論文 / 著書情報
Article / Book Information

題目(和文)	
Title(English)	A Robust Visual-Feature-Extraction Method in Public Environment
著者(和文)	ｶｺﾞｾｰ
Author(English)	Gangchen Hua
出典(和文)	学位:博士（工学）, 学位授与機関:東京工業大学, 報告番号:甲第9871号, 授与年月日:2015年3月26日, 学位の種別:課程博士, 審査員:長谷川　修,新田　克己,三宅　美博,小野　功,長谷川　晶一
Citation(English)	Degree:, Conferring organization: Tokyo Institute of Technology, Report number:甲第9871号, Conferred date:2015/3/26, Degree Type:Course doctor, Examiner:,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

Tokyo Institute of Technology

Doctoral Thesis

A Robust Visual-Feature-Extraction Method in Public Environment

Author:
Gangchen Hua

Supervisor:
Osamu Hasegawa

A thesis submitted in fulfilment of the requirements
for the degree of Doctor of Philosophy

in the

Department of Computational Intelligence and Systems Science

February 2015

TOKYO INSTITUTE OF TECHNOLOGY

Abstract

Interdisciplinary Graduate School of Science and Engineering
Department of Computational Intelligence and Systems Science

Doctor of Philosophy

A Robust Visual-Feature-Extraction Method in Public Environment

by Gangchen Hua

In this study we describe a new feature extracting method that can extract robust features from a sequence of images and also performs satisfactorily in a highly dynamic environment. This method is based on the geometric structure of matched local feature points. When compared with other previous methods, the proposed method is more accurate in appearance-only simultaneous localization and mapping (SLAM). When compared to position-invariant robust features[1], the proposed method is more suitable for low-cost, single conventional-lens cameras with narrow fields-of-view.

We tested our method in an outdoor environment at Shibuya station. We captured these images by using a conventional hand-held, single-lens video camera. These environments of experiments are public environments without any planned landmarks. The results show that the proposed method can accurately obtain matches for two visual-feature sets, and using the proposed method, stable and accurate *appearance-only SLAM* can be achieved in public dynamic environments.

In chapter 6, we show the proposed method based *hybrid SLAM* and *visual odometry* also work better than other previous methods in public dynamic environments.

Acknowledgements

First and foremost, thanks are given to my supervisor associate professor Osamu Hasegawa. With a number of times for our discussion, I learned uncountable things from him.

Thanks to professor Katsumi Nitta, professor Yoshihiro Miyake, associate professor Shoichi Hasegawa, associate professor Isao Ono for very helpful suggestions to improve my research and presentation.

Thanks to all the members of Hasegawa research groups for friendship and good times.

Also, I want to acknowledge the support from Core Research for Evolutional Science and Technology (CREST) program of Japan Science and Technology Agency(JST) and Toyota Motor Corporation(TMC) in terms of discussion and funding.

Finally, thanks to my family, without their support, I'm nothing.

Contents

Chapter 1

INTRODUCTION

Simultaneous localization and mapping (SLAM) is widely used to generate maps for localization or autonomous robotic navigation.

Appearance-only SLAM is a type of low-cost solution. Moreover, SLAM based on visual features has abundant information that can be used for matching and recognition.

There are two kinds of visual features, including local visual features and global visual features. Global visual features ([2]etc.) extract one feature for each image and are used for vision-based generic recognition (object classification, scene classification etc.) generally. Typical local visual features extract multiple feature points at each image's multiple interesting locations. Local visual features based solutions are widely used for vision-based specific recognition (robotic navigation, SLAM etc.). Because the vision-based SLAM is a kind of specific recognition, we have to use the local visual feature. scale-invariant Feature Transform (SIFT)[3] and Speeded Up Robust Features(SURF)[4] are typical local visual features.

SIFT calculates scale-invariant features at interesting locations, these features are 128-dimension vectors. The feature matching and indexing process uses a modification of the k-d tree algorithm called the Best-bin-first[5] search method that can identify the nearest neighbors with high probability using only a limited amount of computation. SURF is similar to SIFT, however, for faster processing SURF's feature is 64-dimension. The proposed method uses SURF to track feature points.

M. Cummins *et al.* [6] proposed a rapid method based on the probabilistic bail-out condition for appearance-only SLAM. It is called FAB-MAP. An offline dictionary need to be generated before running, so FAB-MAP is not a complete online incremental solution. FAB-MAP uses SURF as local visual feature. F However, the appearance of objects in the actual world is always dynamic. Many appearance-only SLAM methods are based on the hypothesized static environment. These methods use SIFT[3] or SURF[4]. These local-visual features do not have a strong influence on moving objects such as walking humans in cafeterias, stations, or shopping malls.

For appearance-only SLAM, the visual features' robustness and the effectiveness of the matching are important.

A. Kawewong *et al.*[1] proposed a method that tracks robust features in a sequence of images, called position-invariant robust features (PIRF). PIRF extracts common features by referring to past images. So it is more robust than original SIFT or SURF in dynamic environments. In addition, A. Kawewong *et al.* proposed two online-incremental-appearance-only methods for SLAM PIRF-nav[7] and PIRF-nav2.0[8] on the basis of [1]. The methods in PIRF-nav[7] and PIRF-nav2.0[8] perform better than [6] in dynamic

environments. Besides, [7] and [8] are fully online incremental methods. Fig. 1.1 shows the basic algorithm of PIRF extraction.

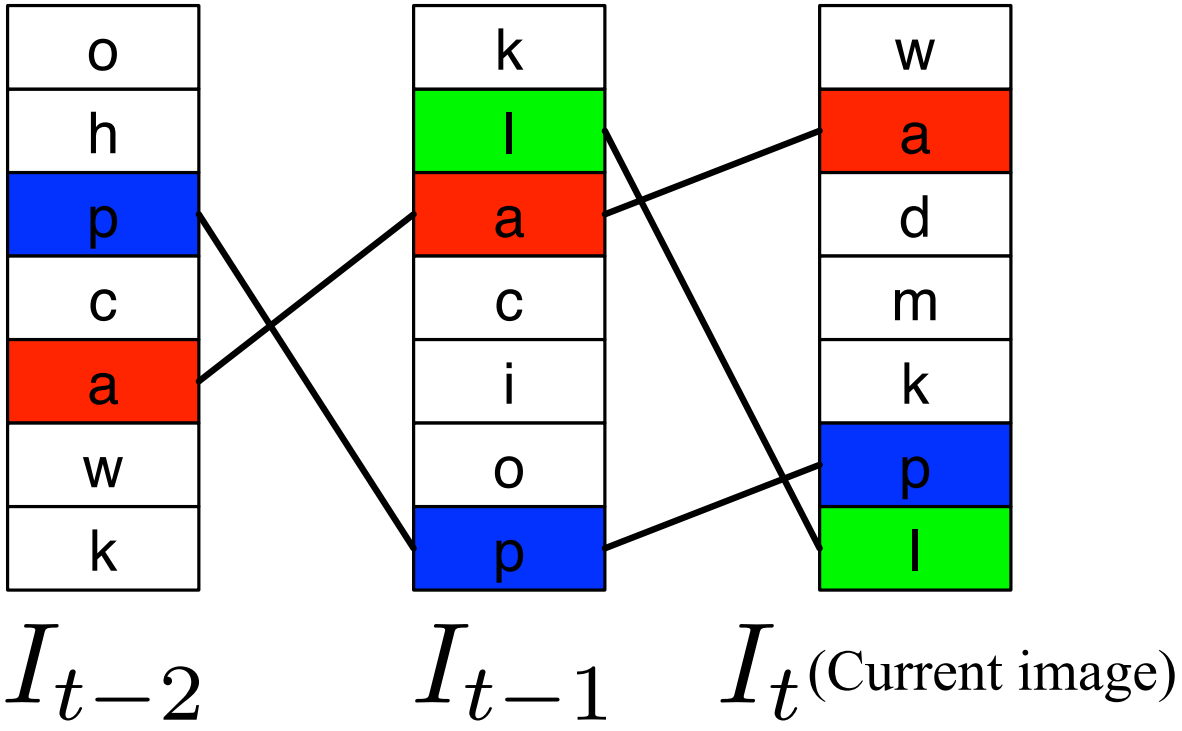


FIGURE 1.1: Basic algorithm of PIRF extraction [1]: a,b,c ... are visual features(SIFT or SURF) of each image. PIRF [1] extracts current image's robust features by referring to past images. Only common features are extracted. In this figure, current image I_t's PIRFs are a and p.

However, [7] is based on SIFT ,[6] and [8] are based on SURF. Because they match and index features only on basis of n-dimensional nearest neighbors , they are called pure bag-of-words (BoW) methods.

Fig. 1.2 shows a disadvantage of previous pure BoW SLAM methods. Because SIFT or SURF, so in certain situations, BoW SLAM methods are not sufficiently robust.

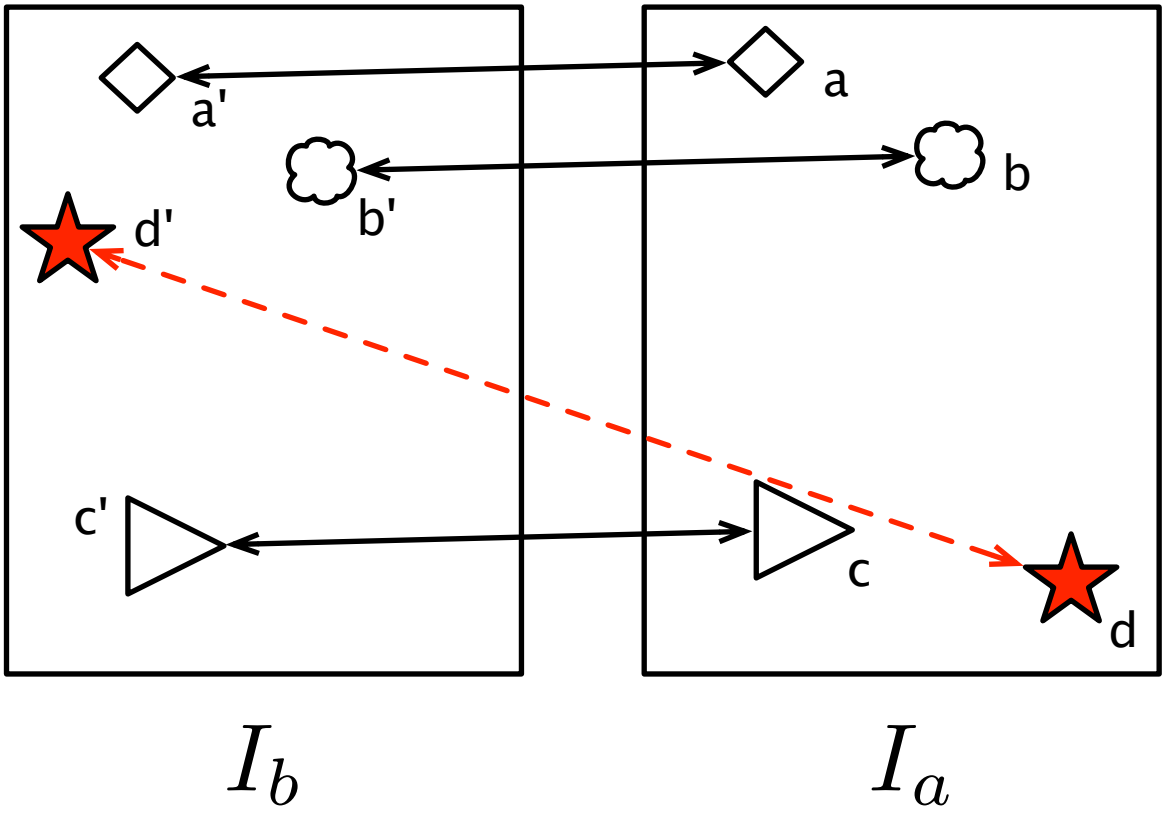


FIGURE 1.2: Disadvantage of pure BoW methods: I_a and I_b are images photographed at almost the same location. $a \leftrightarrow a'$, $b \leftrightarrow b'$, $c \leftrightarrow c'$ and $d \leftrightarrow d'$ are local feature points' matches betweens two places. All correct matches' links should be approximatively horizontal. $a \leftrightarrow a'$, $b \leftrightarrow b'$ and $c \leftrightarrow c'$ are correct matches, $d \leftrightarrow d'$ is incorrect. Although, pure BoW methods can not distinguish them.

To avoid this problem, we propose a method called Incremental Center of Gravity Matching (ICGM) that uses relative geometric structure of local feature points to track robust features in a sequence of images. Fig. 1.3 shows the ICGM's basic algorithm to distinguish incorrect matches and correct matches. Similar to PIRF, we propose a SLAM method that extracts robust features by referring to past image (I_{t-1}) on the basis of ICGM. It is called single-directional ICGM (Fig. 4.2). It works better than PIRF. Moreover, because of the reasons described in 2.1.3 and 2.1.4. PIRF as well as single-directional ICGM always causes significant loss of features. So we also proposed a method called double-directional ICGM (Fig. 4.3). Double-directional ICGM extracts robust features not only by referring to past image (I_{t-1}) but also by referring to future image (I_{t+1}).

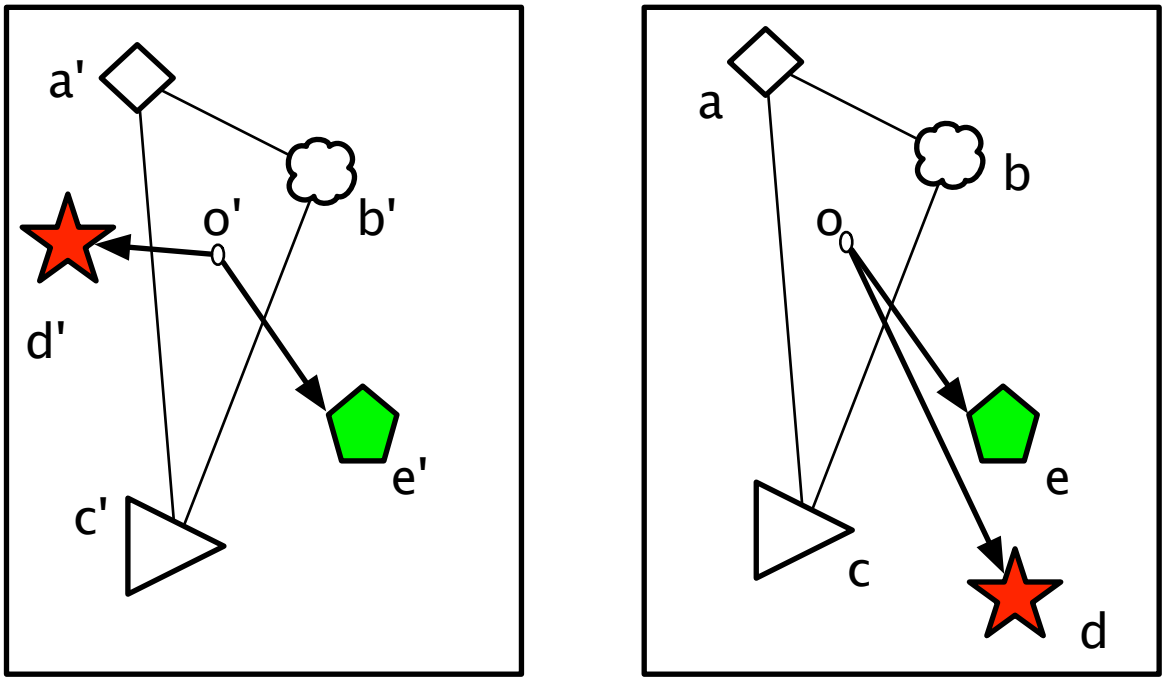


FIGURE 1.3: Basic algorithm of ICGM: Similar to Fig. 1.2, $a \leftrightarrow a'$, $b \leftrightarrow b'$, $c \leftrightarrow c'$,$d \leftrightarrow d'$ and $e \leftrightarrow e'$ are local feature points matched by Best-bin-first[5]. However, $d \leftrightarrow d'$ is incorrect. Assuming that the algorithm has already known that $a \leftrightarrow a'$, $b \leftrightarrow b'$, $c \leftrightarrow c'$ are correct, and the algorithm attempts to calculate reliabilities of $d \leftrightarrow d'$ and $e \leftrightarrow e'$. o is the center of gravity of $\triangle abc$, o' is the center of gravity of $\triangle a'b'c'$. $\vec{oe} \approx \vec{o'e'}$, meanwhile, $\vec{od} \neq \vec{o'd'}$. ICGM calculates feature points' matches' reliability on the basis of these vectors' relationship. Because $\vec{oe} \approx \vec{o'e'}$, the match $e \leftrightarrow e'$'s reliability is high. $\vec{od} \neq \vec{o'd'}$, so the match $d \leftrightarrow d'$'s reliability is low. ICGM avoids matches with low reliability. So ICGM removes $d \leftrightarrow d'$ and keeps $e \leftrightarrow e'$.

Because of increasing of features, the double-directional ICGM's performance is the best. However, the double-directional ICGM needs the image of "future (I_{t+1})", so double-directional ICGM causes delay. In the following sections, we would like to discuss this in detail.

Besides, single-directional ICGM and double-directional ICGM are online fully incremental solutions.

Chapter 2

SLAMs

There are many kinds of SLAMs(simultaneous localization and mapping). Because Bag-of-Words based SLAMs are fast and accurate, in this chapter, at first bag-of-words approaches are introduced. Then appearance-only and hybrid SLAMs are introduced.

2.1 Bag-of-Words Approaches

Bag-of-Words's concept is the same as text analysis. Bag-of-Words approaches creat dictionaries as bags of unique keywords. Fig 2.1 shows the bag of words model in documents.

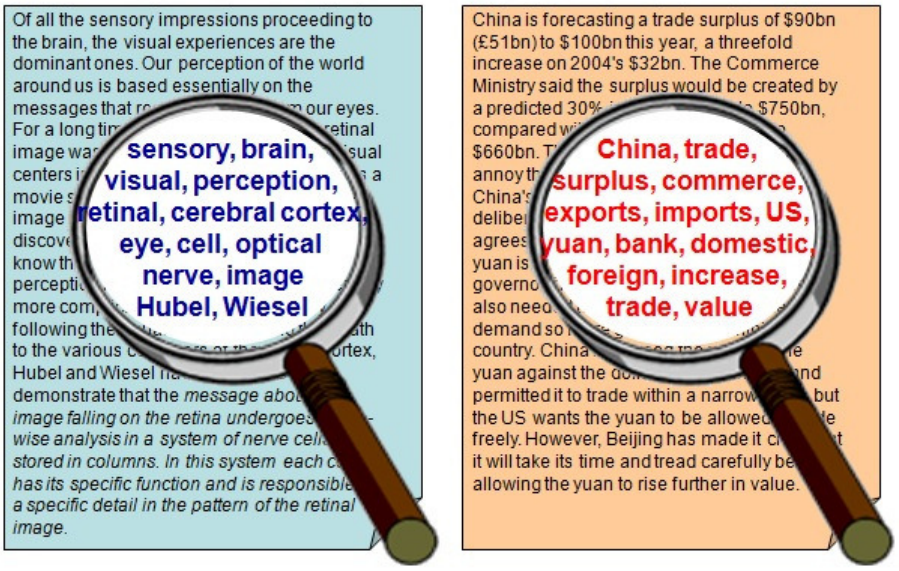


FIGURE 2.1: BoW model in documents.

The idea is similar in computer vision. In computer vision, images are represented as bags of visual words. Visual words are described by descriptors. Fig 2.2 shows the bag of words model for computer vision.

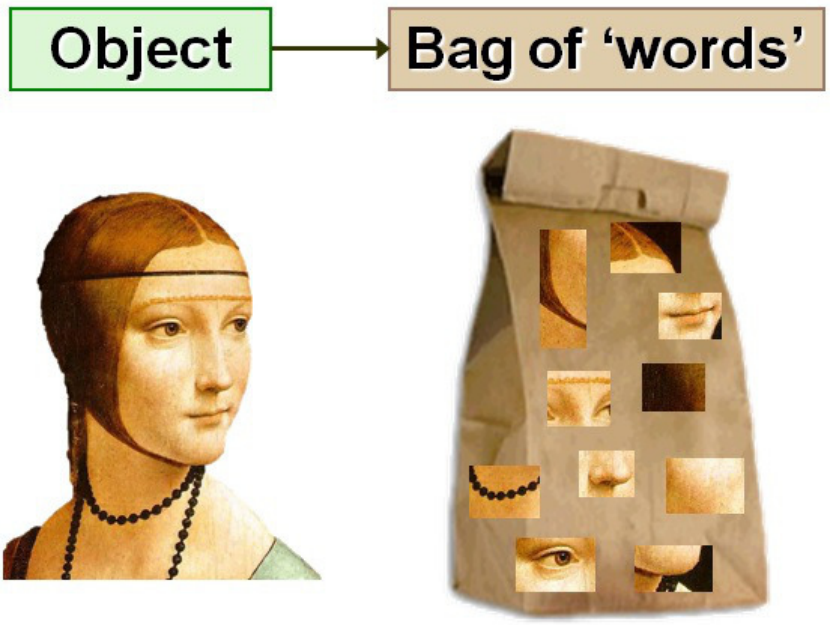


FIGURE 2.2: BoW model in images.

BoW is used for object or scene recognition. By constructing dictionaries objects or scenes are discribed as histograms of the frequency words that are in the images. Fig. 2.3 shows the concept.

In image processing, scale and rotation invariant features[3][4] are used.

Dictionary without ordering of the words is constructed after descriptors are extracted.

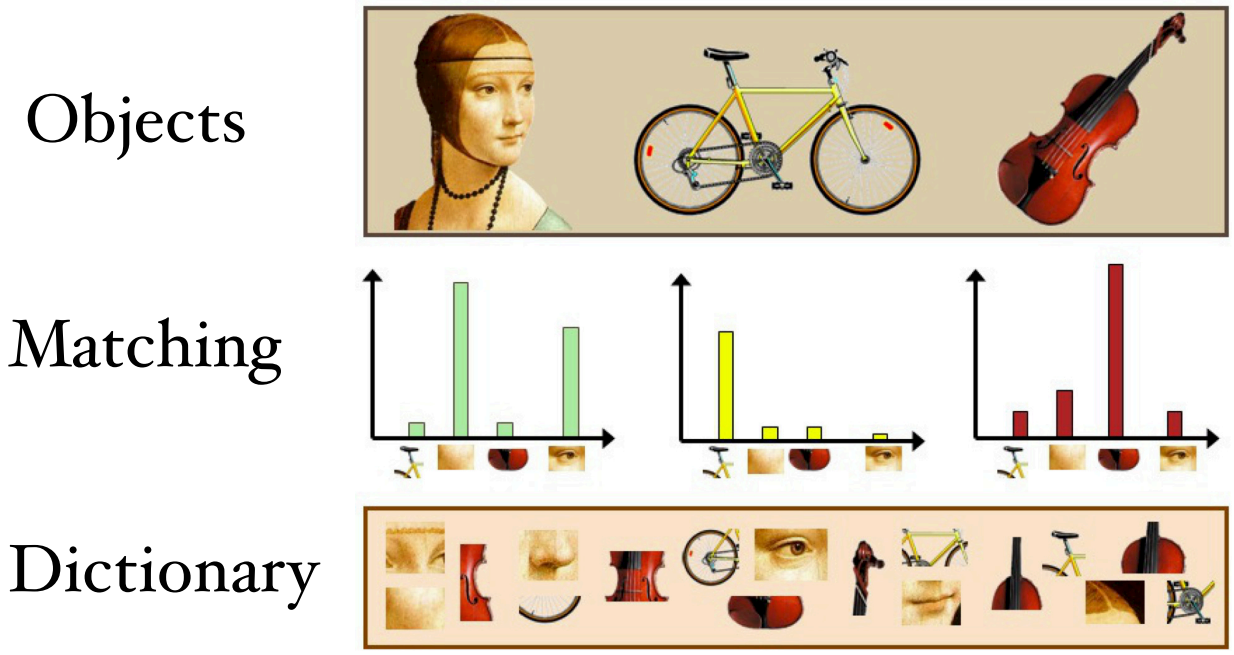


FIGURE 2.3: BoW: Object are representing as histograms of words occurrences.

2.2 Appearance-only SLAMs

Appearance-only SLAMs are metric SLAMs.

Appearance-only SLAMs means all localization are achieved by image feature. Appearance-only SLAMs do not calculate accurate trajectory of cameras. So they can not localize the camera in global coordinates accurately. They are not suitble for robotic SLAMs, but they are basis of hybrid robotic SLAMs.

Torralba et al. [9] porposed a method for location recognition by using Gaussian Mixed Model.

Lazebnik et al. [10] porposed a method to cluster SIFT features on the basis of k-means algorithm.

Cummins et al. [6] proposed a BoW method into the Bayesian probabilistic framework, [6] works well in the localization.

In 2008, Angeli et al. [11] proposed fast and incremental BoW. It is a incremental method, the systmen do not need to creat a off-line dictionary. [11] incrementally collects new words while localizing places.

Later, Kawewong,A. et al. [12][7][1][13] proposed a SLAM method based on SIFT called PIRF-nav which is robust against appearance's changing of environment. Referring to [12], it works better than other methods.

[12] is a online and incremental method. But it dose not create a BoW dictionary, this causes redundancy matches. So [12] is slower than BoW methods([11],[6]).

In 2011, Tongprasit,N. and Kawewong,A.[8][14][15] proposed a faster method called PIRF-nav2.0. PIRF-nav2.0 is based on SURF. It creates incremental Bow dictionary to recognize places. It is 3 times faster than [12]. In addition it is as accurate as [12]. Fig. 2.4 shows the [8]'s matching process with incremental dictionary.

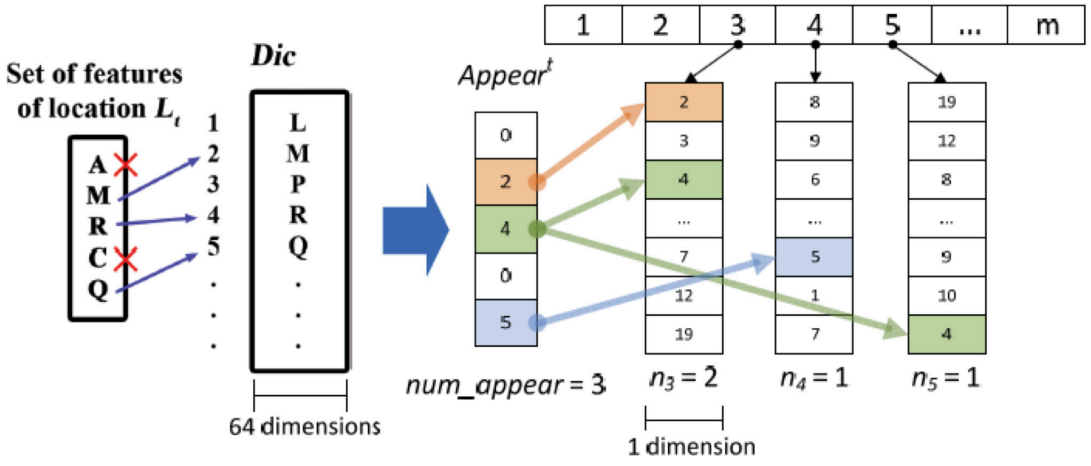


FIGURE 2.4: [8]'s matching process with incremental dictionary. In this occasion, location t matched 2 features with location 3, matched 1 features with location 4. In the current location L_t, the system finds matching features in *Dic* and puts the index numbers into $Appear^t$. Then the likelihood between images can be calculated. New words of L_t are inserted into the *Dic* incrementally.

2.3 Hybrid SLAMs

Hybrid SLAMs are combined by metric SLAMs and topological SLAMs. Hybrid SLAMs use some kinds of odometry to calculate camera trajectories. Hybrid maps are established based on learned metric features and trajectories. Robot's accurate location in the map can be calculate based on hybrid map. So hybrid vision based SLAMs are very suitable for accurate robotic navigations.

In 2005, Joan Sol et al. [16] elaborated that the (SLAM) is one of the fundamental ones in mobile robotics. It concerns the incremental construction of an environment model from the fusion of sensory data (laser range finder, vision,...) and proprioceptive data acquired by an inertial measurement unit or by odometry. Sensory data are processed to extract features.

Pfingsthorn, Max et al. [17] won the Best Mapping Award in the RoboCup Rescue Virtual Robots competition in 2006 by a hybrid SLAM method. This method is combining the strengths of topological maps and occupancy grids. This method maintains a graph with sensor observations stored in vertices and pose differences including uncertainty information stored in edges. Through its graph structure, updates are local and can be efficiently communicated to peers. The graph links represent known traversable space, and facilitate tasks like path planning.

Blanco, J. et al .[18] poposed a approach that is based on the reconstruction of the robot path in a hybrid discrete-continuous state space, which naturally combines vision and topological maps.

But dynamic environment is a big challenge of robotic hybrid SLAMs. Morioka et al.[19] proposed a method called 3D-PIRF based on [1] and [7]. This method is combining omni-vision and tire odometry. It localizes the robot by 8 points algorithm[20] based on matched features and odometry.

Fig. 2.5 shows that in [19], the trajectory's errors are accumulated gradually. [19] detecte loop-closure by PIRF, and correct the trajectory.

2.4 Summary

BoW based PIRF-nav2.0[8] is fast and accurate.

Because it match and index features only on basis of n-dimensional nearest neighbors, it is called pure bag-of-words (BoW) methods. Because of the reason described in Fig. 1.2, PIRF-nav2.0[8] is hard to improve the performance anymore.

In addition, [19]'s hybrid map can work in dynamic environments. However, because of the same reason described in Fig. 1.2, its loop closure detected ratio are not satisfied. It is based on PIRF[7], so it is slow.

My proposed method is intend to improve appearance-only and hybrid SLAM's performance.

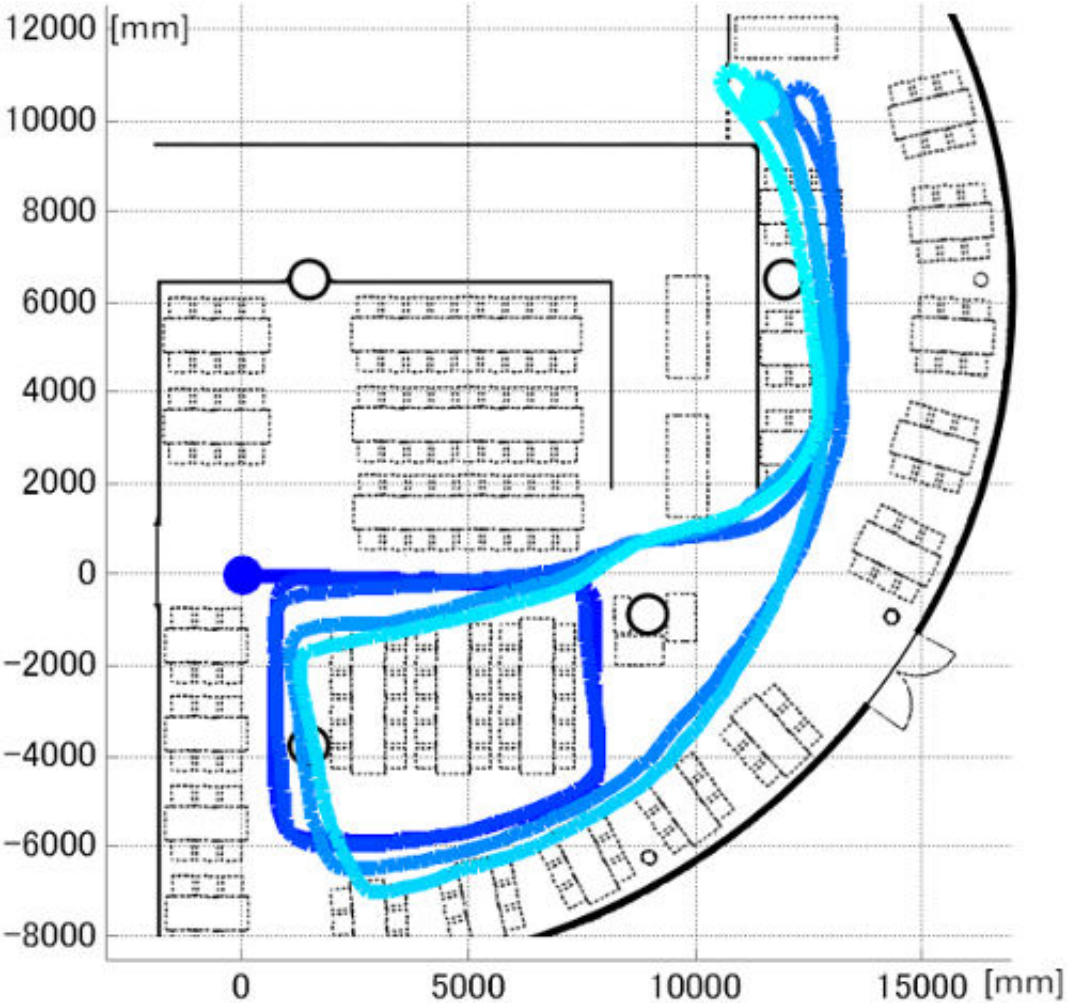


FIGURE 2.5: Trajectory calculated from only odometry in [19]. Errors are accumulated gradually.

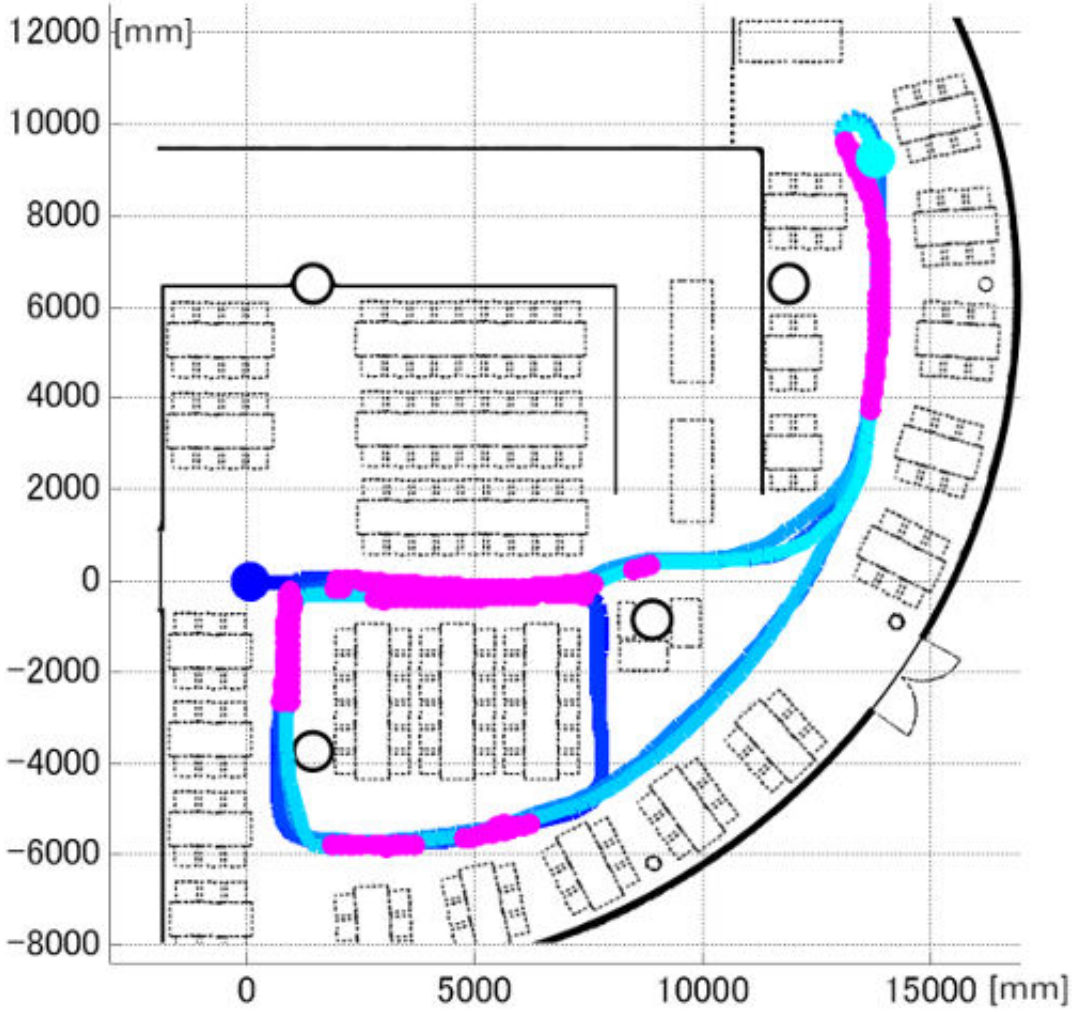


FIGURE 2.6: Trajectory learned using [19]. The trajectory was modified.

Chapter 3

Image Features

Image feature's robustness is critical to vision based SLAMs. In this Chapter, I introduce 3 kinds of image features, including Scale invariant feature transform(SIFT), Speed Up Robust Features(SURF) and Position Invariant Feature(PIRF).

SIFT and SURF are scale and rotation invariant features. PIRF is a kind of robust feature for SLAM in dynamic environment.

3.1 Scale invariant feature transform(SIFT)

Scale invariant feature transform(SIFT)[3] is a computer vision algorithm, it is used to detect and describe the local features of image, it seeks extreme point in the space scale, and extract its scale and rotation invariant descriptor. The algorithm is proposed by David Lowe in 1999.

Its application areas include object recognition, perception and robot map navigation, image stitching, 3d modeling, gesture recognition, image tracking etc.

SIFT is robust to light, noise and small angle change. Because of its robustness, SIFT descriptor can match with large database correctly. SIFT feature based object detection rate is quite high. Because SIFT features' information is plenty, SIFT is suitable for match in large databases. Thus, there are many kinds of BoW SLAMs based on SIFT.

Lowe's classic SIFT algorithm is decomposed into the following four steps:

1. Detect extreme points in scale space: search images of all scales. By using gaussian function to identify scale and rotation invariant points of interest.

2. Key point positioning: on each candidate location, uses a repeated interpolation algorithm to determine the location. The choices of key points based on each candidate locations' stability.

3. Direction determination : calculate key points neighborhood's gradient direction. Determinate main direction of key points. By determining main direction, key points can be described with rotation invariant characteristic.

4. Key point description: create descriptions of key points based on there neighborhood's gradient direction.

3.1.1 Detect extreme points in scale space

SIFT algorithm is look for key points in the different scales, and scale space need to use the gaussian blur, Lindeberg et has proved the gaussian convolution kernels is the only reasonable transform kernels for scale space.

Scale space was presented by Iijima in 1962 initially, later Lindeberg[21][22] Blanco, J. L. [23] Salden, Alfons H.[24]Mikolajczyk, K. et al.[25] Wang, M. et al. [26] promoted gradually, scale space is used widely in computer vision field.

Scale space theory's basic idea is: introduce a scale parameter to the image information processing model, by changing scale parameter a continuously under the multi-scale

space, main edges of scale space's sequences are extracted, and use main edges as features of image. On the bases of these edges, scale invariant features are extracted.

Scale space is implemented by gaussian pyramid, gaussian pyramid building is divided into two parts:

1. Blur the image by different scales of gaussian filter.

2. Down sample the image.

The 2D gaussian filter is:

$$G(x, y, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-m/2)^2+(y-n/2)^2}{2\sigma^2}} \tag{3.1}$$

Refers to the image pyramid model, the original image is down sampled constantly from big to small. The original image is first layer of the pyramid, new images of down sampling are new layers of the pyramid, a pyramid has n layers:

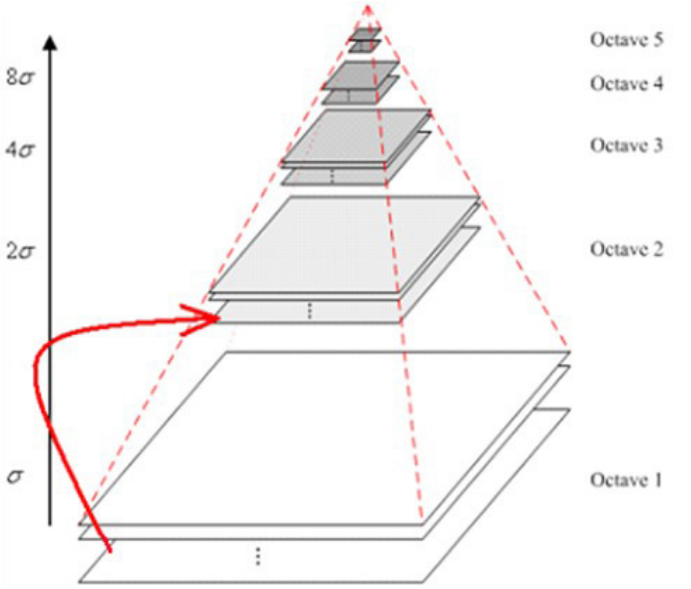


FIGURE 3.1: SIFT pyramid[3].

A image's scale space is defined as convolution between original image and different gaussian filters:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{3.2}$$

* means convolution operation.

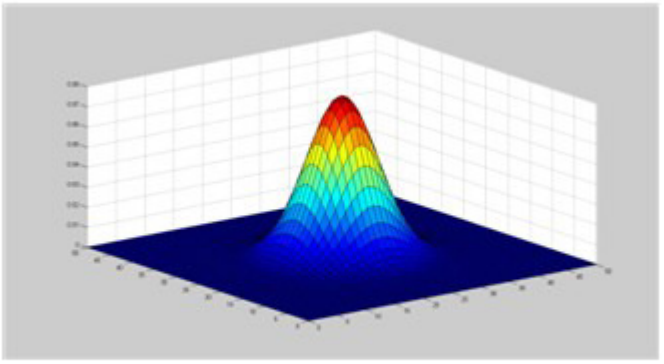


FIGURE 3.2: 2D gaussian filter.

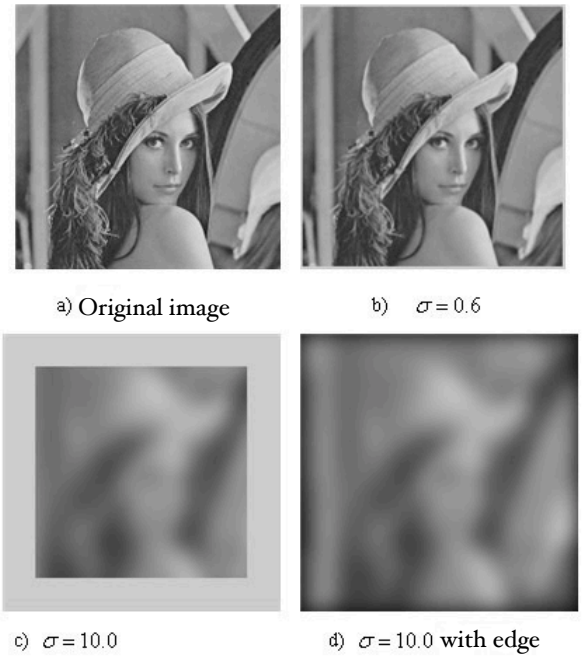


FIGURE 3.3: Images of gaussian blur with different σ.

Points of interest are extracted difference of gaussian(DoG) preliminary.

DoG is calculated by the following formula:

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned} \tag{3.3}$$

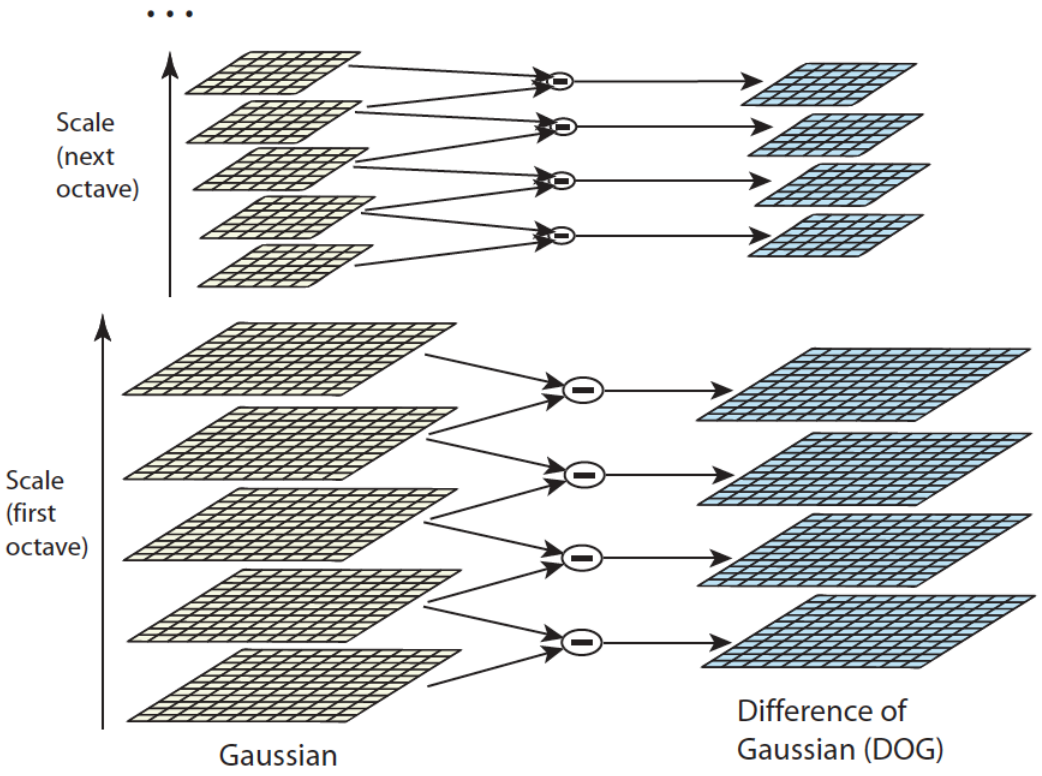


FIGURE 3.4: DoG calculation[3].

Key point is composed of the local extreme value point of the DoG. In order to find DoG extreme points, each pixel point compared to and all of its adjacent points, to test if its value is the highest or lowest. As shown in Fig. 3.5, to ensure that the extreme value is robust, a point must compare with adjacent points and adjacent scales corresponding points. The total amount is 26.

3.1.2 Key point positioning

Although extreme points are extracted difference of gaussian(DoG) preliminary. As shown in Fig. 3.6, not all detected extreme points are real extreme points. More accurate locations of points of interest is extracted based on a repeated interpolation algorithm.

To improve the robustness of extreme points, we must use a curve to fit the DoGs. GoG's taylor expansion is:

$$D(X) = D + \frac{\partial D^T}{\partial X} + \frac{1}{2} X^T \frac{\partial^2 D}{\partial X^2} X \tag{3.4}$$

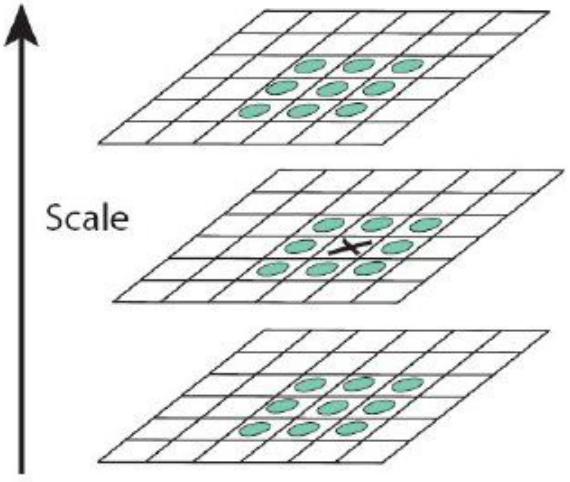


FIGURE 3.5: Sift[3] feature detection based on 26 points.

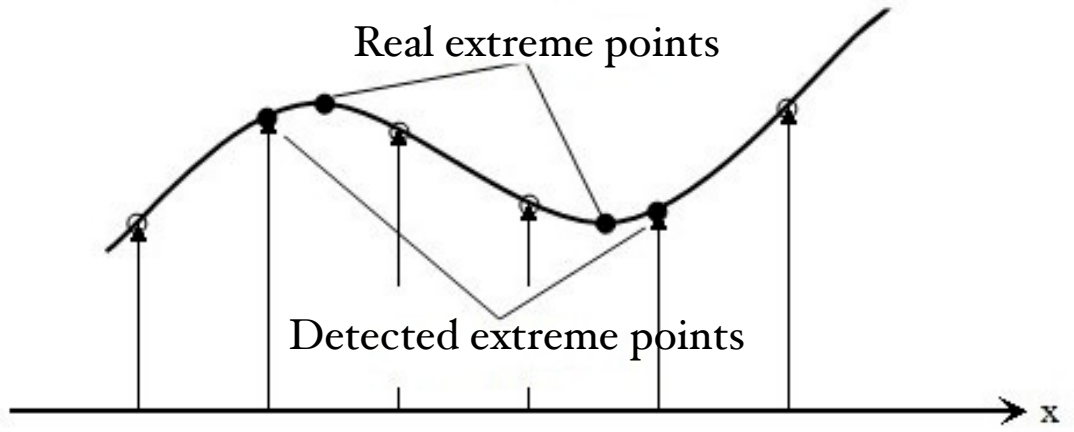


FIGURE 3.6: Some detected extreme points are not real extreme points[3].

Where $X = (x, y, \sigma)^T$. Derivate and let equation equal to zero, offset of detected extreme point is:

$$\hat{X} = -\frac{\partial^2 D^{-1}}{\partial X^2}\frac{\partial D}{\partial X} \tag{3.5}$$

$\hat{X}$ means offset between detected extreme point and real extreme point. When $\hat{X} > 0.5$, the interpolation has been shifted to the center of its neighboring points. So current

extreme point's location must be changed to the new location. Then repeatedly interpolate in the new locations until convergence. Lowe[3] suggested that interpolation should be repeated 5 times.

Besides, DOG operator will produce strong edge response. To eliminate the unstable edge response, calculate points of interest's Hessian matrix, the Hessian matrix H[27][28] is:

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \tag{3.6}$$

α and β are eigenvalues of H, α and β representatives x and y's gradient.

$$\begin{aligned} Tr(H) &= D_{xx} + D_{yy} = \alpha + \beta \\ Det(H) &= D_{xx}D_{yy} - D_{xy}^2 = \alpha + \beta \end{aligned} \tag{3.7}$$

Assuming that α is larger eigenvalue, and β is smaller eigenvalues. Let $\alpha = r\beta$, then

$$\frac{Tr(H)^2}{Det(H)} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(r\beta+\beta)^2}{r\beta^2} = \frac{(r+1)^2}{r} \tag{3.8}$$

D's curvature is proportional to H's eigenvalue, when $\alpha = \beta$ the formula's value is the smallest. Meanwhile, if the formula's value is large, α and β's difference is large. In edge's condition, gradient values in a certain direction is large and in the other direction the gradient value is small. So in order to avoid point on edge, the formula's value must be lower than a threshold.

$$\frac{Tr(H)^2}{Det(H)} < \frac{(r+1)^2}{r} \tag{3.9}$$

In lowe's paper, let $r = 10$.

By using finite difference method based on Taylor expansion, D's derivatives can be calculate by refereeing to neighborhoods conveniently.(Fig. 3.7 and Fig. 3.8)

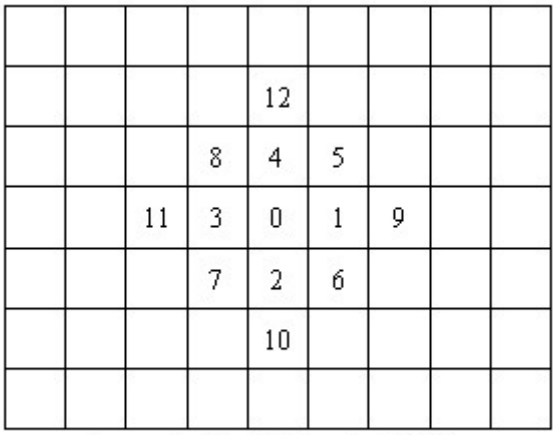


FIGURE 3.7: Finite difference method neighborhoods' value indexes.

$$\left(\frac{\partial f}{\partial x}\right)_0 = \frac{f_1 - f_3}{2h}$$

$$\left(\frac{\partial f}{\partial y}\right)_0 = \frac{f_2 - f_4}{2h}$$

$$\left(\frac{\partial^2 f}{\partial x^2}\right)_0 = \frac{f_1 + f_3 - 2f_0}{h^2}$$

$$\left(\frac{\partial^2 f}{\partial y^2}\right)_0 = \frac{f_2 + f_4 - 2f_0}{h^2}$$

$$\left(\frac{\partial^2 f}{\partial x \partial y}\right)_0 = \frac{(f_8 + f_6) - (f_5 + f_7)}{4h^2}$$

$$\left(\frac{\partial^4 f}{\partial x^4}\right)_0 = \frac{1}{h^4}[6f_0 - 4(f_1 + f_3) + (f_9 + f_{11})]$$

$$\left(\frac{\partial^4 f}{\partial y^4}\right)_0 = \frac{1}{h^4}[6f_0 - 4(f_2 + f_4) + (f_{10} + f_{12})]$$

FIGURE 3.8: Approximate derivatives calculation based on Taylor expansion.

3.1.3 Direction determination

In order to make the descriptor rotation invariant, the main direction of feature points must be found.

After the calculate adjacent points gradient, a direction histogram is established. Direction histogram's 0 360 degrees is divided into 36 columns (bins), each column is 10 degrees. As shown in FFfigure 5.1, the direction histogram's peak represents the main direction of the key points.(Fig. 3.9)

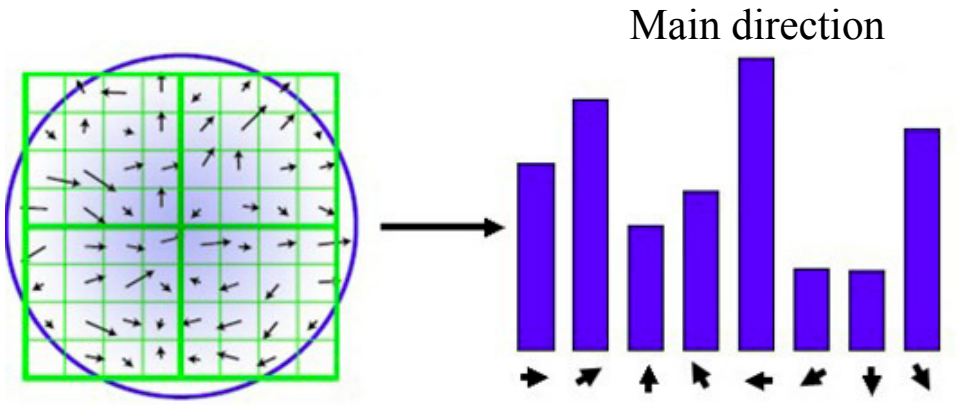


FIGURE 3.9: Direction histogram of SIFT[3].

For descriptor's characteristic of rotation invariant, the feature point's coordinate is rotated based on main direction.(Fig. 3.10)

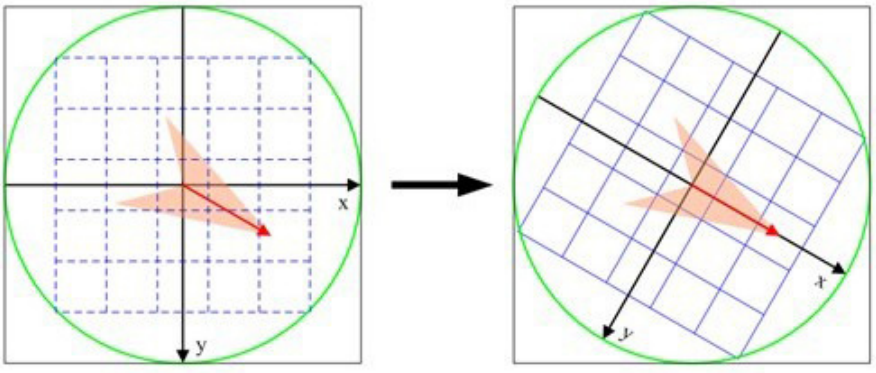


FIGURE 3.10: Main direction rotation[3].

3.1.4 Key point description

Through the above steps, for each key point, has three information: the position, scale and direction. Next is to build a descriptor for each key point.

Many kinds of descriptors has been proposed, such as Gaussian derivatives [29], moment invariants [29], complex features [30][31], steerable filters [32], phase-based local features [33].

The latter, Lowe [3] proposed a method performs better than the others [34].

SIFT uses a set of vectors to describe key points. For robustness, this descriptor not only include key points, also contains points around the pixels.

Lowe advice that the descriptor had batter to use 4 * 4 key window in the scale space to calculate eight direction of gradient information. The descriptor is a 4 * 4 * 8 = 128 dimensional vector.(Fig. 3.11)

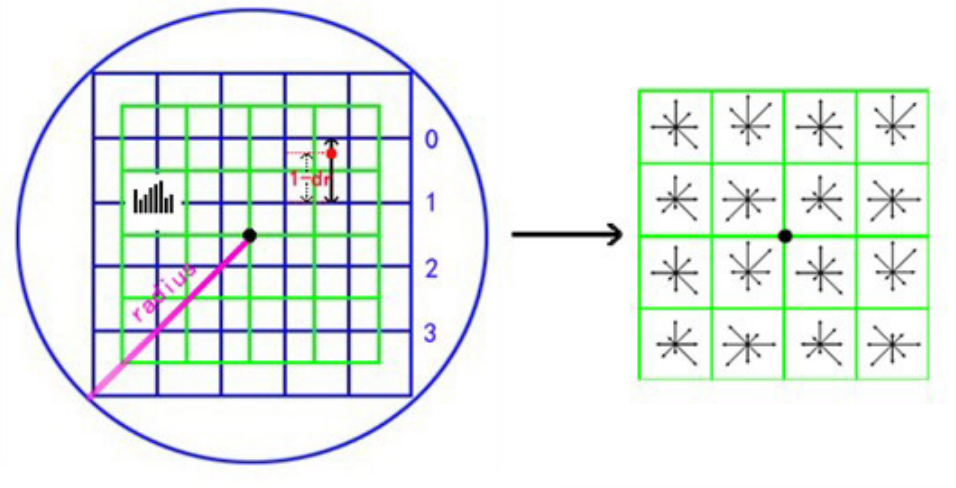


FIGURE 3.11: SIFT's 128 dimensional descriptor[3].

3.1.5 Key point matching

Since SIFT descriptor is scale and rotation invariant, SIFT's matching is very robust. Although a scene is resized and rotated, SIFT can match them effectively.

The SIFT descriptor is vector, so SIFT's matching is finding nearest neighbor of each descriptor. The descriptor matching and indexing process uses a modification of the k-d tree algorithm called the Best-bin-first[5] search method that can identify the nearest neighbors with high probability using only a limited amount of computation.

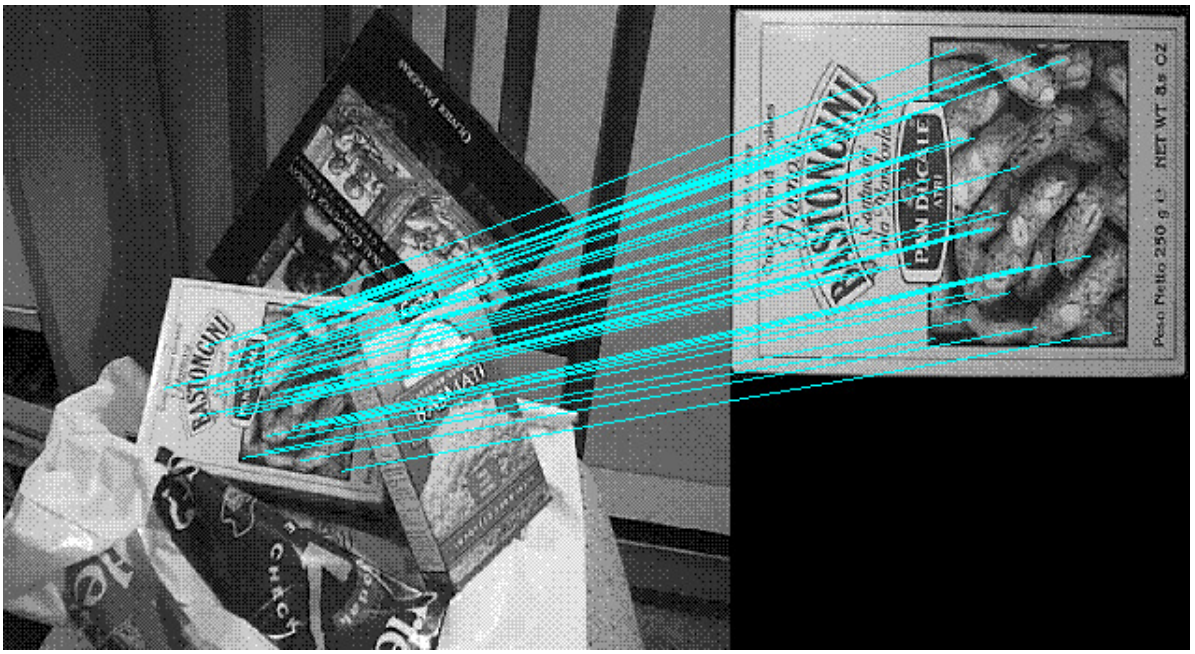


FIGURE 3.12: The appearance of SIFT[3]'s matching. SIFT is a kind of scale-invariant and rotation-invariant feature.

3.2 Speed Up Robust Features(SURF)

SURF (Speed Up Robust Features)[4] is another kind of scale-invariant and rotation-invariant feature. Similar to SIFT, SURF is based on multi-scale space theory and the feature detector is also based on Hessian matrix.

SURF creates a pyramid (Fig. 3.13) without 2:1 down sampling. Due to the use of integral images, SURF filters the stack using a box filter approximation of second-order Gaussian partial derivatives.

Since SURF achieves gaussian blur and hessian matrix calculation at the same based on approximation of second-order gaussian partial derivatives and SURF do not need to down sampling the original image, SURF's extraction is 3 times faster than SIFT. Fig. 3.14 and 3.15 Shows the Gaussian second order partial derivatives in y-direction and xy-direction.

SURF descriptor and its matching is based on similar properties to SIFT. But SURF use 64 dimensional vector as descriptor. So SURF's matching is faster than SIFT too.

Experiment results in paper show that SURF is as effective as SIFT. So SURF is a fast and ideal candidate for SLAM. The proposed method choose SURF as feature point.

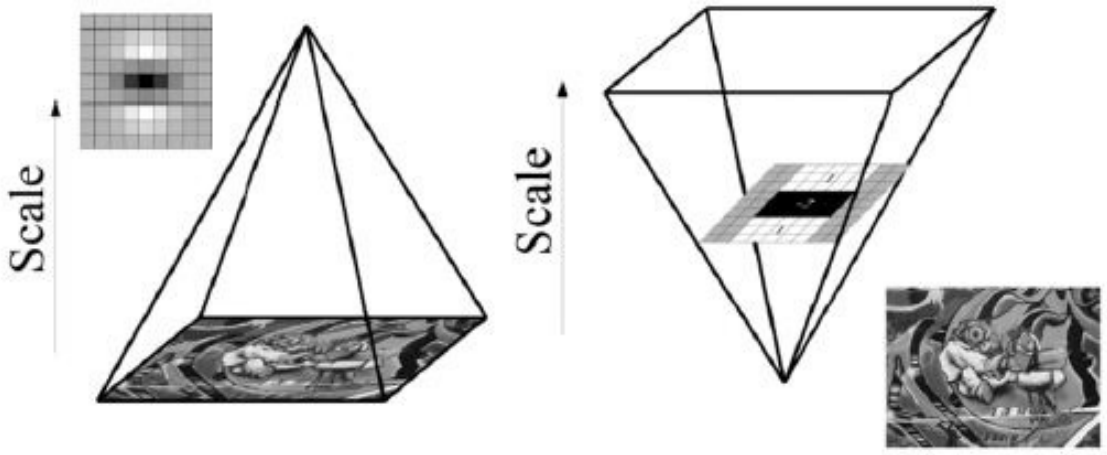


FIGURE 3.13: SURF[4]'s scale space pyramid.

3.3 Position Invariant Robust Feature(PIRF)

Because of scale and rotation invariant characteristic, for object recognition, SIFT[3] and SURF[4] work effectively.

However, the appearance of objects in the actual world is always dynamic. Many appearance-only SLAM methods are based on the hypothesized static environment. These methods use SIFT[3] or SURF[4].

These local-visual features do not avoid moving objects such as walking humans in cafeterias, stations, or shopping malls. Features on walking humans is useless and harmful to localization.

For appearance-only SLAM, the visual features' robustness and the effectiveness of the matching are important.

A. Kawewong *et al.*[1] proposed a method that tracks robust features in a sequence of images, called position-invariant robust features (PIRF). PIRF extracts features by referring to past images based on SURF or SIFT. So it is more robust than original SIFT or SURF in dynamic environments.

PIRF's extraction is illustrated in Fig. 3.17.

By using PIRF, dynamic component of environment can be ignored.(Fig. 3.18)

3.4 Summary

SIFT and SURF are scale and rotation invariant features. But they are not robust against changing of environment appearance. PIRF is robust against dynamic components of environment. However because of the reason described in Fig. 1.2, PIRF is not robust enough. Besides other problems of PIRF will be descried in chapter 4.

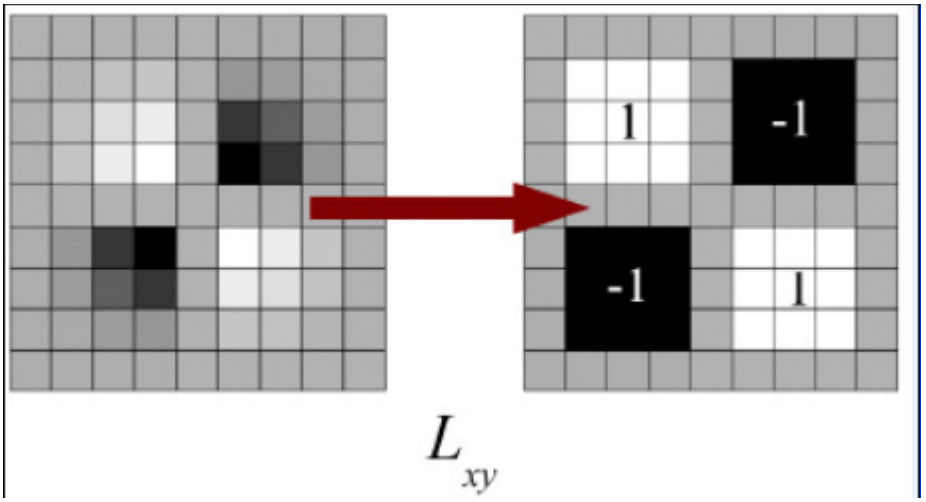


FIGURE 3.14: The approximate gaussian second order partial derivatives in xy-direction[4].

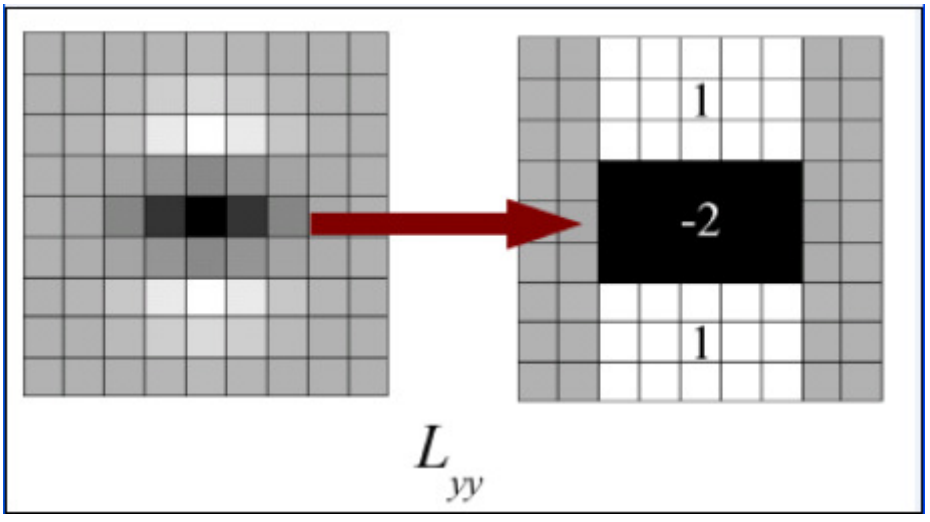


FIGURE 3.15: The approximate gaussian second order partial derivatives in y-direction[4].

FIGURE 3.16: Crowed station. Because there were many walking humans, this scene is very noisy for SLAM.

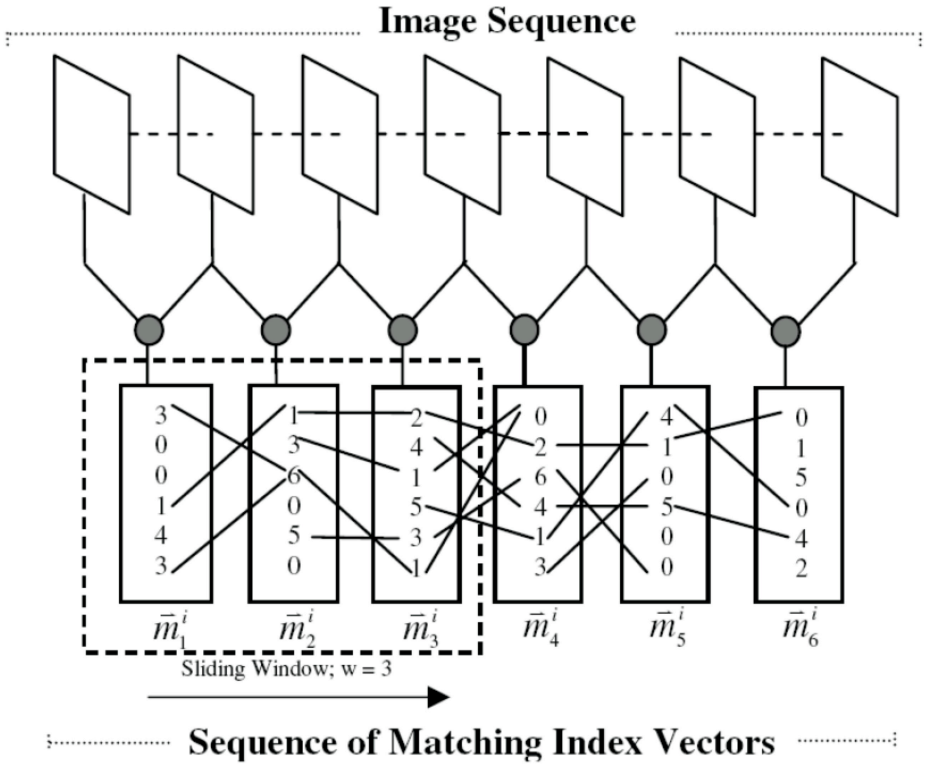


FIGURE 3.17: [1]'s extraction. Every image pair is compared using feature matching, resulting in six matching index vectors. A vector element is the index of the corresponding feature in the next image. For example, for the first sub-place $(\vec{m}_1^i, \vec{m}_2^i, \vec{m}_3^i)$of I_1, I_2, I_3, I_4there are only three features appearing in all images: (1,3,6,1), (4,1,1,2),(6,3,6,1). (1,3,6,1) is interpreted, respectively, as the 1st, 3rd, 6th, and 1st feature of image I1, I2, I3, I4. These four features are interpolated to obtain a single representative PIRF. Therefore, there would be 3, 4, 4, 3 PIRFs for the 1st,2nd, 3rd, and 4th sub-place respectively, 14 PIRFs in all for the whole i^{th} place.

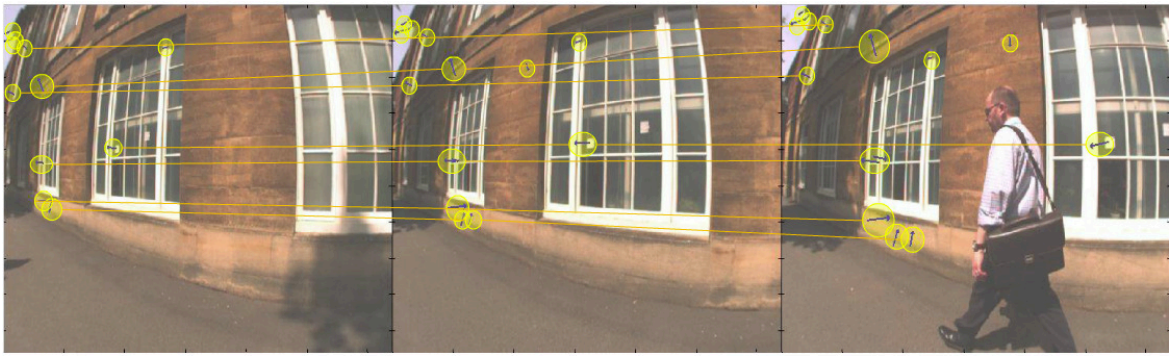


FIGURE 3.18: Yellow Points are PIRFs, human is ignored by PIRF[1].

Chapter 4

Incremental Center of Gravity Matching Based SLAM

In this chapter, i present a highly accurate visual-feature-extracting method, which can be applied to SLAM and navigation even in highly dynamic environments.

The details of the proposed method are described.

4.1 Extraction of Robust Features

We choose SURF [4] as local visual feature. We use a method called Incremental Center of Gravity Matching (ICGM) based on local visual feature points' relative geometry structure to extract robust feature.

4.1.1 Definition of Robust Feature

The definition of robustness of a feature differs according to its application. In this study, the objective is location recognition. With respect to PIRF [1], we define a robust feature as a position-invariant feature. These features should satisfy the following condition:

First, the definition of the environment corresponds to the entire visual field of the camera. The static component of the environment will not change in the long term. For instance, in a railway station, there are many pedestrians. Because the features of these pedestrians change rapidly, they are not considered to be the features of the static component of the environment. Features that appear on places such as walls and billboards will not change in the long term. Therefore, these features can be treated as robust features.

4.1.2 Incremental Center of Gravity Matching

ICGM is a method that avoids dynamic visual-features and incorrect matches effectively. In this section, first, we describe the relationship between a camera's descriptive geometry and dynamic environment; then, we elucidate this method in detail.

Camera's Descriptive Geometry and Dynamic Environment From descriptive geometry:

$$q = MQ,\ where\ q = \begin{bmatrix} x \\ y \\ w \end{bmatrix},\ M = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix},\ Q = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{4.1}$$

where q represents the homogeneous coordinates of the camera, Q is a 3D point in the real world, and M is the camera intrinsic matrix.

The camera movement is assumed to have six degrees of freedom. Assuming that from time t_0 to t_1, the camera moved by $\Delta X_0,\ \Delta Y_0,\ and \Delta Z_0$ and rotated at the angles of

$\theta_x^0,\ \theta_y^0, and\ \theta_z^0$. The angles $\theta_x^0,\ \theta_y^0, and\ \theta_z^0$ are the initial angles of rotation around the x-, y-, and z-axes, respectively. The rotation matrix of the camera is

$$R_0 = R_x(\theta_x^0) \cdot R_y(\theta_y^0) \cdot R_y(\theta_y^0) \tag{4.2}$$

Where,

$$R_x(\theta_x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\theta_x & -sin\theta_x \\ 0 & sin\theta_x & cos\theta_x \end{bmatrix}$$

$$R_y(\theta_y) = \begin{bmatrix} cos\theta_y & 0 & sin\theta_y \\ 0 & 1 & 0 \\ sin\theta_y & 0 & cos\theta_y \end{bmatrix} \tag{4.3}$$

$$R_z(\theta_z) = \begin{bmatrix} cos\theta_z & -sin\theta_z & 0 \\ sin\theta_z & cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The images I_0 and I_1 are captured at t_0 and t_1, respectively. Two visual-feature points on I_0 and I_1 are q_0 and q_1 and $q_0{}'$ and $q_1{}'$, respectively . The points q_0,$q_0{}'$ and q_1,$q_1{}'$ are the same in the real world but are observed at different times. The points q_0 and q_1 are related to $q_0{}'$ and $q_1{}'$, respectively.

$$q_0 = M \cdot \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix},\ q_1 = M \cdot \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} \tag{4.4}$$

The points q_0 and q_1 are assumed to be the visual-feature points of the static component of the environment. Because the camera moved by $\Delta X_0,\ \Delta Y_0, and\ \Delta Z_0$ and rotated around $\theta_x^0,\ \theta_y^0, and\ \theta_z^0$ in the environment, the following equation can be obtained:

$$q_0' = M \cdot R_0 \cdot \begin{bmatrix} X_0 - \Delta X_0 \\ Y_0 - \Delta Y_0 \\ Z_0 - \Delta Z_0 \end{bmatrix},\ q_1' = M \cdot R_0 \cdot \begin{bmatrix} X_1 - \Delta X_0 \\ Y_1 - \Delta Y_0 \\ Z_1 - \Delta Z_0 \end{bmatrix} \tag{4.5}$$

We define the relative vector $Diff$ as follows:

$$Diff_{ij} = \frac{q_i}{w_i} - \frac{q_j}{w_j} = \begin{bmatrix} \frac{x_i}{w_i} \\ \frac{y_i}{w_i} \\ 1 \end{bmatrix} - \begin{bmatrix} \frac{x_j}{w_j} \\ \frac{y_j}{w_j} \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{x_i}{w_i} - \frac{x_j}{w_j} \\ \frac{y_i}{w_i} - \frac{y_j}{w_j} \\ 0 \end{bmatrix} \tag{4.6}$$

The relative vector means two feature points' relative position's vector.

The 2D coordinates of the visual-feature points on the image are denoted by $x/w, y/w$.

$$p = \begin{bmatrix} \frac{x}{w} \\ \frac{y}{w} \end{bmatrix} \tag{4.7}$$

The local visual-feature point on the image is denoted by p, and the 2D visual-feature points' relative vector v is defined as follows:

$$v_{ij} = p_i - p_j = \begin{bmatrix} \frac{x_i}{w_i} - \frac{x_j}{w_j} \\ \frac{y_i}{w_i} - \frac{y_j}{w_j} \end{bmatrix} \tag{4.8}$$

First, assuming that the camera did not move from t_0 to t_1,

$\Delta X_0,\ \Delta Y_0,\ \Delta Z_0,\ \theta_x^0,\ \theta_y^0,\ \theta_z^0 = 0$

$$v_{01} = p_0 - p_1 = v_{0'1'} = p_0' - p_1' \tag{4.9}$$

Then, assuming that the camera moved only slightly from t_0 to t_1,

$$(\Delta X_0,\ \Delta Y_0,\ \Delta Z_0) \to (0,\ 0,\ 0),\ R_0 \to \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.10}$$

Thus

$$v_{01} - v_{0'1'} \to \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{4.11}$$

Eq. (4.11) shows that after the viewing location and angle changed slightly, the static component of the environment's feature points should maintain their approximate relative geometric structure.

It is assumed that Q_2 is a 3D point that appears on the dynamic component of the environment from t_0 to t_1. With respect to the camera, Q_2 has moved to $\Delta X_1,\ \Delta Y_1,\ \Delta Z_1$ and $(\Delta X_1,\ \Delta Y_1,\ \Delta Z_1) \not\to (0,\ 0,\ 0)$. The corresponding visual-feature points of Q_2 on I_0 and I_1 are q_2 and q_2', respectively.

$$q_2 = M \cdot \begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \end{bmatrix},\ q_2' = M \cdot R_0 \cdot \begin{bmatrix} X_2 - \Delta X_1 \\ Y_2 - \Delta Y_1 \\ Z_2 - \Delta Z_1 \end{bmatrix} \tag{4.12}$$

Beacuse $(\Delta X_1,\ \Delta Y_1,\ \Delta Z_1) \not\to (0,\ 0,\ 0)$

$$v_{02} - v_{0'2'} \not\to \begin{bmatrix} 0 \\ 0 \end{bmatrix},\ v_{12} - v_{1'2'} \not\to \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{4.13}$$

Therefore, in common

$$\|v_{02} - v_{0'2'}\|,\ \|v_{12} - v_{1'2'}\| > \|v_{01} - v_{0'1'}\| \tag{4.14}$$

Eq. (4.14) shows that after the viewing location and angle changed slightly, there are different changes to the static and dynamic points of the relative vector of the 2D visual-feature points. The relative vector of the static points changed slightly, whereas the relative vector between the static and dynamic points changed considerably. Eq. (4.14) summarizes the basic concept of this study. These static feature points are considered as the robust features of a dynamic environment.

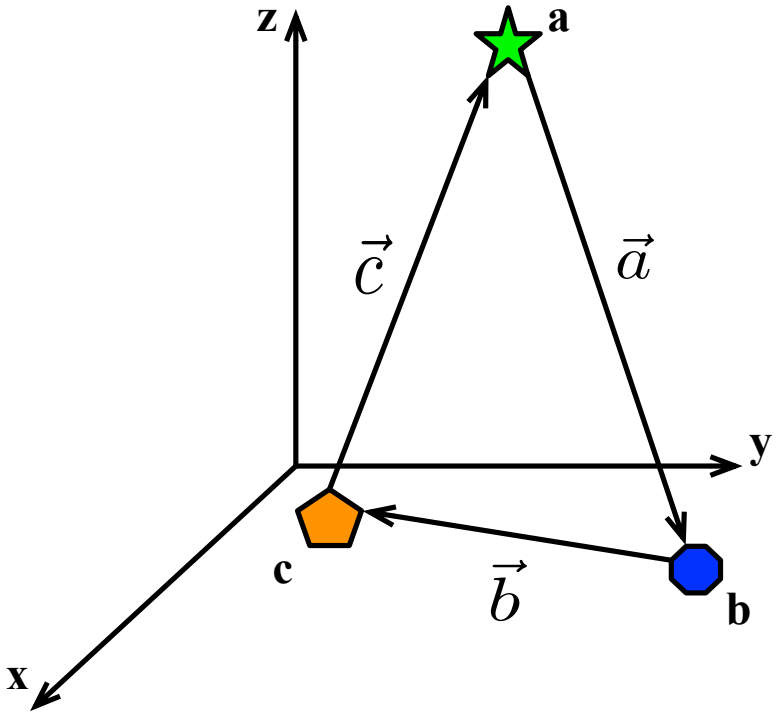
Although Eq. (4.14) was only proved in a condition in which the camera moved slightly, in actual situations, the requirement that the camera "moves slightly" is not so tight.

From Fig. 4.1, we observe idea of this study simply.

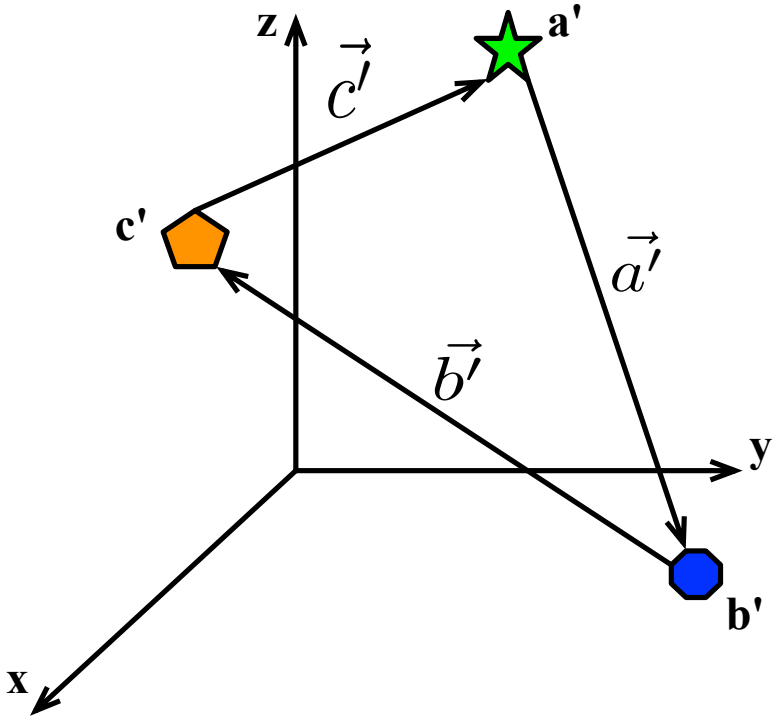
Incremental Center of Gravity Matching (ICGM) Initially, on the basis of Eq. (4.7), we define the center of gravity as follows:

$$CG_N = \begin{bmatrix} \mathrm{X} \\ \mathrm{Y} \end{bmatrix} = \frac{1}{N} \sum_{i=0}^{N} p_i \tag{4.15}$$

This center of gravity for a set of local feature points on the image. CG includes the geometric structure of feature points. The proposed method involves the use of the



(a) Relative position at time point t-1



(b) Relative position at time point t

FIGURE 4.1: Assume that we captured two images at time point t-1 and t, during this duration camera only moved very slightly. Points a,b,c indicate locations of visual features with respect to the environment, fig. (a) shows their relative position at time point t-1, $\vec{a}$, $\vec{b}$, $\vec{c}$ are their relative vectors. At time point t, in fig. (b), feature c has changed its position. We can obtain $\vec{a} \approx \vec{a'}$, $\vec{b} \neq \vec{b'}$, $\vec{c} \neq \vec{c'}$

center of gravity to extract robust visual-features in dynamic environments. We define the center of gravity vector as follows:

$$CGV_{Nx} = CG_N - p_x = \frac{1}{N}\sum_{i=0}^{N}(p_i - p_x) \tag{4.16}$$

In Eq. (4.16) [refer to Eq. (4.8)], $p_i - p_x$ is the relative vector of the 2D visual-feature points. By using (16), we can calculate the average of these relative vectors between a feature point and set of feature points.

It is assumed that at t_0, $\mathrm{p^i} = (p_0,\ p_1,\ ...p_i)$ is a set of robust visual-feature points on I_0, and CG_{i0} is the center of gravity of $\mathrm{p^i}$. The corresponding set at t_1 on I_1 is $\mathrm{p^{i'}} = (p'_0,\ p'_1,\ ...p'_i)$, and CG_{i1} is the center of gravity of $\mathrm{p^{i'}}$. The terms $\mathrm{p^i},\ \mathrm{p^{i'}}$ are a part of the set of entire feature points, but it does not include all robust visual-feature points. Then, we can test whether or not the remaining points are the robust visual features of the environment.

For example, $p_j,\ p'_j$ is a pair of robust visual-feature points at t_0 and t_1, which are excluded from $\mathrm{p^i},\ \mathrm{p^{i'}}$. By referring to Eqs. (11) and (16), the following Eq. can be obtained:

$$CGV_{i0j} - CGV_{i1j'} = (CG_{i0} - p_j) - (CG_{i1} - p'_j) \rightarrow \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{4.17}$$

Moreover, $p_k,\ p'_k$ is a assumed to be a pair of unstable visual-feature points at t_0 and t_1, which are also excluded from $\mathrm{p^i},\ \mathrm{p^{i'}}$. By referring to Eqs. (13) and (16), the following equation can be obtained:

$$CGV_{i0k} - CGV_{i1k'} = (CG_{i0} - p_k) - (CG_{i1} - p'_k) \not\rightarrow \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{4.18}$$

We define the ratio of difference RoD as follows:

$$RoD_{(x,y)} = \frac{\|CGV_x - CGV_y\|}{\|CGV_x\| + \|CGV_y\|} \tag{4.19}$$

The result

$$RoD_{(i0k,i1k')} > RoD_{(i0j,i1j')} \tag{4.20}$$

is similar to (14)

On the basis of Eq. (4.20), we can set a threshold Thr to distinguish between robust and unstable visual features. If the center of gravity of a vector pair satisfies $RoD <= Thr$, the feature point is considered as a robust visual-feature point. If the center of gravity of a vector pair satisfies $RoD > Thr$, the feature point is considered as an unstable visual-feature point.

Eq. 4.19 is robust to the changing of relative vectors' length.

We proposed a method to extract robust visual features on the basis of Eq. (4.20). Because the center of gravity changed during processing, the proposed method is called as ICGM, and its algorithm is shown as Algorithm1.

4.1.3 Single- and Double-directional ICGMes

Using ICGM can extract robust features from a sequence of images. In some ways, ICGM is similar to PIRF [1]. Fig. 4.2 shows the Single-Directional robust feature extraction method based on ICGM which extracts robust features from I_t and I_{t-1}. It means matches obtained by ICGM between I_t and I_{t-1} will be extracted as robust features of I_t and I_{t-1}. Therefore, the robust features of I_t can be extracted by referring to I_{t-1}. These features are more robust than those extracted only from I_t. However, it also causes a significant loss of features. In particular, while using a conventional single-lens camera, the loss is high. Fig. 4.3 shows a method that is different from that of the Single-Directional ICGM. It is called double-directional ICGM. Double-directional ICGM extracts robust features of I_t not only by referring to I_{t-1}, but also by referring to I_{t+1}. In Section 2.1.4, we show certain factors that affect the extraction of ICGM while using a conventional single-lens camera and prove that the double-directional ICGM works more effectively.

4.1.4 Factors affect extraction of ICGM

Several factors affect the extraction of ICGM, such as the camera's speed of motion and the size of ICGM extraction window.

Algorithm 1: Incremental Center of Gravity Matching's Algorithm

Input: Match feature points between I_0 and I_1: $\mathrm{p} = (p_0,\ p_1,\ ...p_n)$, $\mathrm{p}' = (p'_0,\ p'_1,\ ...p'_n)$ and ICGM's matching threshold Thr

Output: Robust feature points' set on I_0 and I_1: $\mathrm{p_R},\ \mathrm{p'_R}$

$CG0, CG1, CGV_0, CGV_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$;

$\mathrm{p_R},\ \mathrm{p'_R} = \Phi$;

$GetGoodSet = false$;

$S_{CG} = 0$;

`/* Get initial two robust feature points */`

while $GetGoodSet$ is $false$ **do**

  $i,\ j = 0$;

  while $i == j$ **do**

    $i \leftarrow$ random int between 0 and n;

    $j \leftarrow$ random int between 0 and n;

  end

  $CGV_0 \leftarrow p_i - p_j$;

  $CGV_1 \leftarrow p'_i - p'_j$;

  if $RoD_{(0,1)} <= Thr$ **then**

    $GetGoodSet \leftarrow true$;

    $CG0 \leftarrow \frac{1}{2}(p_i + p_j)$;

    $CG1 \leftarrow \frac{1}{2}(p'_i + p'_j)$;

    $S_{CG} \leftarrow 2$;

    delete $p_i,\ p_j$ from p;

    delete $p'_i,\ p'_j$ from p′;

  end

end

`/* Test remaing feature points in p and p′ */`

for $k = 0; k \leq n-2; k \neq i$ **do**

  $CGV_0 \leftarrow CG0 - p_k$;

  $CGV_1 \leftarrow CG1 - p'_k$;

  if $RoD_{(0,1)} <= Thr$ **then**

    $CG0 \leftarrow \frac{S_{CG}}{S_{CG}+1}(CG0 + \frac{1}{S_{CG}}p_k)$;

    $CG1 \leftarrow \frac{S_{CG}}{S_{CG}+1}(CG1 + \frac{1}{S_{CG}}p'_k)$;

    $S_{CG} \leftarrow S_{CG} + 1$;

    insert p_k to $\mathrm{p_R}$;

    insert p'_k to $\mathrm{p'_R}$;

  end

end

return $\mathrm{p_R},\ \mathrm{p'_R}$;

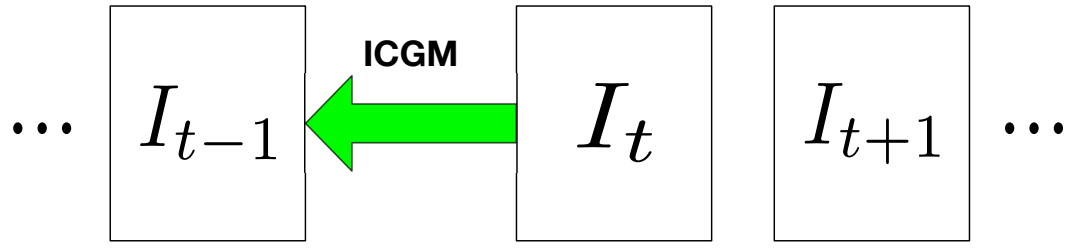


FIGURE 4.2: Single-Directional ICGM: Single-directional ICGM. The single-directional ICGM only extracts robust features from I_t and I_{t-1}.

We mention two situations which use a conventional single-lens camera: the camera rotates at a certain speed, and it approaches or moves away from an infinite object at a certain speed.

For these two typical situations, the double-directional ICGM is better than a single-directional ICGM.

Camera rotating at a certain speed In this situation, we assume that the angular velocity of the camera is denoted by ω, and the camera's field-of-view is denoted by the angle γ. Therefore, the interval at which the camera views a completely different scene is $T_{Disappear} = \gamma/\omega$. We define $T_{Duration}$ as the duration of extraction using the single-directional ICGM (duration from $t-1$ to t). We also assume that the points of static features approximately follow a uniform distribution.

In this situation, the ratio of features extracted by the single-directional ICGM P_α is given as follows:

$$P_\alpha = \begin{cases} \frac{T_{Disappear} - T_{Duration}}{T_{Disappear}} = 1 - \frac{T_{Duration}}{T_{Disappear}} & T_{Duration} \leq T_{Disappear} \\ 0 & T_{Duration} > T_{Disappear} \end{cases} \tag{4.21}$$

The ratio of features extracted using the double-directional ICGM P_β is given as follows:

$$P_\beta = \begin{cases} 1 & T_{Duration} < \frac{1}{2} \cdot T_{Disappear} \\ 2 \cdot P_\alpha & T_{Duration} \geq \frac{1}{2} \cdot T_{Disappear} \end{cases} \tag{4.22}$$

To compare the two approaches, we define a parameter λ_α as

$$\lambda_\alpha = \frac{T_{Duration}}{T_{Disappear}} \tag{4.23}$$

Therefore, $P_\alpha = 1 - \lambda_\alpha$.

From Fig. 4.4, we observe that the double-directional ICGM can extract more static visual-feature points than those by the single-directional ICGM. In particular, when $\lambda_\alpha \leq \frac{1}{2}$, the double-directional ICGM does not lose any static visual-feature points.

Camera approaching or moving away from an infinite object at a certain speed In this situation, we use the same definition of $T_{Duration}$ as that mentioned

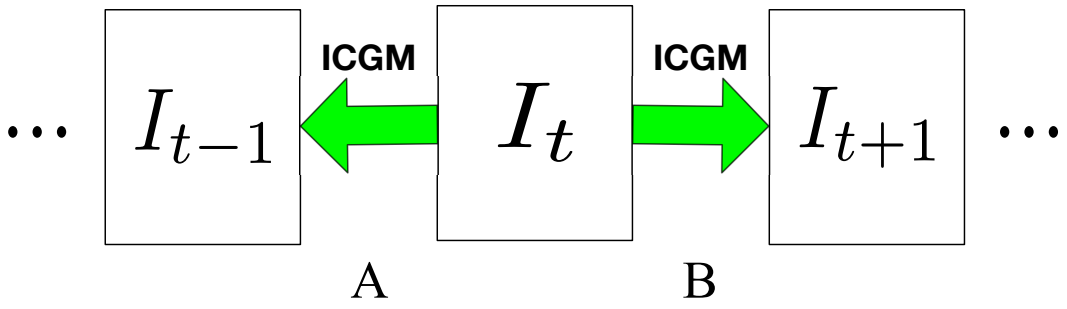


FIGURE 4.3: Double-directional ICGM. The double-directional ICGM extracts robust features from I_t, I_{t-1}, and I_{t+1}. It is assumed that A and B are the set of features extracted from I_t and I_{t-1} and I_t and I_{t+1}, respectively. Then, the robust features extracted using the double-directional ICGM are C = A ∪ B.

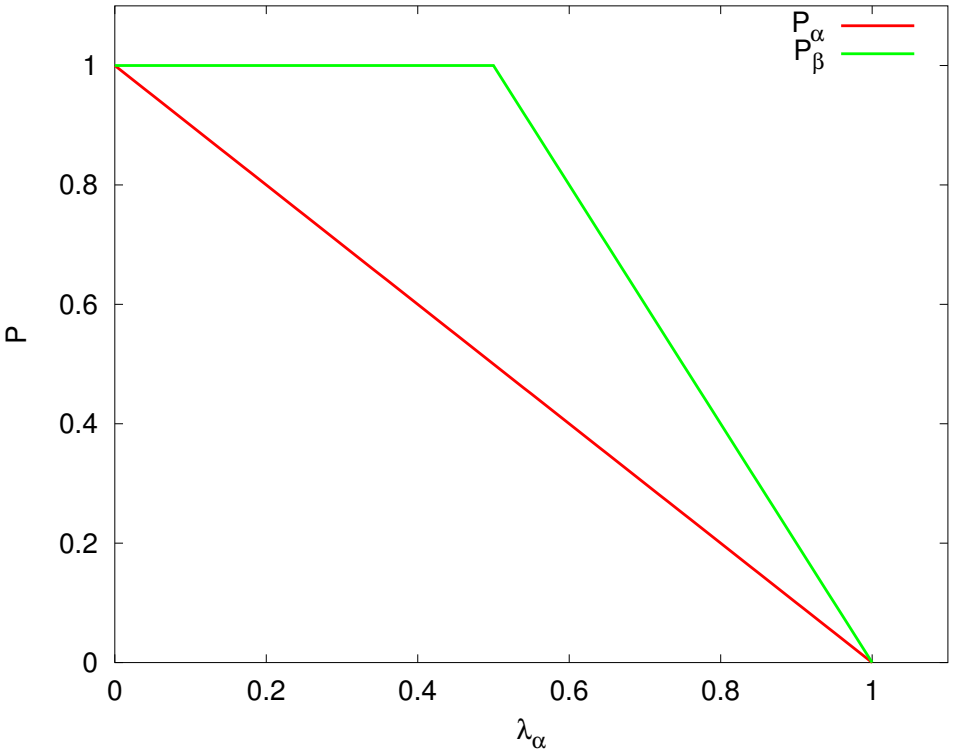


FIGURE 4.4: Comparison between Double and Single-directional ICGMs when the camera rotates at a certain speed: P_α is the ratio of features extracted by the single-directional ICGM. P_β is the ratio of double-directional ICGM. Double-directional ICGM always loses fewer features than single-directional ICGM.

above. We assume that the vertical and horizontal fields-of-view are equal to η and θ, respectively. In addition, the static feature points follow a uniform distribution.

Therefore, when the camera moves away from a sufficiently large infinite object with speed ν, assuming that at time t, the distance between the camera and the object is d, the area at this time is given as s, where

$$s = 4 \cdot tan(\eta)tan(\theta) \cdot d^2 \tag{4.24}$$

Before the duration for ICGM, $T_{Duration}$, at time $t - T_{Duration}$, the distance between the camera and the object was $d' = d - T_{Duration} \cdot \nu$. Then, the field-of-view for time $t - T_{Duration}$ is

$$s' = 4 \cdot tan(\eta)tan(\theta) \cdot (d - T_{Duration} \cdot \nu)^2 \tag{4.25}$$

Therefore, in this situation

$$P_\alpha = \frac{s'}{s} = \frac{(d - T_{Duration} \cdot \nu)^2}{d^2} \tag{4.26}$$

Similarly, we define

$$\lambda_\beta = \frac{T_{Duration} \cdot \nu}{d} \tag{4.27}$$

Because the single-directional ICGM only generalizes the information of the past, a part of the field-of-view is ignored when the camera moves away from the object, i.e., $P_\alpha = (1 - \lambda_\beta)^2$. The double-directional ICGM generalizes the information of both the past and future. Because the field of view of a camera viewpoint is completely included by that of the other viewpoints (including past and feature), the field-of-view loss will be zero; therefore, in all cases, $P_\beta = 1$.

When the camera approaches an infinite object, because the camera previously had a larger field-of-view, $P_\alpha = 1$ $P_\beta = 1$. Fig. 4.5 shows the comparison between the double and single-directional ICGMes in this situation.

Discussion of the two methods We developed an improved visual-feature-extraction method for ICGM called as the double-directional ICGM. When compared with the single-directional ICGM, the double-directional ICGM can extract robust features from the dynamic environment more effectively.

The main difference between the two methods is that the single-directional ICGM extracts features only on the basis of the previous state of the environment, which means that only if a feature was previously present for a sufficiently long duration, then that feature is considered to be robust. Features extracted by the single-directional ICGM are definitely robust. In future, features that are observed for a sufficiently long duration can be considered as a robust feature. Therefore, the "future" data can be used. By using both sets of information, the double-directional ICGM can solve many problems due to the single-directional ICGM. Referring to Fig. 4.4 and Fig. 4.5. When $\lambda_\alpha < 0.5$, the double-directional ICGM (with appropriate Thr) dose not lose any feature. $\lambda_\alpha < 0.5$ means that the camera should not rotate too fast. To extract features as many as possible, when the camera rotates, the overlap's acreage of two neighboring images should be larger than each image's acreage's 50%. The method to decide Thr is discussed in section 3.1 in detail.

For several conditions, we described the factors that will affect the extraction of (single-directional and double-directional) ICGM. The parameters λ_α and λ_β are very important in the extraction of ICGM because unless we increase these parameters, we cannot extract more robust features. However, when λ_α and λ_β are increased, the number of extracted features decreases. To extract a sufficient number of features while increasing λ_α and λ_β, the double-directional ICGM is very effective. Although we only mentioned how the factors affect the extraction of ICGM in a few simple conditions, the movements of an average camera can be separated into an approximate combination of these simple situations. Therefore, the double-directional ICGM is usually better than the single-directional ICGM.

4.1.5 Relationship with PIRF [1]

PIRF [1] is also a method that is used to extract robust features from a sequence of images. With reference to Algorithm 1, when $Thr \to \infty$, the Single-Directional ICGM $\to$ PIRF [1](window size = 2). However, if the window size = 2, PIRF [1] is not sufficiently robust. If we want to extract additional robust features using PIRF [1], we should enlarge the window size. Nevertheless, while enlarging the window size, PIRF [1] will provide fewer visual features, which will be a problem.

When Thr is set as normalsize, the Single-Directional ICGM is more robust than PIRF [1](window size = 2). Furthermore, the double-directional ICGM can increase the number of extracted features. These ICGM characteristics can be beneficial during procedures such as SLAM.

4.2 Online-Incremental-Appearance-only SLAM

We used a method similar to that cited in [8] to achieve online-incremental SLAM. PIRF [1] is used as a feature for the study in [8]. The proposed method uses features that are extracted using ICGM.

An online-incremental SLAM method that is based on the detection of loop closures is cited in [8], i.e., the dictionary cited in [8] is dynamic and it will automatically register a new word. The online-incremental method requires more calculation than that by the batch method.

The study cited in [8] can be divided into two phases: the creation of the dictionary and its testing. In addition, ICGM is suitable for online-incremental SLAM. Moreover, ICGM can not only be used to extract features but also to detect loop-closure. This characteristic of ICGM also improves the precision of SLAM.

Although the double-directional ICGM can be used while creating the dictionary and its testing, to calculate the robust features of I_t, the double-directional ICGM requires I_{t+1}. Therefore, the robot system should obtain the information of "future" events, which is not possible. Hence, for real-time systems such as automatic robots,, the testing phase cannot employ the double-directional ICGM to increase the robust features, and we can instead use the single-directional ICGM. However, real-time systems can also use the double-directional ICGM to create a dictionary. Platforms such as pedestrian navigation do not require real-time processing; therefore, they can use the double-directional ICGM while creating the dictionary and testing it to improve the effectiveness of the system.

Fig. 4.6 shows the System framework of ICGM based online-Incremental-appearance-only SLAM. Incremental dictionary is described in Fig. 2.4.

4.3 Summary

This chapter described the proposed's algorithms for crowed pubulic environment, including feature extraction algorithm(single and double directional ICGM) and ICGM based SLAM algorithm.

Next chapter presents results of the proposed method in crowed pubulic environment.

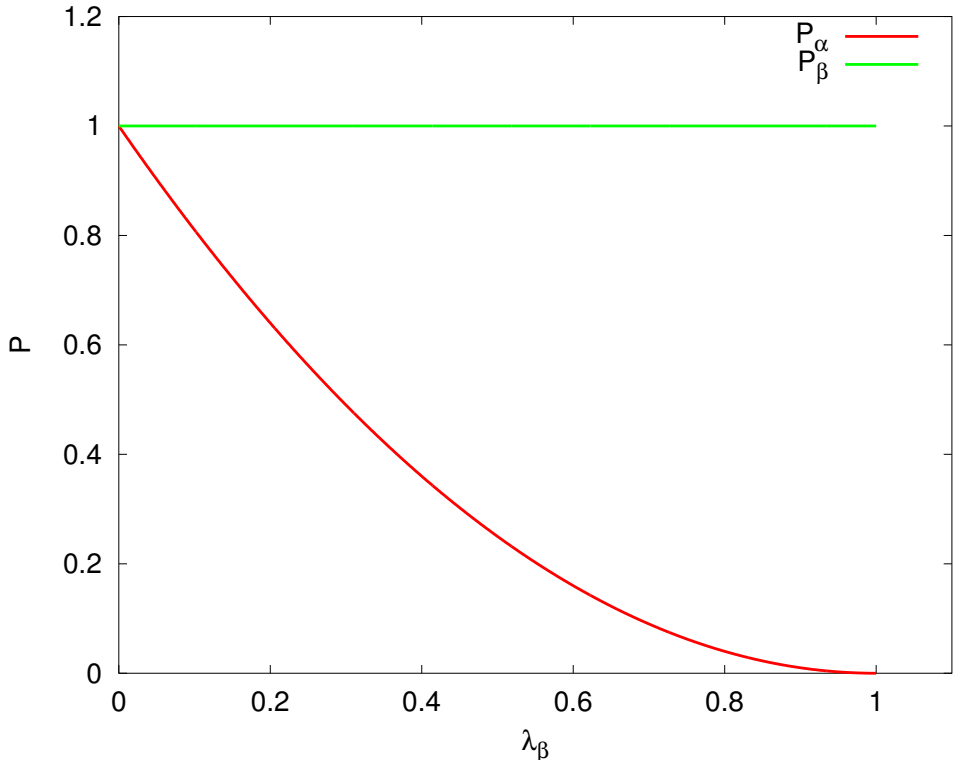


FIGURE 4.5: Comparison between Double and Single Directional-Approaches when the camera approaches or moving away from an infinite object at a certain speed: similar to Fig. 4.4, the double-directional ICGM never loses any feature. Meanwhile, single-directional ICGM loses many features.

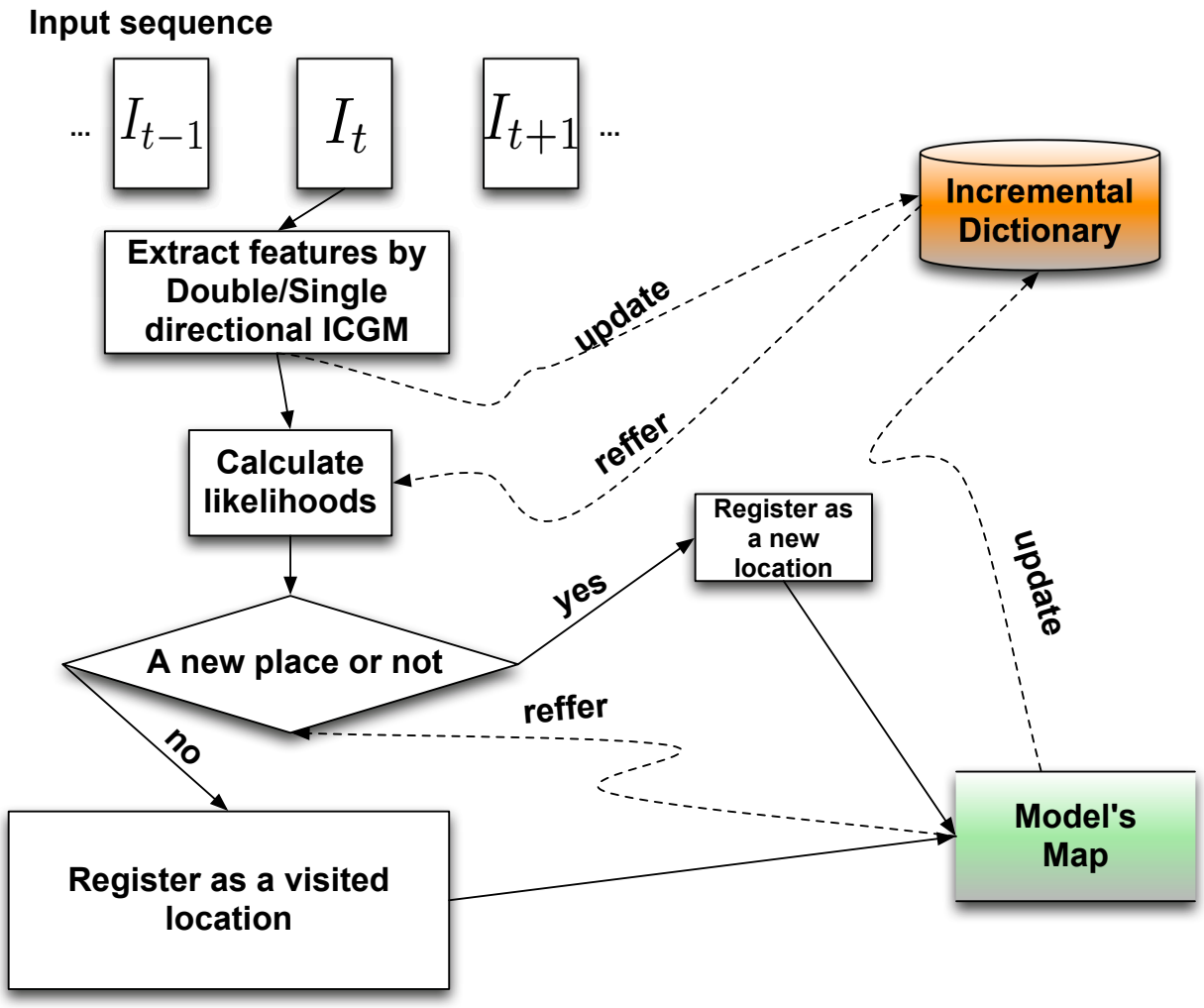


FIGURE 4.6: System framework of ICGM based online-Incremental-appearance-only SLAM.

Chapter 5

Experiments

In this chapter, we elucidate the results of a location-recognition experiment in a crowed environment. In addition, we will show the results for a SLAM experiment in a dynamic outdoor environment.

5.1 Experiment A : Visual-Feature Matching By ICGM

ICGM is a visual-feature matching method basically. This experiment's dataset is captured at an indoor environment. In the environment their are some dynamic objects.

Fig.5.2(b) shows the situation of conventional SURF matching. In this image there are many incorrect matchings. Besides, features on moved objects are not be ignored. Fig.5.2(c) is the situation of ICGM, there are not any incorrect matching and dynamic objects are ignored correctly. Conventional SURF matches' number is 975 and ICGM's number is 374. We observed that when the number of ICGM's features is 374, there is no incorrect matching. Fig.5.1 shows ICGM's number of features' trend while the Thr changes. When $Thr = 0.08(1/Thr = 12.5)$ the ICGM extracted 374 matches without false-positive and false-negative. While Thr is getting normalizer, false-negative matches are increasing. When $Thr = 0.031(1/Thr = 31.4)$, because the Thr is too strict, the ICGM can not extract any match.

So far, the Thr needs to be decided by user. It means that the user needs to change the Thr until the ICGM works without false-positive and false-negative match. However, an appropriate Thr is suitable for images, which were captured with similar intervals. So for one dataset, the user only needs to decide the threshold once.

This experiment shows that ICGM can work stably and effectively in dynamic environment even camera moved a lot.

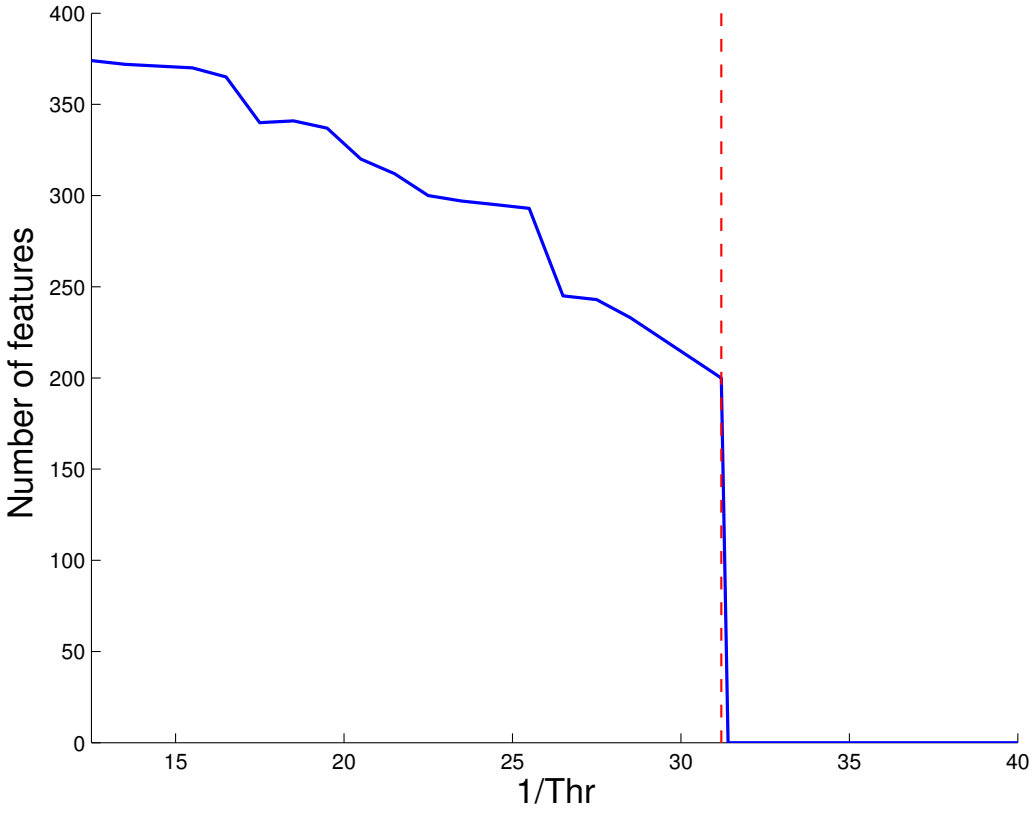
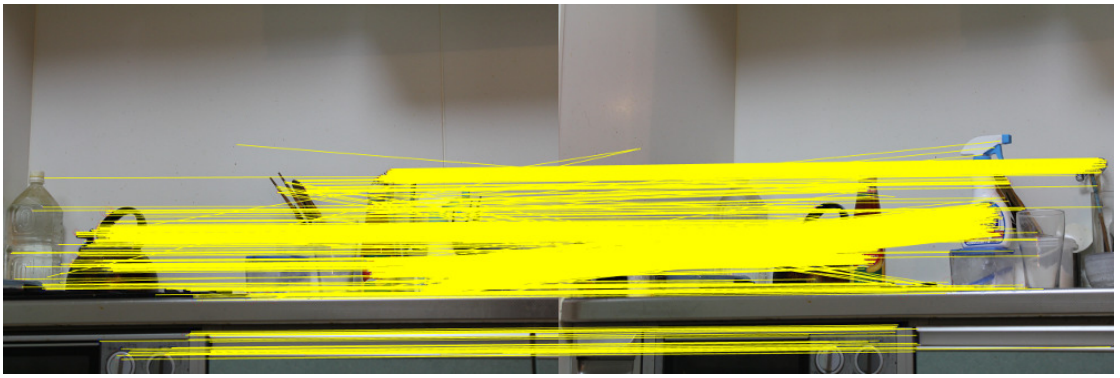
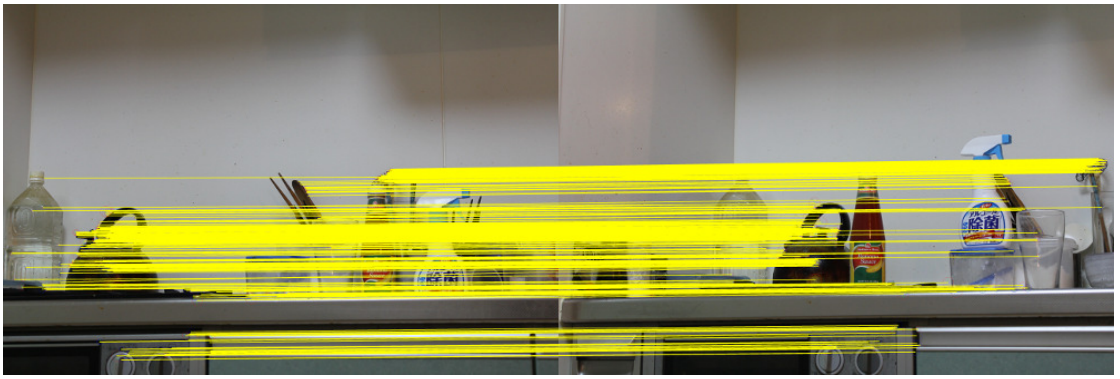


FIGURE 5.1: ICGM's number of features' trend while the *Thr* changes on the basis of Experiment A

(a) Experiment A's dataset. These two images are captured at different time. Objects' positions in red ellipses have been changed. Besides camera's view-angle has changed.



(b) Conventional SURF matching between two images. There are many incorrect matchings. Moreover, many feature points on dynamic objects are matched.



(c) Incremental center of gravity matching between two images. All matchings are correct, dynamic objects are ignored correctly.

FIGURE 5.2: Images of experiment A

5.2 Experiment B : U-shaped Route at Shibuya Station (Location Recognition)

This experiment's dataset is captured using a hand-held camera (made by sony) (resized resolution: 480*320) at Shibuya Station, the frequency is 0.5 frame per second. The station was crowded. The learned route was approximately 80 m, and the learning duration was 5 min. The test datasets were similar. First, we extracted the features (using ICGM and PIRF) from the learning database. Then, we extracted the features from the testing database. Finally, we performed the experiment for the learning and testing features. This experiment was not a SLAM experiment, but it tests ICGM's SLAM system in terms of the place recognition performance.

Fig. 5.3 shows a comparison between double- and single-directional ICGMs, and this experiment shows that double-directional ICGM can obtain more robust features than that obtained with single-directional ICGM.

Fig. 5.4 are images of experiment B.

In this experiment, the precision of PIRF [1] was 82.65%, whereas, the precision of double-directional ICGM is 98.56% and $Thr = 0.075$.

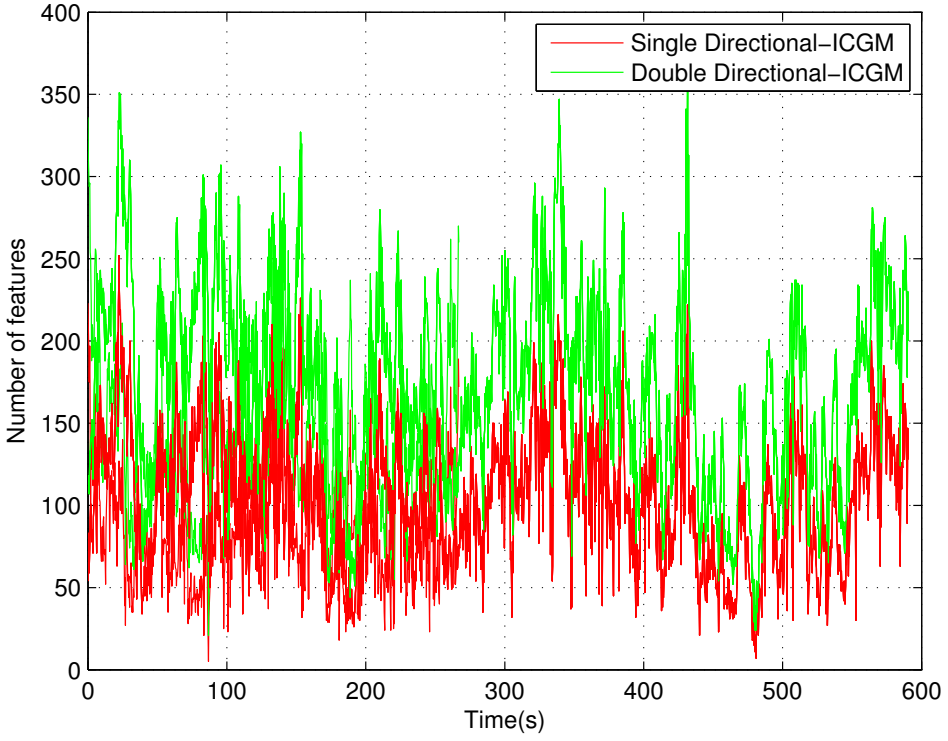
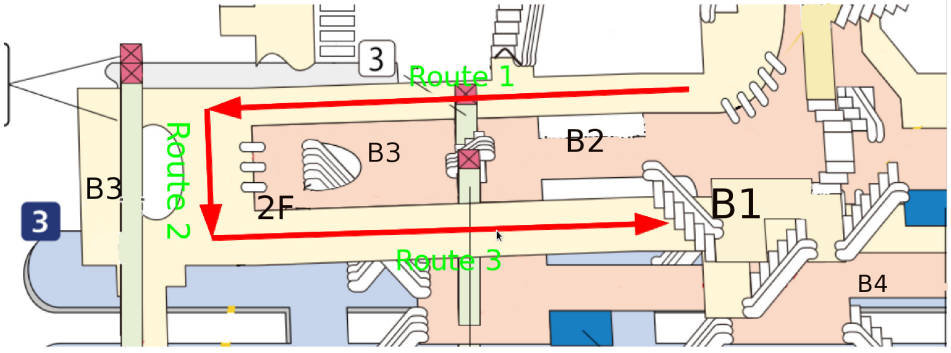


FIGURE 5.3: Features' amount's comparison between Double and Single-Directional ICGMs on the basis of Experiment B



(a) Experiment B's route, the route is composed by route 1, 2 and 3. Route 1 and 3 are abrupt slopes, route 1 and 3's exteriors are similar. It is easy to confuse 1 with 3 using vision information.

(b) Image of crowed Shibuya Station at route 1. Pedestrians in red ellipse are dynamic components of the environment.

(c) Visual-feature extraction's result on the basis of Experiment B. normalize circles with different colors represent extracted visual-features. Dynamic components of this environment are ignored correctly.

FIGURE 5.4: Images of experiment B

5.3 Experiment C : Minamidai Outdoor (SLAM)

This dataset for this experiment was also captured using a hand-held camera (resized resolution: 480*320), the frequency is 0.5 frame per second. This scenario was not as crowded. However, there were a few dynamic objects (cars, humans). The learned route was about 170 m, and the learning duration of 9.5 min.

Fig. 5.5 is images of experiment B. Fig. 5.6 shows the result of double-directional ICGM.

Table 5.1 shows the results of this experiment. The proposed methods' accuracy and ratio of recall is better than those of PIRF-nav2.0[8]. Because the *proposed method (single-directional)* used the single-directional ICGM in the testing phase and the *proposed method (double-directional)* used the double-directional ICGM in the testing phase, the *proposed method (single-directional)* extracts fewer features than those by *proposed method (double-directional)*. Thus, the *proposed method (double-directional)* is better than the *proposed method (single-directional)*. Double-directional ICGM as well as single-directional ICGM obtained best results while $Thr = 0.082$. In addition, because FAB-MAP is a batch method that generates dictionary in offline, it is the fastest of all the methods. However, FAB-MAP is not a fully online incremental method.

(a) Appearance of outdoor in Minamida. Human and car in red ellipses are dynamic components of the environment.

(b) Visual-feature extraction's result on the basis of Experiment C. Dynamic components of this environment are ignored correctly.

FIGURE 5.5: Images of experiment C

TABLE 5.1: Results of experiment C

	Recall	Precision	Total Processing Time (second)
Proposed method (double-directional ICGM)	96.1%	97.5%	204.25
Proposed method (single-directional ICGM)	86.1%	95.2%	196.51
FAB-MAP [6]	38.8%	95.4%	155.6
PIRF-nav2.0 [8]	86.2%	78.8%	182.4

FIGURE 5.6: Results of experiment C. Red points are loop-closing detected locations. And the yellow lines represent the experiment's route.

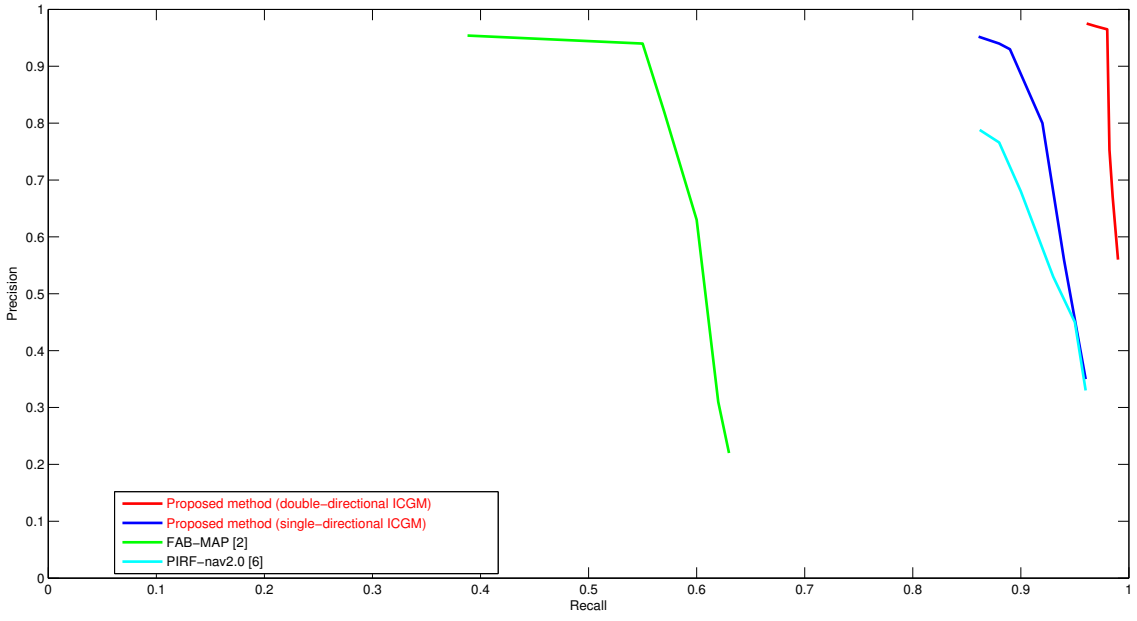


FIGURE 5.7: Precision and recall-ratio of experiment C.

5.4 Summary

Experiments have shown that our proposed method not only can learn objects in complex dynamic environment but also can achieve in a dynamic public environment. All images were photographed using a conventional hand-held single-lens camera. The proposed method was shown to have high accuracy and robustness.

Experiment A shows that using the proposed method, motion segmentation was accurate.

On the basis of Experiment C, Table 5.2 shows a brief review of SLAM methods. The proposed methods can run without offline dictionary generation. The proposed methods are online incremental methods. Although the proposed method (double-directional) causes a delay, its accuracy is highest. On occasions of high real-time demand (SLAM for robot etc.), we should choose the proposed method (single-directional). However, on other occasions, for example, pedestrian navigation, the proposed method (double-directional) works better.

Table 5.2: Review of SLAM methods

	Ability to run incremental without offline dictionary generation	Accuracy	Ability to process without delay
Proposed method double-directional ICGM	Yes	Very high	No
Proposed method single-directional ICGM	Yes	High	Yes
FAB-MAP [6]	No	Low	Yes
PIRF-nav2.0 [8]	Yes	Moderate	Yes

Today, high-performance hand-held smart phones have become very popular. For the proposed method to possess high robustness while using hand-held devices, ICGM may be applied to many types of platforms (including hand-held smart phones) for navigation by pedestrians. This is our future goal.

Deciding the threshold Thr automatically is one of our important future study.

When compared to PIRF-Nav 2.0, the processing speed of the proposed method is possibly relatively fast. However, we are currently considering replacing SURF with a type of corner detector (FAST[35], HARRIS[36] etc.). Although SURF[4], SIFT[3] possesses more information for visual recognition, FAST[35] or HARRIS[36] is faster. Furthermore, ICGM can extract robust corner detectors on the basis of their geometric structure. Therefore, we intend to use FAST[35], HARRIS[36] for faster processing.

Chapter 6

Applications of ICGM

ICGM not only can apply to vision only SLAM but also can apply to hybrid SLAM and visual odometry. In this chapter, ICGM based hybrid SLAM and visual odometry are introduced.

6.1 Hybrid SLAM Based on RGBD Camera

Kinect is a sensor devoloped by microsoft which can grab depth's information and RGB-vision information from environment at the same time. For its high performance, high accuracy, low power consumption, it is very suitable for use in mobile robots.

In this study, we propose a SLAM/Navigation method to combine depth's information and RGB-vision information grabbed by kinect with mobile robot's odometry information.

Similar to 3D-PIRF[19] we modify trajectories based on loop closure.

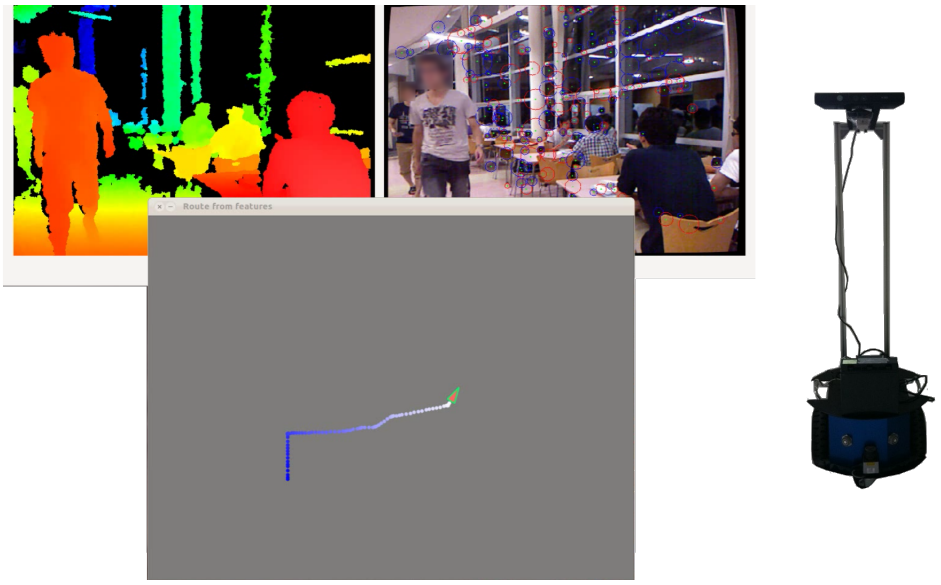


FIGURE 6.1: Mobile robot's hardware system, and its SLAM in dynamic environment. Right image is robot with Kinect installed. Left image is a sample of SLAM based on Kinect grabbed information from crowded environment. Left top of this image is depth information transformed to RGB, right top of this image is common RGB vision information. Features extracted by ICGM are marked as circles in common vision RGB information. The blue line in the image below depth's information and common RGB vision information is robot moved route calculated by SLAM.

6.1.1 Incremental Hybrid Map Construction and Modification Based on Loop-closure Detecting (SLAM)

We propose a SLAM method can be used in highly dynamic environment. And we create hybrid map in SLAM, this hybrid map including route's topological information, vision features and their corresponding coordinates. And the Hybrid map can be used for autonomous navigation directly. Hybrid map's contraction is indicated in Fig.6.2.

[p]

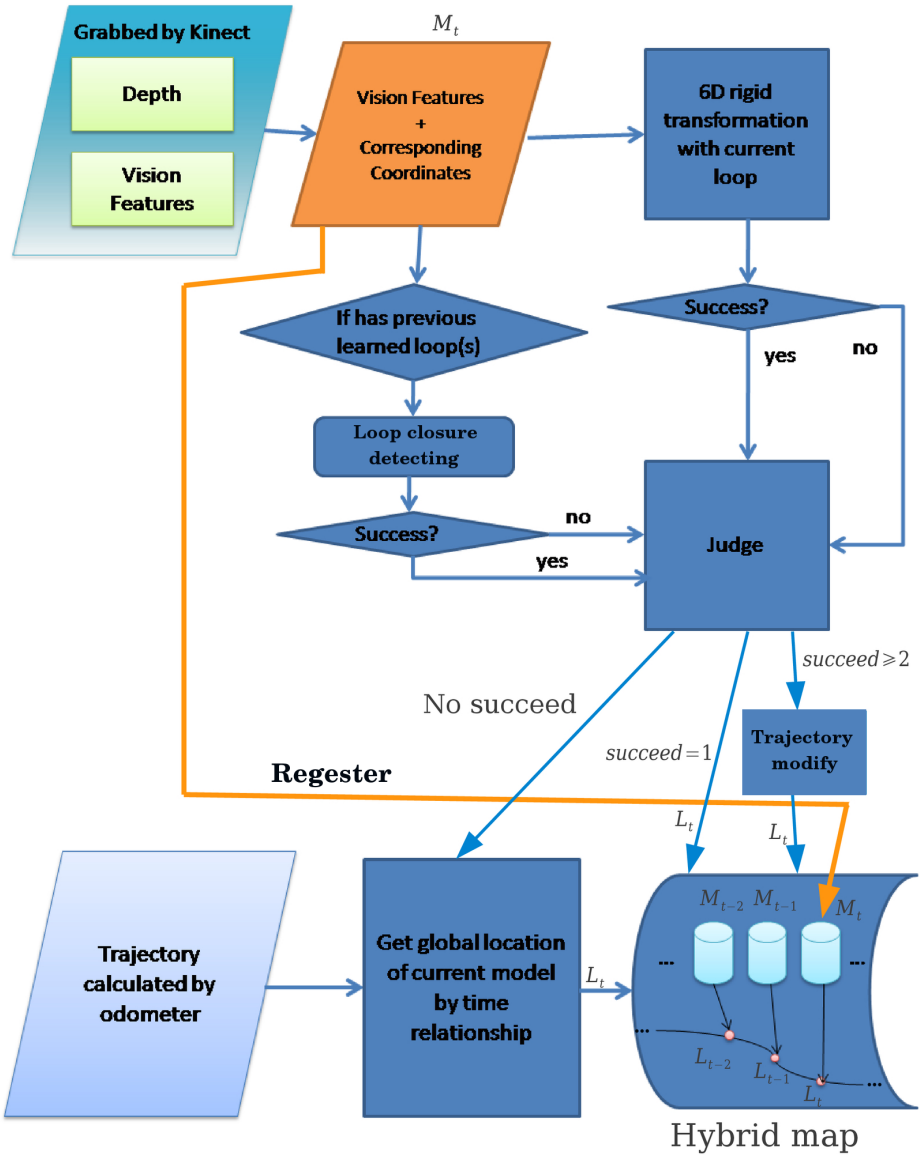


FIGURE 6.2: Hybrid map's contraction. This figure describe SLAM's system's structure. We have two metric sources of robot's movement, one is depth information and vision information grabbed by kinect, the other is trajectory calculated by odometry. Assume in time t, we get one model depth information and one vision information. After process(including ICGM extraction and Kinect's calibration etc), we obtain a set of robust vision features and these features' corresponding 3D coordinates, called a metric model. M_t is metric model of time t. We try to get M_t's location by test with current loop and previous learned loop(s). The location pose of M_t is L_t. If we can not obtain L_t from current loop and previous learned loop(s), we get L_t from trajectory calculated by odometry. Then, M_t and L_t are registered into the hybrid map.

TABLE 6.1: Comparison of related navigation methods in dynamic environment

	Sensor	Learning envirommment		Human in Navigatin	Precision	Processing Speed(per frame)	Loop closure detected ratio
		static	dynamic				
Hatao et al. [37]	2D LRF, odometry	-	◯(several)	2	◯	200ms	-
Muller et al. [38]	2D LRF, odometry	◯	-	6	◯	250ms	-
Koch et al. [39]	Omnivision Cameras	◯	-	8	△	667ms	△
Morioka et al. [19]	Omnivision, odometry	-	◎(over 20)	over 20	◯	3870ms	◯
Proposed method	3D LRF, Vision, odometry	-	◎(over 20)	over 20	◎	202ms	◎

In this section we would like to introduce the hybrid map's structure and modification based on loop-closure detecting.

6.1.1.1 Basic Structure of Hybrid Map

Proposed hybrid map construct by visual odometry based on 6D rigid transformation[40] between 3D points clouds. We control robot to move in a place, robot can grab depth, RGB-vision and odometry information at the same time. A 3D points cloud can reconstruct from per model's Depth and RGB-vision data, each 3D points clouds is called metric model M_x. We can get a queue of M_x continuously, namely $\mathbb{M}$. By computing 6D rigid transformation between metric models, we can get relative 3D pose R. For instance, R_x can be calculated from two metric models M_x and $M_{x-\phi}$. ϕ is calculation interval of 6D rigid transformation. From set of relative 3D poses $\mathbb{R}$, we can get set of robot's global location poses $\mathbb{L}$, the recursion formula is:

$$L_n = L_{n-1} \oplus R_n \tag{6.1}$$

It is noteworthy that the $\oplus$ and $\ominus$ operations of 3D poses have special definitions. $\oplus$ and $\ominus$ operations mean 3D pose transformation. Based on the set of L we get a global route which the robot moved. Meanwhile, depth and RGB-vision data will be register into corresponding location of the route.

If a hybrid map have a correct topological structure, in autonomous navigation route planing and shortest path calculation will be very convenient, so we try to keep correct topological structure of hybrid map.

But for some reason such as lacking enough effective 3D feature points etc., it is not guaranteed that 6D rigid transformation can calculate 3D pose successfully all the time. So it is very hard to keep correct topological structure of hybrid map only by depth and RGB-vision information. However, we use tires' odometry information to maintain the correct topological structure.

As details, we try to get a inliers's set of 6D rigid transformation by RANSAC[41]. If we can not get a good inliers's set, we treat this calculation as failed. When failed, based on time relationship between tires' odometry information and depth and RGB-vision information, we update relative pose which calculated from tires' odometry to R.

6.1.1.2 Hybrid Map's Modification and Incremental Construction Based on Loop-cloure Detecting

To prevent error accumulation, we propose a hybrid map's modification method based on loop-cloure detecting. Loop-cloure detecting is described in Fig.6.3.

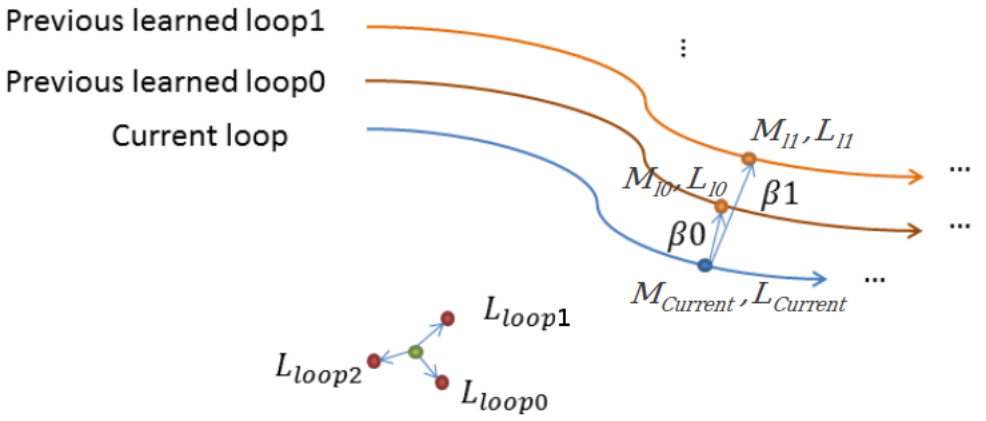


FIGURE 6.3: Loop-cloure detecting. Previous learned loops are loops in learned. Hybrid map. The loops only represent logic relationship not actual topological relationship. Although only two previous learned loops in this figure, suppose that current model $M_{Current}$ detected loop closure with previous learned loops 0, 1, 2. We get relative poses β_i between $M_{Current}$ and M_{li}. L_{loopi} is global location pose of $M_{Current}$ calculated by $loop_i$, $L_{loopi} = L_{li} \oplus \beta_i$. Each L_{loopi} are always not the same. The red points below represent global location pose of $M_{Current}$ calculated by diffrent previous learned loops. We do optimization based on distance's relationship to this location pose's set. The green point represents global location pose of $M_{Current}$ after optimization.

Proposed loop-cloure detecting method is based on ICGM matching vote and 6D rigid transformation. At first we calculate ICGM macthing score between current model and

each previous loop's models. We use a method which similar to PIRF-nav2[8] to calculate the ICGM macthing score. And we get 3 best scored models $model_{best}, model_{second\_best}, model_{third\_best}$ for each loop.

Then we try 6D rigid transformation between current model and 3 best scored models. Suppose 6D rigid transformations are succeed, we can get a relative pose($R_{LC1}, R_{LC2}, R_{LC3}$) between current model and the best macthed 3 models. Assume the 3 6D rigid transformations are all failed, we would treat this loop-closure as failed.

By comparing the 3 output RANSAC[41] modeling results of 6D rigid transformations, We choose the best relative pose β_i. Thus, we can get a current model's global location $L_{loopi} = L_{li} \oplus \beta_i$. Also, we can get another current model's global location $L_{Current}$ from its own loop.

We can get a set of L_{loopi}. It's need to do a optimization to this pose's set for getting a best global location pose $L_{LoopClosure}$ of $M_{Current}$. We minimize the following expression:

$$\sum_{i=0}^{N} \| L_{li} \oplus \beta_i - L_{LoopClosure} \|^2 \tag{6.2}$$

This expression is calculating Mahalanobis distance of 3D pose and N is the sum of loop closure detected loops.

Ideally, $L_{Current}$ and $L_{LoopClosure}$ should be equal. But actually $L_{Current}$ and $L_{LoopClosure}$ can not reach consensus. And:

$$\triangle\ Pose_{Diff} = L_{LoopClosure} \ominus L_{Current} \tag{6.3}$$

Because location calculated from its own loop will have error accumulation, we always suppose $L_{LoopClosure}$ is truth. Our task is let $L_{OwnLoop}$ and $L_{LoopClosure}$ to reach a semi-consensus. The reason why only choose semi-consensus is preventing over-correction. We apportion the difference to a part of relative poses' set $\mathbb{R}$.

$$[R'_a \cdots R'_b] = [R_a \cdots R_b] \oplus \frac{\alpha}{b-a} [\triangle\ Pose_{Diff} \cdots \triangle\ Pose_{Diff}] \tag{6.4}$$

Here, we set $0 < \alpha < 1$, so the modification can not reach the full-consensus but a semi-consensus. a is number of last location modified model in hybrid map, b is number of current model, $[R_a \cdots R_b]$ is vector of relative poses that should be modified, $[\triangle\ Pose_{Diff} \cdots \triangle\ Pose_{Diff}]$ is vector of difference, $[R'_a \cdots R'_b]$ is modified relative poses

vector. After relative poses' modification, we rebuild location poses $[L_a \cdots L_b]$ by expression (8).

There are two kinds of learning called batch learning and online learning. Although batch learning is faster than online learning, hybrid map created by batch learning can not be expanded, it is not so suitable for actual usage. We propose a incremental hybrid map building method based on loop-closure detecting. ICGM can support incremental hybrid map building. Basic approach of incremental hybrid map building is that while recognize location from existed map, update new information to existed map. We use loop-closure detecting described above to recognize location of current model from existed map. After recognized its location and modified route, a new model will be register into this map. Registered information including vision features, their corresponding 3D coordinates and the model's global location. When loop-closure detecting failed in some models, we will set model's global location equal to relative pose calculated by its own loop directly.

In addition, using proposed method learned multiple data sets can be mergered to be one hybrid map. For instance, human can control robot to learn one environment in different time with different light conditions, then get multiple data sets. Based on these data sets we can merger them to be one hybrid map. By using this hybrid map, robot system will suitable for different light conditions.

6.1.2 Navigation Based on Learned Hybrid Map

Autonomous navigation is also based on loop-closure detecting. While navigate, robot system also create a hybrid map to localize itself. The hybrid map's creation is almost the same as SLAM. But in (19) we always set $\alpha = 1$, intend to reach full-consensus in modification. Robot will reach the goal more effectively since a full-consensus reached in map construction.

Learned hybrid map have a set of features and corresponding 3D coordinates' models with their own indexes. Indexes are from 1 to N. N is the longest loop's sum in this hybrid map. We use a pointer p to present the current location of robot in hybrid map. In initial time, robot in the starting point and $p = 1$.

Robot try to detect loop closure in hybrid map using currently grabbed model $M_{Current}$. Suppose robot can detect loop closure at index i of longest loop in hybrid map, we can get a best relative pose $RelPose_{current}$ between $M_{Current}$ and M_i. If loop closure detecting failed, we set $RelPose_{current} = 0$, meanwhile set i equal to index of ICGM best matched model of longest loop in hybrid map. And we set a parameter called ω, ω is logical moving step of robot in hybrid map.

$$Pos_{Move} = L_{i+\omega} \ominus L_i \oplus RelPose_{current} \tag{6.5}$$

Then robot will change its pose by Pos_{Move} automatically. Although robot can not reach the goal fully accurately, it is means that robot has reached the $i + \omega$ model in map analogously. After moving, we set current pointer $p = i + \omega$. And robot will do process as the same as above repeatedly, until p reach N.

Although navigation method described above is not so complicate, because the hybrid map keeps correct topological structure, Dijkstra [42]'s algorithm can be applied to path planning for searching the shortest path.

6.1.3 Experiments

In experiments, we used a iWs09 robot with a Kinect and a odometry installed. Kinect is located at a height of 1300mm. We use C++ to created the algorithm. And we run the algorithm on a Intel corei7 notebook computer. Operating system is ubuntu linux 11.10. The computer with a GTS360m installed for acceleration.

6.1.3.1 Robust Vision Feature Extraction in Highly Dynamic Environment

In this experiment, we show the result of ICGM extraction using proposed double-directional ICGM.

We chooose window size of double-directional ICGM $\mathbb{S} = 3$. In this experiment we take sequential images in cafeteria of tokyo institude of technology. And as we can seen in fig.6.4. Pedestrians who were walking will not get any ICGMs. It means that double-directional ICGM is robust enough and suitable for dynamic environment.

We use the same database grabbed in cafeteria of tokyo institute of technology as SLAM's experiment below. The database including 6739 frames of sequential RGB image. We calculate single-directional ICGM and double-directional ICGM based on this database's squential RGB images. Number of double-directional ICGM is significantly more than number of double-directional ICGM. As numerical results, average number of single-directional ICGM is 160.562, meanwhile average number of double-directional ICGM is 260.863. We get 62.3% increased.

FIGURE 6.4: Appearance of double-directional ICGM's extraction. Pedestrians within red circles were walking. Small blue circles are feature points which centers are darker than backgrounds. Small red circles are opposite. Red feature points' centers are brighter than backgrounds. Each circle's radius is refers to detected scale of each feature point.

6.1.3.2 SLAM's Experiment in Small Scale Dynamic Environment

In this experiment, we have controlled robot to train 2m's route of straight line in our research room (indoor environment) without physical odometer's information (only by Kinect's visual odometer). The learned route is showed in Fig. 6.5. While learning this route, one person walked in front of robot for obstructing as Fig. 6.6. The result shows in table 6.2.

TABLE 6.2: Result of experiment in small scale environment.

Task	Error ratio
Route learning	2.5%

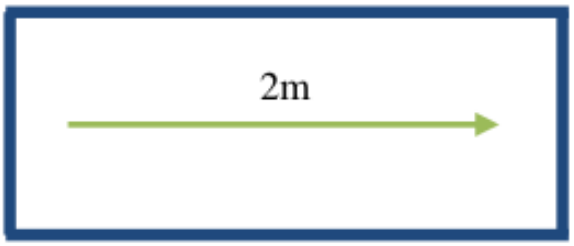


FIGURE 6.5: Training route of experiment in small scale environment.

FIGURE 6.6: A people was walking in front of robot.

6.1.3.3 SLAM's Experiment in Large Scale Highly Dynamic Environment

We do this experiment in cafeteria of Tokyo institute of technology. It is 20m * 20m sized. We launch our method in this cafeteria at night 8:00. At that time, there were 70 people in cafeteria approximately. They eat or walk in this environment as their own wills. It was a highly dynamic environment. For notarizing effectiveness of proposed SLAM's method in highly dynamic environment, we do experiment in this environment.

In this environment, we control robot by a joystick. While moving, robot record odometry information by odometry and record vision, depth information by Kinect. We control robot to move two loops clockwise as route of Fig. 6.7. In the end of learning, the robot return staring point. And total distance moved is about 80m. In the process, robot get 6739 frames vision and depth information.

Fig. 6.8 shows trajectory calculated only by odometry. Ideally, calculated starting point and terminal point should be the same. However, trajectory calculated by odometry shows a big error. And it can not keep correct topological struture. Meanwhile, Fig. 6.9 shows heterogeneous trajectory calculated by odometry and Kinect with loop closure detecting. Althout it is using information grabbed by odometry, 97.6% of this trajectory

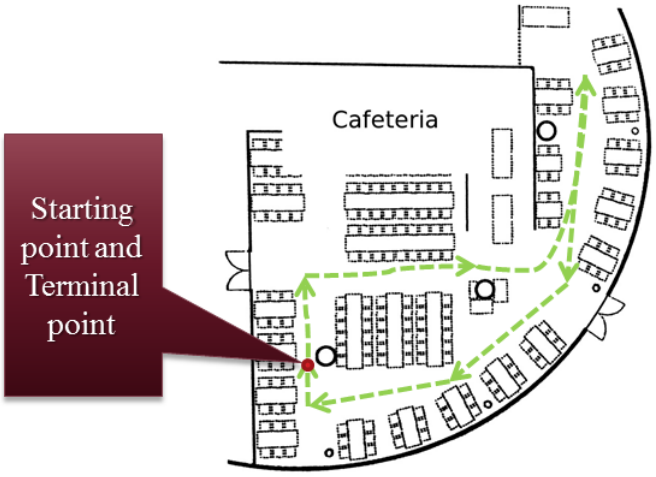


FIGURE 6.7: Learned route in cafeteria.

is calculated by recorded Kinect's vision, depth information.It has correct topological struture. Calculated starting point and terminal point are almost the same. As numerical evaluation, distance between calculated starting point and terminal point is 150mm in average and error of robot's angle is within 10.2 degrees.

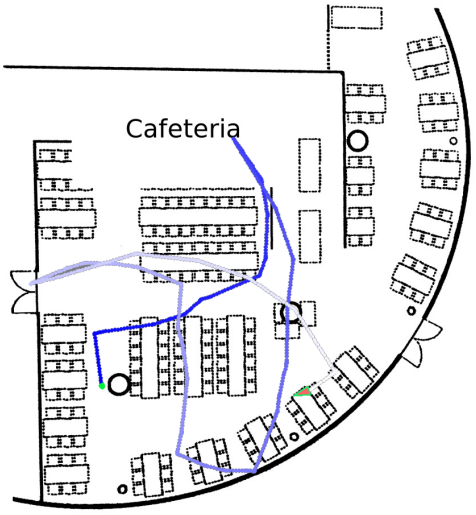


FIGURE 6.8: Trajectory calculated only by odometry. The green point is starting point, and the triangle is terminal point.

Loop closure detecting rate is 63.8%. Also, table 6.3 shows comparative of loop closure detecting rate between method proposed by Morioka, proposed SLAM system using single-directional ICGM method and proposed SLAM system using double-directional ICGM method. The result shows that double-directional ICGM method is effective for visual odometry be used in highly dynamic environment.

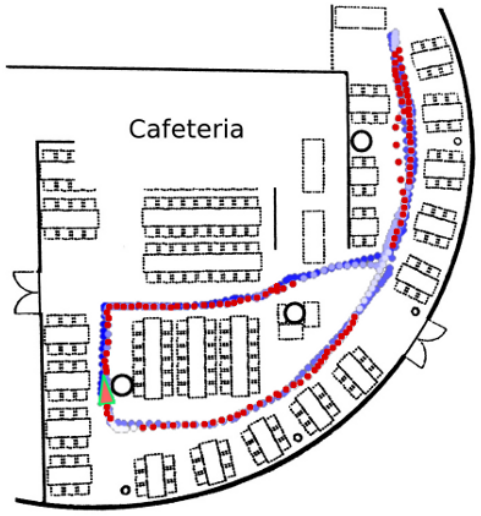


FIGURE 6.9: Heterogeneous trajectory calculated by odometry and Kinect with loop closure detecting. The triangle represents staring point and terminal point. They are almost at the same location. Red Points represent locations where loop closure detected.

TABLE 6.3: Loop closure detected rate

	Detected Rate
Morioka	44.2%
Proposed method(single-directional ICGM)	47.2%
Proposed method(double-directional ICGM)	63.8%

6.1.3.4 Autonomous Navigation's Experiment in Highly Dynamic Environment

We do a navigation's experiment in the same cafeteria based on map learned at SLAM's experiment above. And experiment's environment is almost the same as SLAM's experiment above. The route shows in fig. 6.10. Fig. 6.11 shows that in navigation, the environment was very crowded. But proposed system can work in the environment stably.

The average calculating time for 1 frame is 321 milliseconds. Robot's moving speed was 125mm/s. And average deviation from planned route was 60mm, maximum deviation from planned route was 92mm. table 6.4 show navigation result's comparison between method proposed by Morioka and proposed method. We did not set the moving speed faster for safety factor.

FIGURE 6.10: Appearance of navigation in cafeteria.

TABLE 6.4: Navigation comparison

	Processing speed(ms)	Moving speed(mm/s)	Average error(mm)
Morioka	3870	38.4	150
Proposed method	321	125	60

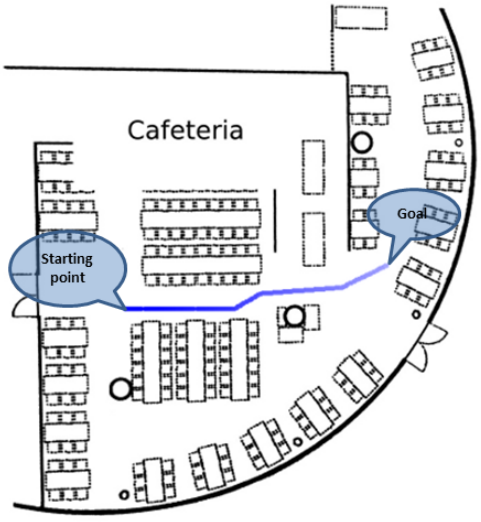


FIGURE 6.11: Navigation's experiment. The length of navigation route is about 12m.

6.2 Visual Odometer Based on Mono Handy Camera

Kayanuma in our research group proposed a ICGM based mono visual odometry.

Visual odometry's objective is calculate the camera's trajectory by mono image sequence. It is very useful for robots navigation etc.

Visual odometry calculates two images' 6D spatial transformation based on epistolary equations. To establish epipolar equations correctly, features matching between two images is necessary. So feature's robustness and matching's correctness is critical.

crowded environment(station,downtown etc.) is a big challenge of visual odometry. Previous visual odometry[43][44] can not work well in public crowded environment.

Kayanuma uses ICGM to extract and match features for visual odometry.

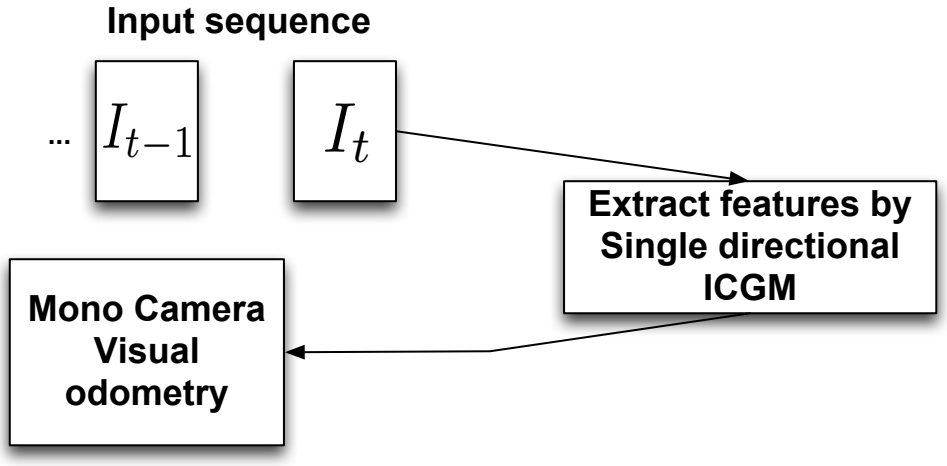


FIGURE 6.12: ICGM based visual odometry.

6.2.1 Experiment: Shibuya Station Indoor

This dataset is grabbed by handy camera in shibuya station indoor. The environment was very crowded. (Fig. 6.13)

The route is 100m * 50m approximately and the length of the route is about 200m.(Fig. 6.14)

Although this is no ground-truth, the result in Fig. 6.15 shows that visual odometry with ICGM is better than visual odometry without ICGM.

FIGURE 6.13: Very crowded shibuya station indoor.

6.2.2 Experiment: Shibuya Station Outdoor

This dataset is grabbed by handy camera in shibuya station outdoor. The environment was very crowded too. (Fig. 6.16)

The route is 10m * 20m approximately and the length of the route is about 50m.(Fig. 6.17)

We use google earth's GPS data as groundtruth, the result in Fig. 6.18 shows that visual odometry with ICGM is better than visual odometry without ICGM.

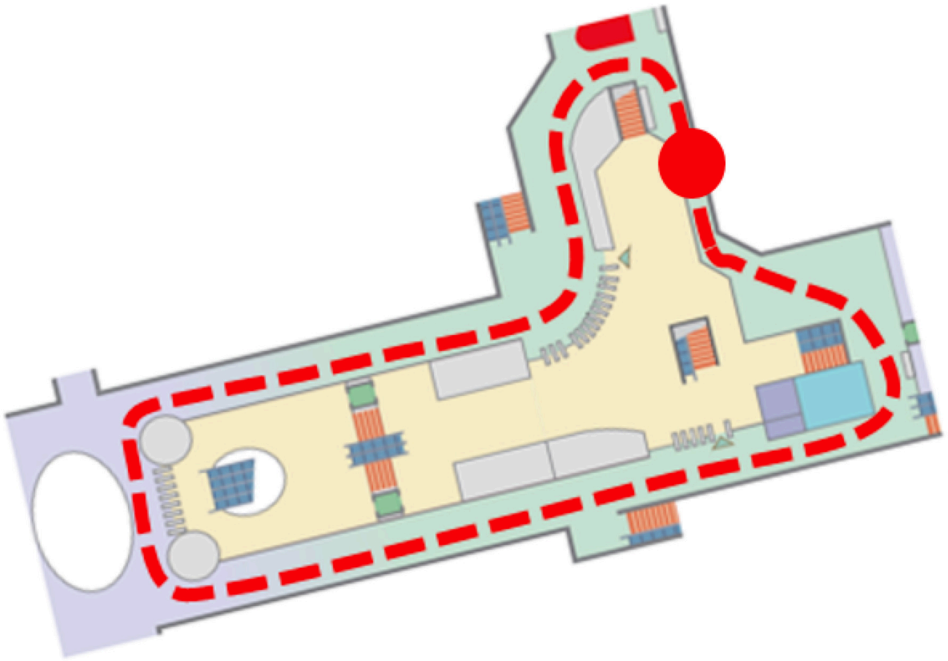


FIGURE 6.14: Shibuya station indoor's Learned route.

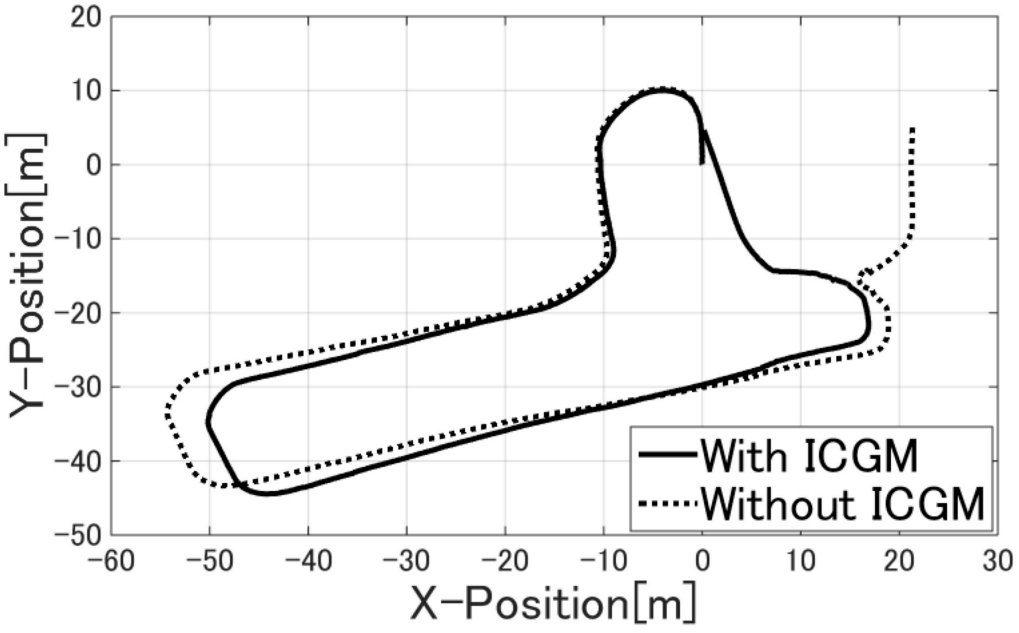


FIGURE 6.15: Shibuya station indoor's comparison between ICGM based and Non-ICGM visual odometry.

TABLE 6.5: Result comparison based on shibuya station outdoor dataset

	error[%]	inlier[%]	time [s per frame]
with ICGM	6.01	71.1	0.32
without ICGM	8.09	70.0	0.31

FIGURE 6.16: Very crowded shibuya station outdoor.

FIGURE 6.17: Learned route of shibuya station outdoor.(GOOGLE EARTH)

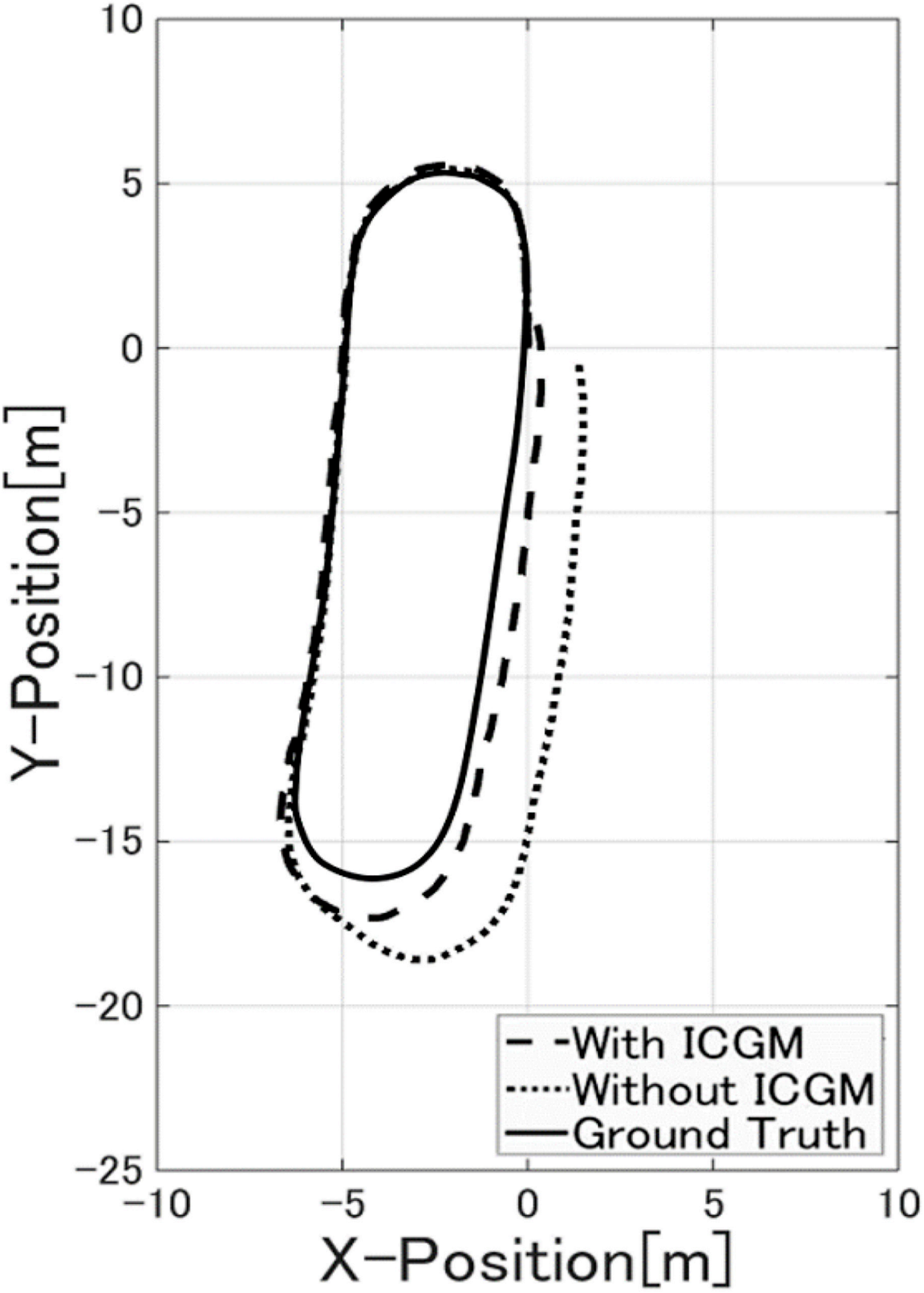


FIGURE 6.18: Shibuya station outdoor's comparison between ICGM based and Non-ICGM visual odometry.

6.3 Summary

This chapter described other applications of ICGM: hybrid SLAM and visual odometry. Results shows that ICGM works good in crowded public environment.

Chapter 7

Conclusion and Future Studies

Incremental Center of Gravity Matching (ICGM) is a novel method to extract robust visual-features from sequnce of images in in crowded public environment.

Using the proposed method, dynamic objects(pedestrians, cars etc.) can be ignored. Robust features extracted by the proposed method can improve performance of vision based localization.

This paper descried the basic algorithm of ICGM and its applications: appearance-only SLAM, hybrid SLAM and visual odometry.

Results of appearance-only SLAM, hybrid SLAM and visual odometry shows that the proposed method: Incremental Center of Gravity Matching (ICGM) works good in crowded public environment.

Experiments results proved that ICGM is a effective matching and feature extraction method. I believe that in the future, the proposed method should be more widely used.

Today, high-performance hand-held smart phones have become very popular. For the proposed method to possess high robustness while using hand-held devices, ICGM may be applied to many types of platforms (including hand-held smart phones) for navigation by pedestrians. This is our future goal.

Deciding the threshold Thr automatically is one of our important future study.

When compared to PIRF-Nav 2.0, the processing speed of the proposed method is possibly relatively fast. However, we are currently considering replacing SURF with a type of corner detector (FAST[35], HARRIS[36] etc.). Although SURF[4], SIFT[3] possesses more information for visual recognition, FAST[35] or HARRIS[36] is faster. Furthermore, ICGM can extract robust corner detectors on the basis of their geometric structure. Therefore, we intend to use FAST[35], HARRIS[36] for faster processing.

Bibliography

[1] A. Kawewong, S. Tangruamsub, and O. Hasegawa. Position-invariant robust features for long-term recognition of dynamic outdoor scenes. IEICE Transactions on Information and Systems, E93-D(9):2587–2601, 2010.

[2] Antonio Torralba, Aude Oliva, Monica S Castelhano, and John M Henderson. Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search. Psychol Rev, 113(4):766–786, October 2006. doi: 10.1037/0033-295X.113.4.766. URL http://dx.doi.org/10.1037/0033-295X.113.4.766.

[3] D.G. Lowe. Object recognition from local scale-invariant features. IEEE International Conference on Computer Vision, 2:1150–1157, 1999.

[4] H. Bay, T. Tuytelaars, and L.V. Gool. Surf: Speeded up robust features. ECCV, 3951:404–417, 2006.

[5] J. Beis and D. G. Lowe. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. Conference on Computer Vision and Pattern Recognition, pages 1000–1006, 1997.

[6] M. Cummins and P. Newman. Invited Applications Paper FAB-MAP: Appearance-Based Place Recognition and Mapping using a Learned Visual Vocabulary Model. In 27th Intl Conf. on Machine Learning (ICML2010), 2010.

[7] A. Kawewong, N. Tongprasit, S. Tangruamsub, and O. Hasegawa. Online incremental appearance-based slam in highly dynamic environments. Int. J. of Robotics Research, 30(1):33–55, 2011.

[8] N. Tongprasit, A. Kawewong, and O. Hasegawa. Pirf-nav 2:speeded-up online and incremental appearance-based slam in highly dynamic environment. IEEE Workshop on Applications of Computer Vision (WACV), 2011.

[9] A. Torralba, K.P. Murphy, W.T. Freeman, and M.A. Rubin. Context-based vision system for place and object recognition. In Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on, pages 273–280 vol.1, 2003.

[10] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In CVPR (2), pages 2169–2178. IEEE Computer Society, 2006. ISBN 0-7695-2597-0. URL http://dblp.uni-trier.de/db/conf/cvpr/cvpr2006-2.html#LazebnikSP06.

[11] Adrien Angeli, David Filliat, Stéphane Doncieux, and Jean-Arcady Meyer. Fast and incremental method for loop-closure detection using bags of visual words. IEEE Transactions on Robotics, 24(5):1027–1037, 2008. URL http://dblp.uni-trier.de/db/journals/trob/trob24.html#AngeliFDM08.

[12] A Kawewong. Pirf-nav: An online incremental appearance-based localization and mapping in dynamic environments. Ph.D Thesis, Tokyo Institute of Technology, 2010.

[13] Aram Kawewong, Sirinart Tangruamsub, and Osamu Hasegawa. Wide-baseline visible features for highly dynamic scene recognition. In Xiaoyi Jiang and Nicolai Petkov, editors, CAIP, volume 5702 of Lecture Notes in Computer Science, pages 723–731. Springer, 2009. ISBN 978-3-642-03766-5. URL http://dblp.uni-trier.de/db/conf/caip/caip2009.html#KawewongTH09.

[14] Aram Kawewong, Noppharit Tongprasit, and Osamu Hasegawa. Pirf-nav 2.0: Fast and online incremental appearance-based loop-closure detection in an indoor environment. Robotics and Autonomous Systems, 59(10):727–739, 2011. URL http://dblp.uni-trier.de/db/journals/ras/ras59.html#KawewongTH11.

[15] Noppharit Tongprasit, Aram Kawewong, and Osamu Hasegawa. Data partitioning technique for online and incremental visual slam. In Chi-Sing Leung, Minho Lee, and Jonathan Hoyin Chan, editors, ICONIP (1), volume 5863 of Lecture Notes in Computer Science, pages 769–777. Springer, 2009. ISBN 978-3-642-10676-7. URL http://dblp.uni-trier.de/db/conf/iconip/iconip2009-3.html#TongprasitKH09.

[16] Joan Solà, Thomas Lemaire, Michel Devy, Simon Lacroix, and André Monin. Delayed vs undelayed landmark initialization for bearing-only SLAM. In In Proceedings of the the IEEE Int. Conf. on Robotics and Automation workshop on SLAM - Workshops, 2005. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.109.4354.

[17] Max Pfingsthorn, Bayu Slamet, and Arnoud Visser. A scalable hybrid multi-robot SLAM method for highly detailed maps. pages 457–464, 2008. doi: 10.1007/978-3-540-68847-1_48. URL http://dx.doi.org/10.1007/978-3-540-68847-1_48.

[18] J. L. Blanco, J. A. Fernández-Madrigal, and J. Gonzalez. Toward a unified bayesian approach to hybrid Metric–Topological SLAM. Robotics, IEEE Transactions on, 24(2):259–270, April 2008. ISSN 1552-3098. doi: 10.1109/tro.2008.918049. URL http://dx.doi.org/10.1109/tro.2008.918049.

[19] H. Morioka, Y. Sangkyu, and O. Hasegawa. Vision-based mobile robot's slam and navigation in crowded environments. IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS), 2011.

[20] Richard I. Hartley. In defense of the Eight-Point algorithm. IEEE Trans. Pattern Anal. Mach. Intell., 19(6):580–593, June 1997. ISSN 0162-8828. doi: 10.1109/34.601246. URL http://dx.doi.org/10.1109/34.601246.

[21] Tony Lindeberg. Scale-space theory in computer vision. Kluwer Academic, 1 edition, December 1994. ISBN 0792394186. URL http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0792394186.

[22] T. Lindeberg. On the axiomatic foundations of linear Scale-Space: Combining Semi-Group structure with causality vs. scale invariance. In Gaussian Scale-Space Theory: Proc. PhD School on Scale-Space Theory. Kluwer Academic Publishers, 1994. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.16.6231.

[23] J. L. Blanco, J. A. Fernández-Madrigal, and J. Gonzalez. Toward a unified bayesian approach to hybrid Metric–Topological SLAM. Robotics, IEEE Transactions on, 24(2):259–270, April 2008. ISSN 1552-3098. doi: 10.1109/tro.2008.918049. URL http://dx.doi.org/10.1109/tro.2008.918049.

[24] Alfons H. Salden, Bart, and Max A. Viergever. Dynamic Scale-Space theories. In Scale-Space Theories in Computer Vision, pages 248–259, 1997. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.35.390.

[25] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. volume 1, pages 525–531, 2001. doi: 10.1109/iccv.2001.937561. URL http://dx.doi.org/10.1109/iccv.2001.937561.

[26] M. Wang and A. Knoesen. Rotation- and scale-invariant texture features based on spectral moment invariants. Journal of the Optical Society of America. A, Optics, image science, and vision, 24(9):2550–2557, September 2007. ISSN 1084-7529. URL http://view.ncbi.nlm.nih.gov/pubmed/17767226.

[27] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. volume 1, pages 525–531, 2001. doi: 10.1109/iccv.2001.937561. URL http://dx.doi.org/10.1109/iccv.2001.937561.

[28] G. W. Stewart. Matrix Algorithms. Society for Industrial and Applied Mathematics, January 1998. ISBN 978-0-89871-414-2. doi: 10.1137/1.9781611971408. URL http://dx.doi.org/10.1137/1.9781611971408.

[29] L. M. J. Florack, B. M. Ter Haar Romeny, J. J. Koenderink, and M. A. Viergever. General intensity transformations and differential invariants. Journal of Mathematical Imaging and Vision, 4(2):171–187, May 1994. ISSN 0924-9907. doi: 10.1007/bf01249895. URL http://dx.doi.org/10.1007/bf01249895.

[30] A. Baumberg. Reliable feature matching across widely separated views. In Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on, volume 1, pages 774–781, 2000. doi: 10.1109/cvpr.2000.855899. URL http://dx.doi.org/10.1109/cvpr.2000.855899.

[31] Frederik Schaffalitzky and Andrew Zisserman. Multi-view matching for unordered image sets, or "how do i organize my holiday snaps?". In ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part I, pages 414–431, London, UK, 2002. Springer-Verlag. ISBN 3540437452. URL http://portal.acm.org/citation.cfm?id=645315.649164.

[32] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 13(9):891–906, September 1991. ISSN 0162-8828. doi: 10.1109/34.93808. URL http://dx.doi.org/10.1109/34.93808.

[33] Multi-scale phase-based local features, volume 1, June 2003. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1211426.

[34] Schmid C. Mikolajczyk, K. A performance evaluation of local descriptors. CVPR, 2:257–263, 2003.

[35] Edward Rosten and Tom Drummond. Fusing points and lines for high performance tracking. In IEEE International Conference on Computer Vision, pages 1508–1515. Springer, 2005.

[36] C. Harris and M. Stephens. A combined corner and edge detector. In Proceedings of the 4th Alvey Vision Conference, pages 147–151, 1988.

[37] N. Hatao, R. Hanai, K. Yamazaki, and M. Inaba. Real-time navigation for a personal mobility in an environment with pedestrians. IEEE Int.Symposium on Robot and Human Interactive Communication, 2009.

[38] J. Muller, C. Stachniss, K. Arras, and W. Burgard. Socially inspired motion planning for mobile robots in populated environments. Proc.of International Conference on Cognitive Systems, 2008.

[39] O. Koch, M. R. Walter, A. S. Huang, and S. Teller. Ground robot navigation using uncalibrated cameras. Proc. of ICRA, 2010.

[40] Berthold K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. JOSA A, 4:629–642, 1987.

[41] M. A. Fischler and R. C. Bolles. Random sample consensus: aparadigm for model fitting with applications to image analysis and automated cartography. Commun. ACM, 24(6):381–395, 1981.

[42] E. W. Dijkstra. A note on two problems in connexion with graphs. Numerische Mathematik, 1(1):269–271, 1959.

[43] Andreas Geiger, Julius Ziegler, and Christoph Stiller. Stereoscan: Dense 3d reconstruction in real-time. In Intelligent Vehicles Symposium (IV), 2011.

[44] Bernd Kitt, Andreas Geiger, and Henning Lategahn. Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme. In Intelligent Vehicles Symposium (IV), 2010.