson formula because Eq. (4.20) is not a well posed problem in which the functional relationship between the camera motion parameters and image motion is not one to one. And The planar homography is used for constructing recursive formulation and for verifying iterative solutions. Also it can be computed from at least three image motion vectors. The second is an image Gradient method which is to determine the parameters robustly although inaccurate and noisy image motion vectors are obtained, which finds an optiminimal estimate so that it minimizes computation errors.

Figure 4.2. Camera ego-motion estimation using ground floor detection.
4.4.1 Derivation of an Inverse Image Jacobian Method

Eq. (4.25) represents the relationship between the camera ego-motion relative to a scene and the image motion field observed in an imaging sensor plane by the image Jacobian matrix. Since it has six unknown camera ego-motion parameters \( \Phi_i, \ i=1, \cdots, 6 \) and the maximum rank of the 3\( \times \)6 image Jacobian matrix \( J_i(x) \) is two, the camera ego-motion vector \( \Phi \) can be determined if three image motion vectors are given which are not collinear.

Now suppose that three image motion vectors \( \hat{x}_i \) for \( i=1, \cdots, 3 \) are given in the normalized image plane. From Eq. (4.9) it is possible to make a set of linear equations with respect to the camera motion vector \( \Phi \):

\[
J \Phi = v
\]  \hspace{1cm} (4.27)

where

\[
J = \begin{pmatrix} 
J_i(x_1) \\
J_i(x_2) \\
J_i(x_3) 
\end{pmatrix}
\]  \hspace{1cm} (4.28)

is the accumulated image Jacobian matrix and

\[
v = \begin{pmatrix} 
\hat{x}_1 \\
\hat{x}_2 \\
\hat{x}_3 
\end{pmatrix}
\]  \hspace{1cm} (4.29)

is the accumulated image motion vector. Thus the camera motion vector \( \Phi \) can be obtained by

\[
\Phi = J^{-1} v.
\]  \hspace{1cm} (4.30)
The above equation discloses an important property. If the functional relationship between the camera motion vector and the image motion vector is one to one then a unique solution will be exist. But the same motion field can be produced by two different camera motion vectors [22]. It means that it is impossible to recover uniquely the camera motion vector from the image motion field alone. Thus the above equation is not a well posed problem and may have multiple solutions.

**Image Jacobian based Newton-Raphson Formula**

Newton-Raphson formula is used for finding a solution recursively if an initial estimate for the desired solution is known. It uses the tangential lines analytically evaluated and may be extended to find complex solutions of simultaneous nonlinear equations [13]. In this paper an camera ego-motion estimation algorithm is derived using the Newton-Raphson formula. In the algorithm the derived image Jacobian matrix is used for finding the differential camera motion vector and the planar homography is used for verifying the solutions.

Let \( \Phi^* \) be the desired camera ego-motion vector and \( \Phi_k \) be a current approximation of the desired camera ego-motion vector. If the currently approximated ego-motion vector is close to the desired camera ego-motion vector, then the desired camera ego-motion vector is represented by the sum of the current approximation and the differential camera ego-motion:

\[
\Phi^* = \Phi_k + \Phi_k \delta
\]  

(4.31)

where

\[
\Phi_k = \begin{pmatrix} d_{x,k} \\ d_{y,k} \\ d_{z,k} \\ \delta_{x,k} \\ \delta_{y,k} \\ \delta_{z,k} \end{pmatrix}
\]  

(4.32)
is the current approximation of the **differential camera ego-motion vector** and the terms $d_k$ and $\delta_k$ denotes the current approximations of the **differential angular and translational velocities**, respectively.

If the camera ego-motion is small then the current differential camera ego-motion vector can be obtained from the differential image motion vectors of image points by using the image Jacobian matrix as in Eq. (4.30):

$$\delta \Phi_k = J_k^{-1} \delta \mathbf{v}_k. \quad \text{(4.33)}$$

The term $J_k$ represents the current approximation of the accumulated image Jacobian matrix. For $N$ image points

$$J_k = \left( J_f(x_{1,k})^T \cdots J_f(x_{i,k})^T \cdots J_f(x_{N,k})^T \right)^T \quad \text{(4.34)}$$

where

$$J_f(x_{i,k}) = \left( I - x_{i,k} k^T \right) \left[ I [v^T \mathbf{x}_{i,k}] \right] \quad \text{(4.35)}$$

is the image Jacobian matrix of the currently approximated image point $x_{i,k}$ obtained from Eq.(4.26). The term $\delta \mathbf{v}_k$ represents the current approximation of the accumulated differential image motion vector:

$$\delta \mathbf{v}_k = \left( \delta \mathbf{x}_{1,k}^T \cdots \delta \mathbf{x}_{i,k}^T \cdots \delta \mathbf{x}_{N,k}^T \right)^T. \quad \text{(4.36)}$$

The current approximation of a differential image motion vector $\delta \mathbf{x}_{i,k}$ may represent the difference between the corresponding image point and the current approximation of an image point:

$$\delta \mathbf{x}_{i,k} = x_{i}^' - x_{i,k} \quad \text{(4.37)}$$
where \( x'_i \) denotes the \( i \)-th corresponding image point and \( x_{i,k} \) the current approximation of the \( i \)-th image point.

The current approximation of an image point can be obtained by the current approximation of the planar homography if the image motions are restricted so that they are occurred with respect to a plane in 3-space such as the ground floor. From Eq. (3.34), it is written as

\[
x_{i,k} = H_k x_i
\]  

(4.38)

where

\[
H_k = M_k - m_k v^T. 
\]  

(4.39)

is the current planar homography matrix from Eq. (3.12) and the matrices \( M_k \) and \( m_k \) are the sub-matrices of the current camera projection matrix \( P'_k \) in a canonical form of

\[
P'_k = [M_k | m_k].
\]  

(4.40)

Since the camera projection matrix \( P' \) for the next image plane is represented by the rotation and translation of the camera with respect to the reference image coordinates in Eq. (2.16), the current camera projection matrix \( P'_k \) for the next image plane is written as

\[
P'_k = [R'_k | -R'_k t_k]
\]  

(4.41)

where the terms \( R_k \) and \( t_k \) are the current approximations of the rotation matrix and translation vector in the reference image coordinates. By substituting Eq. (4.40) and (4.41) in Eq. (4.39), the current approximation of the planar homography matrix \( H_k \) can be obtained by
The current approximation of the rotation matrix can be computed from the previous approximation $R_{k-1}$:

$$R_k = R_{k-1} \delta R_{k-1}$$  \hspace{1cm} (4.43)

where the matrix $\delta R_{k-1}$ denotes the previous approximation of the differential rotation matrix. If differential rotation is small, the matrix may be represented by the differential angular velocity $\delta_{k-1}$ in Eq. (4.32):

$$\delta R_{k-1} = R(\delta_{k-1}) = \begin{bmatrix} 1 & -\delta_{z,k-1} & \delta_{y,k-1} \\ \delta_{z,k-1} & 1 & -\delta_{x,k-1} \\ -\delta_{y,k-1} & \delta_{x,k-1} & 1 \end{bmatrix}.$$  \hspace{1cm} (4.44)

The above equation is valid when differential angular velocity approaches zero and for angles less than 0.1 radians ($5.7^\circ$), then error is less than 0.02% for sine functions and 0.5% for cosine functions. But the expression cannot be used in iterative algorithms because the errors accumulate [14]. In this case the differential rotation matrix can be found using Rodriguez’s formula:

$$\delta R_{k-1} = I + \frac{\sin \theta}{\theta} [\delta_k]_\times + \frac{1 - \cos \theta}{\theta^2} [\delta_k]_\times^\top$$  \hspace{1cm} (4.45)

where

$$\theta = \|\delta_k\|$$  \hspace{1cm} (4.46)

is the size of the current differential angular velocity. This is called the
incremental rotation matrix.

The current approximation of the translation vector can be also represented in recursive formulation which is the product of the previous approximation \( \mathbf{t}_{k-1} \) and the previous approximation of the differential translation vector \( \mathbf{\delta}_{k-1} \):

\[
\mathbf{t}_k = \mathbf{t}_{k-1}\mathbf{\delta}_{k-1}.
\]  

(4.47)

Since the differential translation vector can be obtained from Eq. (4.32), the current approximation \( \mathbf{t}_{k-1} \) may be rewritten as

\[
\mathbf{t}_k = \mathbf{t}_{k-1}\mathbf{\delta}_{k-1}.
\]  

(4.48)

If \( \mathbf{R}_k \) is close enough to the camera ego-motion parameters after iterations, the final camera angular velocity \( \mathbf{\omega} \) can be obtained from \( \mathbf{R}_k \) by using the RPY angles which are represented by composition of elementary rotations, Roll-Pitch-Yaw motions about \( z, y \) and \( x \) axes, respectively:

\[
\mathbf{R}_k = \text{Rot}(z, \omega_z)\text{Rot}(y, \omega_y)\text{Rot}(x, \omega_x)
\]

\[
= \begin{bmatrix}
c_{\omega_x}c_{\omega_y} & c_{\omega_x}s_{\omega_y}s_{\omega_z} - s_{\omega_x}c_{\omega_z} & c_{\omega_x}s_{\omega_y}c_{\omega_z} - s_{\omega_x}s_{\omega_z}
c_{\omega_y}c_{\omega_z} & s_{\omega_y}s_{\omega_z} & c_{\omega_y}s_{\omega_z} - c_{\omega_z}s_{\omega_y}
s_{\omega_y} & c_{\omega_y}s_{\omega_z} & c_{\omega_y}c_{\omega_z}
\end{bmatrix}.
\]  

(4.49)

If the rotation matrix is given by

\[
\mathbf{R}_k = \begin{bmatrix}
r_{11} & r_{12} & r_{13} \\
r_{21} & r_{22} & r_{23} \\
r_{31} & r_{32} & r_{33}
\end{bmatrix}
\]  

(4.50)

then the final angular velocity \( \mathbf{\omega} \) can be obtained by comparing Eq. (4.50) with Eq.(4.49). Thus
\[ \omega_z = \text{Atan2}(r_{21}, r_{11}) \]
\[ \omega_y = \text{Atan2}(-r_{31}, \sqrt{r_{32}^2 + r_{33}^2}) \]
\[ \omega_x = \text{Atan2}(r_{32}, r_{33}) \]  \hspace{1cm} (4.51)

The solution degenerates when \( c \omega_y = 0 \). In this case, it is possible to determine only the sum or difference of \( \omega_z \) and \( \omega_x \) [20].

**Assumption:**
1. Internal camera calibration matrix $K$ is known
2. An image of the ground floor is given
3. Image points are normalized with $\mathbf{K}: \mathbf{x} = \mathbf{K}^{-1}\mathbf{p}$ \hspace{1cm} (2.12)

**Objective:**
Given normalized image points $\mathbf{x}_i$ and image motion vectors $\dot{\mathbf{x}}_i$ for $i = 1, \cdots, N$ in the ground floor region, the ground plane normal $\mathbf{v}$ and the unit vector along the $z$-axis $\mathbf{k}$, find the camera ego-motion parameters $\Phi = [\mathbf{\omega}^T, t^T]^T$.

**Corresponding image points:** $\mathbf{x}'_i = \mathbf{x}_i + \dot{\mathbf{x}}_i$

**Initialization of iterative guesses:** $R_0 = \mathbf{I}, \quad t_0 = \mathbf{0}, \quad \mathbf{x}_{i,0} = \mathbf{x}_i$

for $k = 1$ to $K$ with step of 1

Differential image motion vector: $\mathbf{\delta x}_{i,k} = \mathbf{x}'_i - \mathbf{x}_{i,k-1}$ \hspace{1cm} (4.37)

Accumulated image motion vector: $\mathbf{\delta v}_k = \begin{pmatrix} \mathbf{\delta x}_{1,k} \\ \mathbf{\delta x}_{2,k} \\ \vdots \\ \mathbf{\delta x}_{N,k} \end{pmatrix}$ \hspace{1cm} (4.36)

Image Jacobian matrix: $\mathbf{J}_i(\mathbf{x}_{i,k}) = \left( \mathbf{I} - \mathbf{x}_{i,k} \mathbf{K}^T \right) \left( \mathbf{v}^T \mathbf{x}_{i,k} \right) \mathbf{I} \left[ \mathbf{x}_{i,k} \right] \hspace{1cm} (4.35)$

Accumulated image Jacobian matrix: $\mathbf{J}_k = \begin{pmatrix} \mathbf{J}_i(\mathbf{x}_{1,k}) \\ \mathbf{J}_i(\mathbf{x}_{2,k}) \\ \vdots \\ \mathbf{J}_i(\mathbf{x}_{N,k}) \end{pmatrix}$ \hspace{1cm} (4.34)
Differential camera ego-motion vector: \( \partial \Phi_k = J_k^{-1} \partial \mathbf{v}_k \)  

(4.33)

Incremental rotation matrix: \( \partial \mathbf{R}_{k-1} = I + \frac{\sin \theta}{\theta} [\delta_k]_x + \frac{1 - \cos \theta}{\theta^2} [\delta_k]_x^T \)

\[ \theta = \| \delta_k \| \]  

(4.45) (4.46)

Rotation matrix: \( \mathbf{R}_k = \mathbf{R}_{k-1} \phi \mathbf{R}_k \)

(4.47)

Translation vector: \( \mathbf{t}_k = \mathbf{t}_{k-1} \mathbf{\delta}_k \)

(4.48)

Planar homography: \( \mathbf{H}_k = \mathbf{R}_k^T \left( \mathbf{I} + \mathbf{t}_k \mathbf{v}^T \right) \)

(4.42)

Warp \( \mathbf{x}_i \) by \( \mathbf{H}_k \) : \( \mathbf{x}_{i,k} = \mathbf{H}_k \mathbf{x}_i \)

(4.38)

If \( \max \| \mathbf{x}_i' - \mathbf{x}_{i,k} \| < \tau \) then stop

end of for-loop on \( k \)

Final camera translation velocity: \( \mathbf{t} = \mathbf{t}_k \)

Final camera rotation matrix: \( \mathbf{R}_k = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \)

(4.50)

\( \omega_z = \text{Atan2}(r_{21}, r_{11}) \)

Final camera angular velocity: \( \omega_y = \text{Atan2}(-r_{31}, \sqrt{r_{32}^2 + r_{33}^2}) \)

(4.51)

\( \omega_z = \text{Atan2}(r_{32}, r_{33}) \)

Solution: The final camera ego-motion parameters \( \Phi = (\omega^T \quad \mathbf{t}^T)^T \)
4.4.2 Derivation of an Image Gradient Method

The accuracy of estimating the camera ego-motion depends on the accuracy of image motion estimation and motion based ground floor detection. Although the multi-scale coarse-to-fine estimation algorithm produces accurate image motion information, mismatched image motion vectors may be obtained. In this case it is necessary to find an optimal camera ego-motion vector which minimizes errors.

Let $I$ and $I'$ be the reference and the next images, respectively and $\pi_G$ the ground floor image observed in the reference image. Then an error function with respect to the camera ego-motion vector can be defined for some image points in the ground floor image $\pi_G$ and it is represented by the sum of squared difference between their image values in the reference image $I$ and the corresponding image values in the next image $I'$, which are indicated by their image motion vectors:

$$\varepsilon(\Phi) = \sum_{x_i \in \pi_G} (I(x_i) - I'(x_i + \hat{x}_i))^2.$$  \hspace{1cm} (4.52)

Substituting Eq. (4.9) in the above equation yields

$$\varepsilon(\Phi) = \sum_{x_i \in \pi_G} (I(x_i) - I'(x_i + J_i(x_i)\Phi))^2.$$  \hspace{1cm} (4.53)

At the optimum camera ego-motion vector $\Phi^*$, the partial derivatives of $\varepsilon$ with respect to $\Phi$ should be zero:

$$\left. \frac{\varepsilon(\Phi)}{\partial \Phi} \right|_{\Phi^*} = 0.$$  \hspace{1cm} (4.54)

After expansion of the derivatives, Eq.(4.53) becomes
$$\varepsilon(\Phi) = -2 \sum_{x_i \in \pi_E} \frac{\partial I'(x_i + J_i(x_i)\Phi)}{\partial \Phi} (I(x_i) - I'(x_i + J_i(x_i)\Phi)).$$  \hspace{1cm} (4.55)$$

If the camera ego-motion vector $\Phi$ is small, then the term $I'(x_i + J_i(x_i)\Phi)$ can be approximated by its first-order Taylor expansion about $\Phi = 0$:

$$I'(x_i + J_i(x_i)\Phi) = I'(x_i) + \left( \frac{\partial I'(x_i)}{\partial x} \right)^T J_i(x_i)\Phi$$  \hspace{1cm} (4.56)$$

and its partial derivative with respect to $\Phi$ is written as

$$\frac{\partial I'(x_i + J_i(x_i)\Phi)}{\partial \Phi} = \frac{\partial}{\partial \Phi} \left[ \left( \frac{\partial I'(x_i)}{\partial x} \right)^T J_i(x_i)\Phi \right]$$  \hspace{1cm} (4.57)$$

By substituting Eq. (4.56), (4.57) in Eq. (4.55), the partial derivative of $\varepsilon$ becomes

$$\frac{1}{2} \varepsilon(\Phi) = \sum_{x_i \in \pi_E} J_i(x_i)^T \left( \frac{\partial I'(x_i)}{\partial x} \right) \left( \left( \frac{\partial I'(x_i)}{\partial x} \right)^T J_i(x_i)\Phi - (I(x_i) - I'(x_i)) \right)$$  \hspace{1cm} (4.58)$$

and it should be zero at optimimum

$$\frac{1}{2} \varepsilon(\Phi) \bigg|_{\Phi = \Phi^*} = 0 = \sum_{x_i \in \pi_E} J_i(x_i)^T \nabla I'(x_i) \left( \nabla I'(x_i)^T J_i(x_i)\Phi^* - \partial I(x_i) \right)$$  \hspace{1cm} (4.59)$$

where
\[ \nabla I'(x_i) = \frac{\partial I'(x_i)}{\partial x} \quad (4.60) \]

is \textit{spatial image gradient} vector at an image point \( x_i \) of the next image and

\[ \delta I(x_i) = I(x_i) - I'(x_i) \quad (4.61) \]

is \textit{temporal image difference} at that point between two images. Thus the solution of Eq. (4.59) is given by

\[ \Phi^* = \left( \sum_{x_i \in \Omega} J_i(x_i)^T \nabla I'(x_i) \nabla I'(x_i)^T J_i(x_i) \right)^{-1} \left( \sum_{x_i \in \Omega} J_i(x_i)^T \nabla I'(x_i) \delta I(x_i) \right). \quad (4.62) \]

Let

\[ H_J = \sum_{x_i \in \Omega} J_i(x_i)^T \nabla I'(x_i) \nabla I'(x_i)^T J_i(x_i) \quad (4.63) \]

be the \textit{image Hessian matrix} and

\[ m_J = \sum_{x_i \in \Omega} J_i(x_i)^T \nabla I'(x_i) \delta I(x_i) \quad (4.64) \]

be the \textit{image mismatch vector}. Then Eq. (4.62) becomes a concise form

\[ \Phi^* = H_J^{-1} m_J. \quad (4.65) \]

Since the least squared solution of Eq. (4.30) can be written as

\[ \Phi^* = (J^T J)^{-1} J^T v \quad (4.66) \]
the solution in Eq. (4.65) is the weighted least squared solution by the spatial image gradient \( \nabla I'(x_i) \). Thus the convergence rate of Eq. (4.65) is faster than the inverse image Jacobian method derived in Algorithm 4.1.

The weight least squared solution is also used with the inverse image Jacobian method. In Eq. (4.33), the differential camera ego-motion vector may be computed by the weight least squared solution in Eq. (4.65):

\[
\partial \Phi_k = H_{f,k}^{-1} m_{f,k}
\] (4.67)

where

\[
H_{f,k} = \sum_{x_i \in \sigma_o} J_f(x_{i,k})^T \nabla I'(x_{i,k}) \nabla I'(x_{i,k})^T J_f(x_{i,k})
\] (4.68)

denotes the current approximation of the image Hessian matrix in Eq. (4.63) and

\[
m_{f,k} = \sum_{x_i \in \sigma_o} J_f(x_{i,k})^T \nabla I'(x_{i,k}) \partial I(x_{i,k})
\] (4.69)

denotes the current approximation of the image mismatch vector in Eq. (4.64). And the current approximations of the spatial image gradient and the temporal image difference may be rewritten as

\[
\nabla I'(x_{i,k}) = \frac{\partial I'(x_{i,k})}{\partial x},
\]

(4.70)

\[
\partial I(x_{i,k}) = I(x_i') - I'(x_{i,k}).
\]

(4.71)
Algorithm 4.2. Summary of the image gradient based algorithm.

Assumption:
1. Internal camera calibration matrix $K$ is known
2. An image of the ground floor is given
3. Image points are normalized with $K: \mathbf{x} = K^{-1}\tilde{p}$ (2.12)

Objective:
Given normalized image points $\mathbf{x}_i$ and image motion vectors $\hat{\mathbf{x}}_i$ for $i = 1, \cdots, N$ in the ground floor region, the ground plane normal $\mathbf{v}$ and the unit vector along the $z$-axis $\mathbf{k}$, find the camera ego-motion parameters $\Phi = [\omega^T, t^T]^T$

Corresponding image points: $\mathbf{x}'_i = \mathbf{x}_i + \hat{\mathbf{x}}_i$

Initialization of iterative guesses: $R_0 = I$, $t_0 = 0$, $\mathbf{x}_{i,0} = \mathbf{x}_i$

for $k = 1$ to $K$ with step of 1

Temporal image difference: $\delta I(\mathbf{x}_{i,k}) = I(\mathbf{x}') - I'(\mathbf{x}_{i,k-1})$ (4.71)

Spatial image gradient: $\nabla I'(\mathbf{x}_{i,k}) = \frac{\partial I'(\mathbf{x}_{i,k})}{\partial \mathbf{x}}$ (4.70)

Image Jacobian matrix: $\mathbf{J}_I(\mathbf{x}_{i,k}) = \left(I - \mathbf{x}_{i,k}k^T\right) \left(v^T \mathbf{x}_{i,k}\right) \mathbf{I} \left[\mathbf{x}_{i,k}\right]$ (4.35)

Image Hessian matrix: $\mathbf{H}_{I,k} = \sum_{\mathbf{x} \in \pi_G} \mathbf{J}_I(\mathbf{x}_{i,k})^T \nabla I'(\mathbf{x}_{i,k}) \nabla I'(\mathbf{x}_{i,k})^T \mathbf{J}_I(\mathbf{x}_{i,k})$ (4.68)

Image mismatch vector: $\mathbf{m}_{i,k} = \sum_{\mathbf{x} \in \pi_G} \mathbf{J}_I(\mathbf{x}_{i,k})^T \nabla I'(\mathbf{x}_{i,k}) \delta I(\mathbf{x}_{i,k})$ (4.69)

Differential camera ego-motion vector: $\delta \Phi_k = \mathbf{H}_{I,k}^{-1} \mathbf{m}_{I,k}$ (4.67)
Incremental rotation matrix:

\[ \partial \mathbf{R}_{k-1} = I + \frac{\sin \theta}{\theta} \mathbf{\delta}_k + \frac{1 - \cos \theta}{\theta^2} \mathbf{\delta}_k^\top \]

\[ \theta = \| \mathbf{\delta}_k \| \]  

(4.45) (4.46)

Rotation matrix:

\[ \mathbf{R}_k = \mathbf{R}_{k-1} \mathbf{\delta}_k \]  

(4.47)

Translation vector:

\[ \mathbf{t}_k = \mathbf{t}_{k-1} \mathbf{\delta}_k \]  

(4.48)

Planar homography:

\[ \mathbf{H}_k = \mathbf{R}_k^\top \left( \mathbf{I} + \mathbf{t}_k \mathbf{v}^\top \right) \]  

(4.42)

Warp \( \mathbf{x}_i \) by \( \mathbf{H}_k \):

\[ \mathbf{x}_{i,k} = \mathbf{H}_k \mathbf{x}_i \]  

(4.38)

If \( \max \| \mathbf{x}'_i - \mathbf{x}_{i,k} \| < \tau \) then stop

end of for-loop on \( k \)

Final camera translation velocity:

\[ \mathbf{t} = \mathbf{t}_k \]

Final camera rotation matrix:

\[ \mathbf{R}_k = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \]  

(4.50)

\[ \omega_z = \text{Atan2}(r_{21}, r_{11}) \]

Final camera angular velocity:

\[ \omega_y = \text{Atan2}(-r_{31}, \sqrt{r_{22}^2 + r_{33}^2}) \]  

(4.51)

\[ \omega_x = \text{Atan2}(r_{32}, r_{33}) \]

Solution: The final camera ego-motion parameters \( \mathbf{\Phi} = (\mathbf{\omega}^\top \mathbf{t}^\top)^\top \)
4.5. Experimental Results

This section describes experimental results on planar ego-motion estimation using the proposed inverse Jacobian methods. Tests used both synthetic and real data to verify that the algorithm performed ego-motion estimation correctly.

4.5.1 Synthetic Case

In all of simulations, synthetic data consisted of a random cloud of points which were placed on a plane in 3D space, the ground floor, in front of the simulated camera. Each set of point consisted of 50 randomly chosen sample points. The focal length was set to 1. The distance to the ground floor was 1 m along to the download \( y \)-axis. The depth range was 1.5 to 6 m along to the forward \( z \)-axis. And the maximum width was 2 m.

Various combinations of translational and angular velocities were chosen. A camera was considered to be positioned on the top of a mobile robot. Thus the camera motions were composed of translations in the \( xy \)-plane parallel to the ground floor and rotations around \( y \)-axis.

Zero-mean Gaussian noises of various levels were added to each component of image motion vectors by considering noisy data in image motion estimation. 10 noise levels, in pixels, were considered in tests.

In order to check the correctness and accuracy of the algorithm, three bias and three sensitivities were measured for each noise level as the mean and the standard deviation of the estimates: bias and sensitivities on translational, angular velocities and image transfer error. One thousand trials were performed with 50 image motion vectors.

Translation bias \( h_t \) was computed for each noise level as the mean of the Euclidean distances between the true translational velocities \( t_t \) and
the estimates $\hat{t}_i$ over $M$ trials:

$$b_i = \frac{1}{M} \sum_{i=1}^{M} \|t_i - \hat{t}_i\|$$  \hspace{1cm} (4.72)

where $M = 1000$. Rotation bias might be equivalently written as

$$b_\omega = \frac{1}{M} \sum_{i=1}^{M} \|\omega_i - \hat{\omega}_i\|$$  \hspace{1cm} (4.73)

where $\omega_i, \hat{\omega}_i$ denote the $i$-th true angular velocity and the $i$-th estimate. And image transfer bias was computed as the mean of Euclidean image distances between the true corresponding image points $x'_y$ indicated by the image motion vectors $\hat{x}_y$ and the estimated image points $\hat{x}'_y$ warped by the planar homography matrix composed of $\hat{t}_i$ and $\hat{\omega}_i$ over $M$ trials and $N$ image points:

$$b_x = \frac{1}{MN} \sum_{j=1}^{N} \sum_{i=1}^{M} \|x'_y - \hat{x}'_y\|$$  \hspace{1cm} (4.74)

where

$$x'_y = x_y + \hat{x}_y$$  \hspace{1cm} (4.75)

is the $j$-th true corresponding image point in the $i$-th trial and

$$\hat{x}'_y = R(\hat{\omega}_i)^T \left[ I - \hat{t}_i \right] x_y$$  \hspace{1cm} (4.76)

is the $j$-th estimated image point in the $i$-th trial, from Eq. (3.35).

Translation sensitivity for each noise level was computed as the standard deviation of the translation bias over $M$ trials:
\[
\sigma_i = \sqrt{\frac{1}{M} \sum_{j=1}^{M} (b_i - \hat{b}_i)^2}
\]  
(4.77)

where

\[
h_i = \|t_i - \hat{t}_i\|
\]  
(4.78)

is the \(i\)-th translation bias. And rotation bias and image transfer bias could be computed in this way.

Figure 4.3 plots bias and sensitivities on translational, angular velocities and image transfer error. The performance of the algorithm is proportional to the noise levels. In an extensive series of preliminary simulations in which a set of simulations was preformed with rotation about the \(y\)-axis and another set of simulations with rotation about the \(z\)-axis, the axis of rotation had no impact on bias and sensitivities of the proposed algorithms. Thus the performance of the algorithms is invariant with respect to the rotation axis.

### 4.5.2 Real Case

The experiments on real data were performed planar ego-motion estimation with known camera ego-motion parameters when a camera moves on the ground floor for the cases of pure translation, pure rotation and general motion. For pure translation an image sequence of 8 images are obtained in intervals, 3 cm, when a camera is moving forward on the ground floor. For pure rotation an image sequence of 9 images are obtained in intervals, 180/32(about 5.6) degrees, when a camera is rotating on the ground floor. For general motion, an image sequence of 4 images are obtained in intervals, 180/16(about 11.3) degrees and 4 cm when a camera is
rotated and translated on the ground floor.

Figure 4.4a shows the first image of the image sequence for pure translation, which shows the image points on the ground floor and the image motion vectors. Figure 4.4b plots the bias on translational and angular velocities. Figure 4.4c shows the reconstructed camera positions and orientations. Figure 4.5 and Figure 4.6 are experimental results for pure rotation and general motion.
Figure 4.3. Bias and sensitivities on translational, angular velocities and image transfer error.
(a) image points and image motion vectors

(b) bias on translational and angular velocities

(c) camera positions and orientations

Figure 4.4. Experimental results for pure translation.
Figure 4.5. Experimental results for pure rotation.
Figure 4.6. Experimental results for general motion.
4.6. Comparison and Summary

General ego-motion model is a function of image motion vectors and 3D information. If plane information such as the ground floor is given, the ego-motion model does not carry 3D information any more. The image Jacobian matrix and the camera ego-motion model were derived in a compact matrix-vector form to develop ego-motion estimation algorithms. Two ego-motion estimation algorithms were developed: First, the inverse Jacobian method was proposed which was derived from the Newton-Raphson formula based on image Jacobian since the camera ego-motion model is not a well posed problem. Secondly, the image Gradient method was developed which provides an optiminimal estimate to determine the ego-motion parameters robustly so that it minimizes computation errors although inaccurate and noisy image motion vector are obtained.

The preliminary experiments on synthetic data showed the correctness and accuracy of the proposed algorithms with bias and sensitivities of translational, angular velocities and image transfer for various noise levels over a number of trials. The extensive experiments on real scenes demonstrated that the algorithms performed ego-motion estimation on image sequences correctly.
CHAPTER 5

CONCLUSIONS AND FUTURE WORKS

This thesis addressed two coupled problems in visual navigation for a mobile robot operating in unknown structured environments: local map building and localization. In order to solve the problem of local map building, the ground floor detection algorithm was proposed, which was differed from many traditional approaches. And two camera ego-motion estimation algorithms were developed to solve the problem of localization, which used plane information provided in algorithms such as the ground floor detection algorithm.

The proposed ground floor detection algorithm exploited the geometric fact and some assumptions observed in structured scenes. In order to detect the ground floor in images, planar homography induced by a plane was considered and two methods were derived for computing plane normals. They were direct estimation with three image point correspondences and optimimal estimation with noisy three or more image point correspondences. As a result a robust plane normal estimation algorithm that minimizes the computation error was developed although mismatched image point correspondences are obtained.

In order to show the validity of the proposed method, the multi-scale coarse-to-fine estimation method was adopted in the algorithms so that accurate and dense point correspondences could be obtained and the iterative refinement process was designed to segment the ground floor image from scene images. Also the layered image representation was used for representing the ground floor image in a proper form for visual navigation. As a result, the proposed algorithm detected the only ground
floor image although mismatched image point correspondences were obtained.

The preliminary experiments on real scenes demonstrated the effectiveness of the proposed ground floor detection algorithm and verified the appropriateness of the geometric fact observed in the scene and the correctness of the assumptions about scene images. And also it was shown that the validity of the estimation of the derived optimimal plane normals.

The proposed camera ego-motion estimation algorithm used plane information because the ego-motion model does not carry 3D information any more if plane information is given, such as the ground floor. At first image Jacobian and the camera ego-motion estimation model were derived in a compact matrix-vector form for developing two ego-motion estimation algorithms. The first is the inverse Jacobian method was proposed which was derived from the iterative Newton-Raphson formula based on image Jacobian since the camera ego-motion model is not a well posed problem. Secondly, the image Gradient method was developed to provide an optimimal estimate which determined the ego-motion parameters robustly so that it minimizes computation errors although inaccurate and noisy image motion vector are obtained.

The preliminary experiments on synthetic data showed the correctness and accuracy of the proposed algorithms with bias and sensitivities of translational, angular velocities and image transfer for various noise levels over a number of trials. The extensive experiments on real scenes demonstrated that the algorithms performed ego-motion estimation on image sequences correctly.

In order to obtain more precise ground floor image, we will need an algorithm to produce more accurate image point correspondences. In order to eliminate the assumption in the ground floor detection algorithm, which is, the camera is moving with respect to a fixed environment, an algorithm should be developed to segment static and dynamic objects based on image motions.