A Thesis for the Degree of Ph.D. in Engineering

Texture-Free Keypoint Matching and its Applications

August 2010

Graduate School of Science and Technology
Keio University

Hideaki Uchiyama

Abstract

Feature matching is one of the fundamental issues in computer vision and image processing. Recently, texture based keypoint matching is regarded as a generic approach for various objects. However, this approach does not always work because they need rich texture on the objects. In this thesis, two different approaches using geometric feature and temporal feature are investigated toward texture-free keypoint matching.

In the geometric feature based approach, on-line learning of geometric features for keypoint tracking is developed. Because the features are not invariant to the large range of views, the variety of the features needs to be learnt while moving a camera. Learning new geometric features contributes to wide base-line keypoint matching as keypoint tracking.

In the temporal feature based approach, the change of the brightness in temporal images is utilized as a feature. In order to generate the change, a blinking light is selected as a device. By setting unique blinking pattern into each light, matching lights captured in different views can be performed.

These two approaches are applied to three augmented reality systems and one photogrametric system. In an augmented reality based pool supporting system, camera pose estimation with respect to a pool table is performed using geometrical relationship of table corners. The geometric feature based keypoint tracking contributes to free camera movement with stable augmentation in augmented documents. In addition, the geometric feature can achieve map image retrieval and free camera movement in augmented maps. For the photogrammetric measurement of outdoor constructions in the dark, lights are utilized as markers for establishing correspondences in the dark images.

Acknowledgements

I would like to thank my main supervisor, Prof. Hideo Saito for conducting me to interesting research domains. I enjoyed developing new methods and applications, and making presentations and demonstrations. All experiences brought me up as a researcher.

As second supervisors, I would like to thank Prof. Guillaume Moreau and Prof. Myriam Servières. I had a great experience of a joint work with foreign researchers. Thanks to their grateful help, I really enjoyed and accomplished my research internship in Nantes, France.

I express my appreciation to the thesis committee members: Prof. Ken-ichi Okada, Prof. Issei Fujishiro and Prof. Yoshimitsu Aoki for evaluating my work. I will keep in mind their precious comments and advices for my future research life.

I also appreciate Prof. Shin-ichiro Haruyama, who gave an opportunity to join a project that tried to realize visible light communication to practical use. I learned the difficulties and practical problems for releasing commercial products.

I cannot say thanks to Dr. Julien Pilet enough. Even though I worked with him for less than one year, I learned many things about research planning, coding and technical writing. Because his idea is always interesting, innovative and impressive, I got stimulated many times.

I am so grateful to my colleagues in the laboratory. It is obvious that my researches could not be achieved without their help.

Finally, I would like to thank my family for their patience, encouragement and support during my Ph.D course.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

With the development of computer vision and image processing technologies, digital cameras have been considered and utilized as advanced sensors [28, 33]. The cameras can easily get installed on various devices such as mobile phones because they are becoming smaller and cheaper thanks to the downsizing and energy saving of Complementary Metal-Oxide Semiconductors (CMOS) and Charge-Coupled Devices (CCD). As represented by face authentication systems [105], the applications using cameras are gradually entering and supporting our daily life.

Sensing technologies based on computer vision and image processing are applied to the wide range of research domains. They are usually regarded as a fundamental process. Their early technologies contributed to the development of autonomous robots in robotics [35]. Especially, the term of machine vision is created to indicate the industrial and manufacturing uses of computer vision such that a robot arm is controlled by the analysis of a captured image. The purpose of these technologies is mainly sensing the environment. In other words, the data from the sensing is utilized as input to control systems.

Image synthesis has also been investigated to entertain people by showing never-seen-before images. For example, Image Based Rendering (IBR) is a representative approach to generate newly composed images from a set of images

Table 1.1: Classification of research domains. The category of input means that the output of a method is utilized as input to control systems. The category of output means that the result image is generated to show (output) to people.

Category	Research domains
Input	Remote Sensing, Object Detection [58], Object Recognition [22]
Output	Augmented Reality, Image Based Rendering, Inpainting [55]

in image and video media engineering [19]. Inamoto and Saito generated photo-realistic free viewpoint images of a soccer game by interpolating real camera images [37].

From the aspect of the use of the technologies, the research domains can be classified into two categories as follows: input and output. The examples of the research domains are introduced in Table 1.1. Because the applications of remote sensing and augmented reality are described in this thesis, the details of each research domain are introduced as follows.

Remote sensing is a general term indicating the acquisition of the numerical data of a target object or environment [39]. A 2D camera is utilized as one of the useful sensing devices because it can safely acquire the images to estimate the shape and reflectance property of the target without touching it in real-time. As another type of device, Laser Imaging Detection And Ranging (LIDAR) is also utilized to capture the 3D shape data of a far target using the radiation of short wavelength. The applications of these technologies are mainly the observation and monitoring systems of earth's surface and large objects [106, 56].

Augmented Reality (AR) is getting a lot of attention as a new research domain [6, 7]. The basic concept of AR is to show virtual objects into a real world using a live video see-through Head Mounted Display (HMD) [42] or other display devices [9, 90]. From the well-accepted definition of AR by Azuma [6], 3D displayed virtual objects should keep geometric, photometric and temporal consistencies with a real world in real-time for the interactivity with users. Because it is still difficult to solve all problems in all in one solution, each problem is solved individually. AR based applications are developed in wide practical domains such as advertisement, education and games [68].

1.2 Challenge

In the development of computer vision and image processing based applications, establishing correspondences between two or more images is one of the fundamental processes. This process is generically called feature matching. For example, Structure from Motion (SfM) needs to track the same points on a target object in different view images to estimate its 3D shape [24, 85, 69]. In image retrieval for augmented reality, correspondences between a query image and images in a database need to be established [82]. In addition to 2D-2D feature matching as described above, 2D-3D feature matching between an image and a 3D target object is also performed to estimate a camera pose with respect to the object as illustrated in Figure 1.1.

The technical challenge in natural feature matching is divided into three parts as follows: extraction, description and matching. Feature extraction is a process for detecting discriminative parts in an image. Keypoints (feature points, interest points) [32, 92, 97, 65, 88], line segments (edges) [83, 91, 41] and regions (blobs) [60, 66, 27] are typically utilized as features. After the features are extracted, the feature vector of each feature is described to measure similarity between two features. Because the accuracy of feature matching is significantly influenced by the description, the designed of a feature space is an important issue. Finally, the features in different images are compared using each feature vector to find the same feature. Feature matching is usually addressed as a nearest neighbor searching problem. In order to enhance the scalability of the method, feature matching is performed in pyramid images [12, 1].

The research direction of natural feature matching is mainly the improvement of the repeatability and stability against the change of elements such as illumination and viewpoint. From the aspects of computational costs and robustness to occlusion, keypoint based approaches have recently become the mainstream as described in Chapter 2.

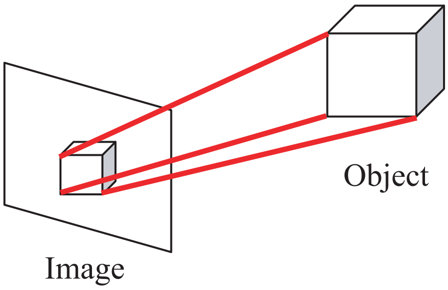


Figure 1.1: Feature matching. Camera pose estimation needs to establish 2D-3D correspondences between an image coordinate system and a world coordinate system.

1.3 Contribution

In the studies of keypoint matching, this thesis tackles texture-free keypoint matching as a technical contribution. As described in Chapter 2, traditional keypoint matching utilized local texture as a feature. In this thesis, two following directions are investigated.

- Geometric feature based keypoint tracking
- Temporal feature based keypoint matching

In general, texture based keypoint matching can be applied to various types of texture. However, the matching does not work well when the texture is binary because it lacks the variety of color levels. For example, texture based keypoint matching on text documents is difficult because their local texture is not discriminative as a feature. For the solution, geometric feature based wide base-line keypoint matching as keypoint tracking is developed.

As another type of keypoint matching, temporal feature is also investigated. If temporal images can be captured at each viewpoint, the temporal change of brightness at each pixel can be available as a feature. The change can be generated from blinking lights. In order to detect the same light in different viewpoints, temporal feature based keypoint matching is developed.

The design and development of applications based on developed methods are also the contribution in this thesis. The challenge of each application is described in parallel as follows.

Pool supporting system takes a marker-less approach toward constraint-free system. The geometrical relationship of table corners is utilized for camera pose estimation. The user having a handheld device can watch the supporting information overlaid on the pool table through the device.

Augmented documents is based on geometric feature based keypoint tracking for stable augmentation because text documents do not have rich texture. The user can watch virtual annotations as written on the document while moving a camera.

Augmented maps needs paper maps in which intersections are printed to use intersections as keypoints. By analyzing the local relationship of intersections, map image retrieval is performed to overlay 3D geographic data on the retrieved map.

Photogrammetric system using lights is developed for the measurement of outdoor constructions in the dark. In order to use lights as markers, temporal feature based keypoint matching is utilized.

1.4 Thesis Organization

This thesis is organized as following chapters. In Chapter 2, the related works of feature matching are reviewed to clarify the position of this thesis. In Chapter 3, the details of technical contributions about texture-free keypoint matching are described. In Chapter 4, an augmented reality based pool supporting system is described as a simple case of geometric feature based keypoint matching. In Chapter 5, an augmented reality system for text documents is described. This system is based on geometric feature based keypoint tracking. In Chapter 6, an augmented reality system on paper maps is described. This system utilizes geometric features not only for augmented reality purposes but also for map image retrieval. In Chapter 7, a photogrametric system using temporal feature is described. Finally, Chapter 8 concludes this thesis.

Chapter 2

Related Works

2.1 Marker Based Approach

A marker is traditionally utilized as an artificial feature. Because the marker has specific shape and texture, the extraction of markers is easier than those of natural features. The stability of their recognition is also better. At the beginning of related works for feature matching, a marker based approach is introduced. The main issue of this approach is the design of markers.

2.1.1 Marker Design

The world's first marker system was developed by Rekimoto in 1998 [87]. In this system, each marker had square black and white patterns. The order of the patterns corresponds to a bit sequence for marker identification. The extended version of this system was installed into a commercial product called "EYE OF JUDGEMENT" by Sony Computer Entertainment Inc. in 2006.

In 1999, Kato and Billingusrt developed ARToolKit [42], which was the most famous toolkit for developing augmented reality applications. In this toolkit, users can draw any pattern inside the black frame of the marker as illustrated in Figure 2.1. This toolkit was ported into several development environments such as JAVA [30], Pocket PC [103] and mobile devices [104].

MR (Mixed Reality) Platform was developed for sensor fusion with markers in 2002 [100]. This platform enabled developing applications using a camera with a magnetic sensor for HMD. Markers used in this platform had a regular hexagonal grid pattern to make more patterns efficiently.

ARTag is a marker system that uses signal processing theory for making markers [24]. The markers have a bi-tonal pattern including checksums and forward error correction for robust marker identification.

Wagner et al. developed three different types of following markers: frame markers, split markers and dot markers [101]. These markers had better visibility and robustness than the traditional square markers.

2.1.2 Applications

A marker was utilized in early augmented reality applications. For example, Matysczok et al. developed an augmented reality bowling system [64]. Because augmented reality based games were quite novel, Andersen et al. investigated the influence of these games for people [3]. For medial uses, several training and supporting systems were developed [94, 25]. In industries, virtual factory planning was conducted using markers such that the arrangement of equipment was virtually overlaid on markers to simulate the best arrangement in the vacant room [21, 80].

In addition to augmented reality, a marker was utilized as a landmark for controlling robots [23, 110]. By making a link between each marker and its location information, robots can get the location information from a marker.

2.2 Texture Based Keypoint Matching

A marker is not preferable for several applications because it needs specific shape and pattern. In order to utilize normal texture, natural feature based approaches are becoming a main stream. Especially, keypoint based approaches have a good characteristic such that keypoint matching can be performed even though some parts of the texture are not visible as occlusion.

In this section, texture based approaches using local descriptors are described. The process is divided into three parts; extraction, description and matching as follows.

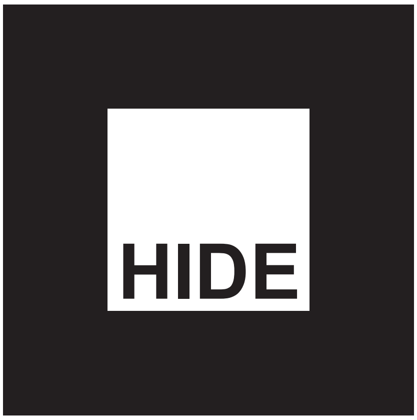


Figure 2.1: Marker for ARToolKit [42]. The main characteristic of ARToolKit is the flexible design of the markers. Inside the black frame, any kinds of asymmetric pattern can be inserted.

2.2.1 Extraction

Keypoint extraction consists in finding pixels which have different appearance from other pixels. Harris corner detector [32] and SUSAN [97] are the most famous detectors. These are utilized as benchmarks for a long time. Shi and Tomasi developed the extraction of good features to track included in the Kanade-Lucas-Tomasi (KLT) tracker [92]. In order to keep the stability against the change of views, Mikolajczyk and Schmid developed a extraction method with scale and affine invariant properties [65]. Rosten and Drummond took a new approach with machine learning techniques for fast extraction, called FAST [88].

Region detectors can also be regarded as a keypoint detector because the center of each region can be dealt as a keypoint [66]. Maximally Stable Extremal Regions (MSER) is a method with affine invariant region extraction, which has the stability and repeatability in multi scales [60, 27].

2.2.2 Description

The simplest descriptor for describing a feature vector is a local image patch, which is utilized to find the same region by template matching. The feature vector is an array of pixel brightness. If the difference of pixel colors between two regions is small, two regions can be judged as the same region. A detailed survey for texture based local descriptors is reported in [66].

The methods including both extraction and description were also developed. The representative methods are Scale Invariant Feature Transform (SIFT) [54] and Speeded Up Robust Features (SURF) [8]. In SIFT, keypoints are extracted from Difference of Gaussians (DoG), and their 128 dimensional descriptor is computed from a histogram of oriented gradients. Compared to SUFT, SURF is designed to be faster, but lower quality of matching. Keypoint extraction in SURF is based on a hessian matrix, and its description is based on Haar wavelet responses. These descriptors are well-designed to be robust to the changes of illumination, rotation and translation and works on rich textures.

SIFT cannot normally achieve interactive frame rate with a normal computer. Because both extraction and description can be parallelized, Graphics Processing Unit (GPU) is utilized to speed up the processes [95]. Wagner et al. modified SIFT to apply to a mobile phone within the limited resources of memory and cache [102].

2.2.3 Matching

The simplest method is full searching. Because it takes much time depending on the number of samples and the dimension of a feature vector, several solutions to reduce the computational costs are developed.

Approximate Nearest Neighbor (ANN) is an approximate nearest neighbor searching method based on kd-trees and box-decomposition trees [5]. Because distance computation is performed to compare two vectors, the retrieval costs depend on the dimension of a feature vector. Regarding tree structures, several approaches are sought such as randomized trees [49], a recursive k-mean tree [75], multiple randomized kd-trees [70] and random ferns [79].

Locality Sensitive Hashing (LSH) is another approximate searching method based on probabilistic dimension reduction with a hash scheme [17]. The computational cost of LSH is always $O(1)$ while the nearest neighbor points may not be found. The design of the hash function is a main issue.

The example of keypoint matching by SIFT is illustrated in Figure 2.2.

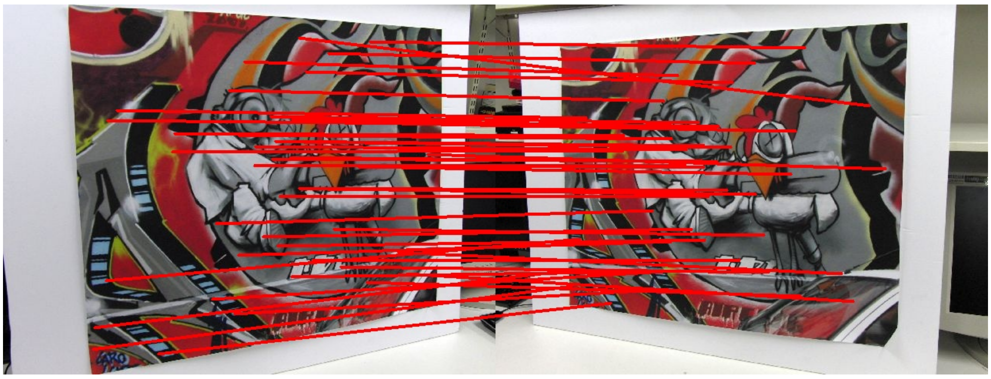
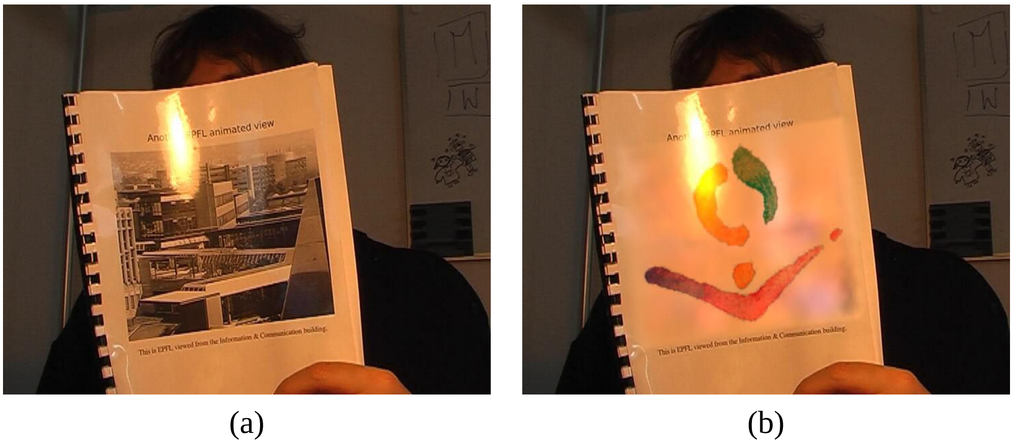


Figure 2.2: Keypoint matching by SIFT [54]. Keypoint extraction and description are performed in both images at first. For each keypoint of the left image, a corresponding keypoint in the right image is searched in matching. In the matching result, there are wrong correspondences because there are similar textures at different points.

2.2.4 Applications

Natural keypoint based approaches are utilized in many research domains because they have fewer constraints than the marker based approach. For the movement of autonomous robots, Visual Simultaneous Localization And Mapping (Visual SLAM) using natural keypoints was developed [18]. Visual SLAM is a method for enabling a large camera movement by estimating a camera pose and reconstructing the 3D coordinates of keypoints alternately. Klein and Murray extended Visual SLAM for augmented reality purposes called Parallel Tracking and Mapping (PTAM) [43]. In PTAM, 3D reconstruction and keypoint tracking are performed in different threads. The tracking thread always runs to track a camera pose. In contrast, 3D reconstruction is performed only when enough disparity occurs.

From the cloud of matched natural keypoints, the shape estimation of non-rigid surface estimation is also achieved [81]. In Figure 2.3, the shading is considered for realistic augmentation.


Ano PFL animated view
This is EPFL viewed from the Information & Communication building.
Ano PFL animated view
This is EPFL viewed from the Information & Communication building.

(a) (b)

Figure 2.3: Non-rigid surface augmentation. (a) The degree of curvature is estimated from the cloud of correspondences. (b) The virtual log is overlaid on the picture. The specularity of the input image is reflected in the result. Courtesy of Julien Pilet [81].

2.3 Geometric Feature Based Keypoint Matching

Texture based approaches are effective for rich textures including various patterns of colors. However, they did not work well for binary textures due to the lack of the variety of colors. In order to deal with this case, a keypoint matching method based on geometric features is developed, which is called Locally Likely Arrangement Hashing (LLAH) [72, 73, 74, 71]. This method utilizes the local geometrical relationship of neighbor keypoints as a feature. In this section, the details of LLAH are described.

2.3.1 Description

Suppose two different clouds of keypoints are already extracted. For example, the center of each word is a keypoint for text documents.

In Figure 2.4, **t** is a target keypoint. First, n nearest neighbor keypoints of the target are selected as **abcdefg** $(n = 7)$. The selecting order of the neighbor keypoints is defined as starting from **a** in an anti-clockwise fashion as illustrated in Figure 2.4. Next, m keypoints out of the n keypoints are selected as **abcde** $(m = 5)$. The number of combinations is ${}_nC_m = \frac{n!}{m!(n-m)!}$, and is equal to the number of descriptors for one keypoint.

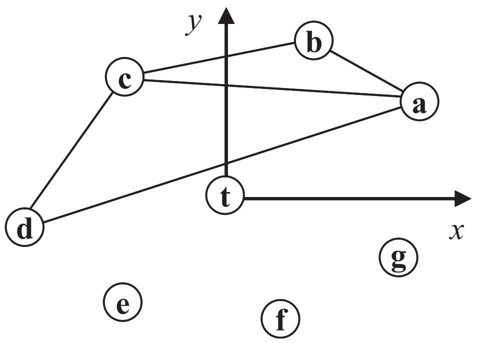


Figure 2.4: Descriptors in LLAH. The descriptors for one point are computed from the combination of neighbor points. The order of the combination can be determined beforehand.

From m keypoints, l keypoints are selected to compute a geometric invariant. The examples are a cross ratio as a perspective invariant [72] and a ratio of two triangles as an affine invariant [73]. The number of combinations is ${}_mC_l$ and is equal to a descriptor dimension.

As an example of a geometric invariant, a ratio of two triangles is selected to explain the way of computing one descriptor. From **abcde**, **abcd** is selected $(l = 4)$. The selecting way of the four points is pre-defined such as **abcd**, **abce**, **abde**, **acde** and **bcde**. For each combination, two triangles are generated by a pre-defined selection such as **abc** and **acd** for **abcd**. Because the number of combination is ${}_mC_4$, the descriptor dimension is ${}_mC_4$.

For quick retrieval, a hash scheme is utilized to reduce the descriptor dimension. The ${}_mC_l$ dimensional descriptor is converted into one dimensional index by using following equation:

$$Index = \left(\sum_{i=0}^{{}_mC_l-1} r_{(i)} k^i \right) \bmod H_{size} \tag{2.1}$$

where $r_{(i)}(i = 0, 1, ..., _m C_l - 1)$ is a quantized value of a geometric invariant, k is quantization level and H_{size} is hash size.

As a result, ${}_nC_m$ indices are computed for one keypoint.

Table 2.1: Structure of index database. Because ${}_nC_m$ indices are computed for one keypoint, a keypoint ID are stored at ${}_nC_m$ indices.

Index	Keypoint IDs
10	11, 46, 56
15	11, 92

2.3.2 Matching

The structure of a descriptor database is a table as described in Table 2.1 because a descriptor is equivalent to an index. This table can be regarded as an inverted index. Compared with the matching in texture based approaches, the matching in LLAH is simple such that keypoint matching is performed by finding the same index between different views. This means that the matching in LLAH does not need to perform approximate neighbor searching.

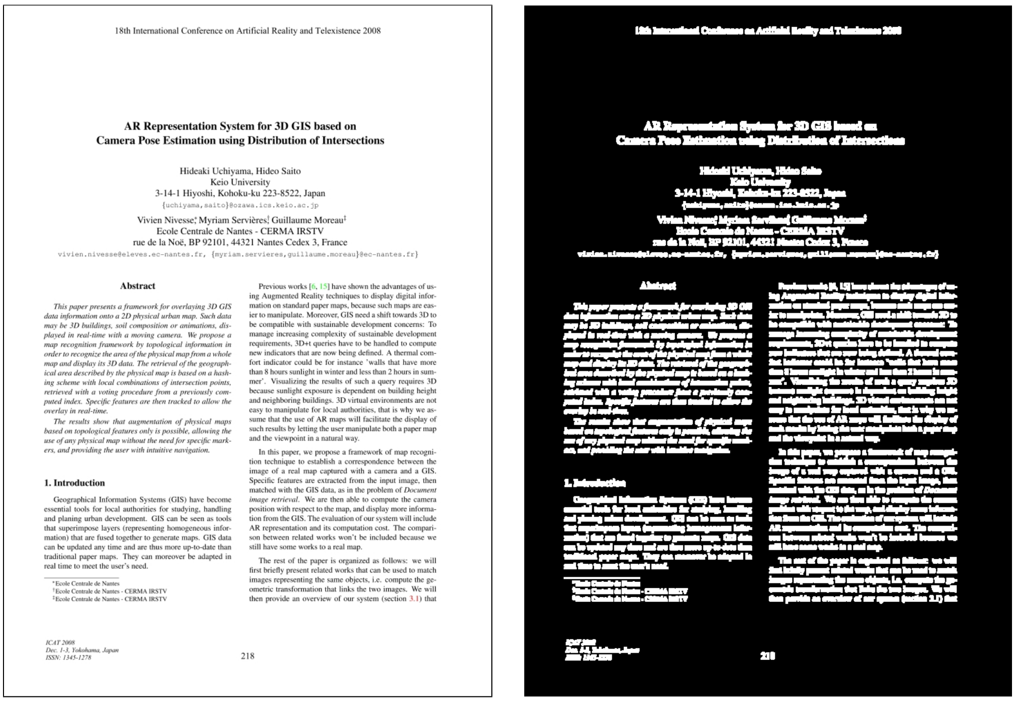
First, keypoints in one image are extracted and their keypoint IDs are stored in the index database. In the matching process, keypoints in the other image are extracted. For each keypoint, ${}_nC_m$ keypoint IDs are firstly retrieved using its ${}_nC_m$ indices. Using the retrieved keypoint IDs, the histogram of keypoint ID versus the number of counts is generated. From the histogram, one corresponding keypoint ID is determined by selecting the maximum number of counts.

2.3.3 Applications

LLAH is mainly designed for tackling document image retrieval [72, 73, 74, 71], which is an application of image retrieval to documents.

In order to make a document database, digital documents are firstly prepared as illustrated in Figure 2.5(a). For each document, a binary image is generated using an adaptive thresholding method with a fixed size filter to extract word regions as illustrated in Figure 2.5(b). The center of each word is utilized as a keypoint. For each keypoint, indices (descriptors) are computed from the neighbor keypoints to make the index database.

In the retrieval, one of the documents is captured from a nearly top view. From the captured image, keypoints are extracted by using the adaptive thresholding



18th International Conference on Artificial Reality and Telexistence 2008

AR Representation System for 3D GIS based on
Camera Pose Estimation using Distribution of Intersections

Hideaki Uchiyama, Hideo Saito
Keio University
3-14-1 Hiyoshi, Kohoku-ku 223-8522, Japan
{uchiyama,saito}@ozawa.ics.keio.ac.jp

Vivien Nivesse[*], Myriam Servières[†], Guillaume Moreau[‡]
Ecole Centrale de Nantes - CERMA IRSTV
rue de la Noë, BP 92101, 44321 Nantes Cedex 3, France
vivien.nivesse@eleves.ec-nantes.fr, {myriam.servieres,guillaume.moreau}@ec-nantes.fr}

Abstract

This paper presents a framework for overlaying 3D GIS data information onto a 2D physical urban map. Such data may be 3D buildings, soil composition or animations, displayed in real-time with a moving camera. We propose a map recognition framework by topological information in order to recognize the area of the physical map from a whole map and display its 3D data. The retrieval of the geographical area described by the physical map is based on a hashing scheme with local combinations of intersection points, retrieved with a voting procedure from a previously computed index. Specific features are then tracked to allow the overlay in real-time.

The results show that augmentation of physical maps based on topological features only is possible, allowing the use of any physical map without the need for specific markers, and providing the user with intuitive navigation.

1. Introduction

Geographical Information Systems (GIS) have become essential tools for local authorities for studying, handling and planing urban development. GIS can be seen as tools that superimpose layers (representing homogeneous information) that are fused together to generate maps. GIS data can be updated any time and are thus more up-to-date than traditional paper maps. They can moreover be adapted in real time to meet the user's need.

Previous works [6, 15] have shown the advantages of using Augmented Reality techniques to display digital information on standard paper maps, because such maps are easier to manipulate. Moreover, GIS need a shift towards 3D to be compatible with sustainable development concerns: To manage increasing complexity of sustainable development requirements, 3D+t queries have to be handled to compute new indicators that are now being defined. A thermal comfort indicator could be for instance 'walls that have more than 8 hours sunlight in winter and less than 2 hours in summer'. Visualizing the results of such a query requires 3D because sunlight exposure is dependent on building height and neighboring buildings. 3D virtual environments are not easy to manipulate for local authorities, that is why we assume that the use of AR maps will facilitate the display of such results by letting the user manipulate both a paper map and the viewpoint in a natural way.

In this paper, we propose a framework of map recognition technique to establish a correspondence between the image of a real map captured with a camera and a GIS. Specific features are extracted from the input image, then matched with the GIS data, as in the problem of *Document image retrieval*. We are then able to compute the camera position with respect to the map, and display more information from the GIS. The evaluation of our system will include AR representation and its computation cost. The comparison between related works won't be included because we still have some works to a real map.

The rest of the paper is organized as follows: we will first briefly present related works that can be used to match images representing the same objects, i.e. compute the geometric transformation that links the two images. We will then provide an overview of our system (section 3.1) that

[*] Ecole Centrale de Nantes
[†] Ecole Centrale de Nantes - CERMA IRSTV
[‡] Ecole Centrale de Nantes - CERMA IRSTV

ICAT 2008
Dec. 1-3, Yokohama, Japan
ISSN: 1345-1278

218

(a) (b)

Figure 2.5: Document image retrieval. (a) Digital document images are generated from PDF. (b) By applying adaptive thresholding to (a), word regions are extracted as white regions. The center of each word regions is a keypoint.

method, and their indices are computed. By using the indices, keypoint matching is performed to retrieve the corresponding document of the captured image from the document database.

2.4 Sensor Fusion

In previous sections, the described methods utilized only a camera. In addition to a camera, several types of sensors are utilized together because the sensors can interpolate the information which the camera does not have [11].

For example, Global Positioning Systems (GPS) provide a position by latitude and longitude. An electron compass provides a direction such as north, south, east and west. A gravity sensor provides an angle with respect to the gravity. Because GPS work in only outdoor cases, wireless network is instead utilized in

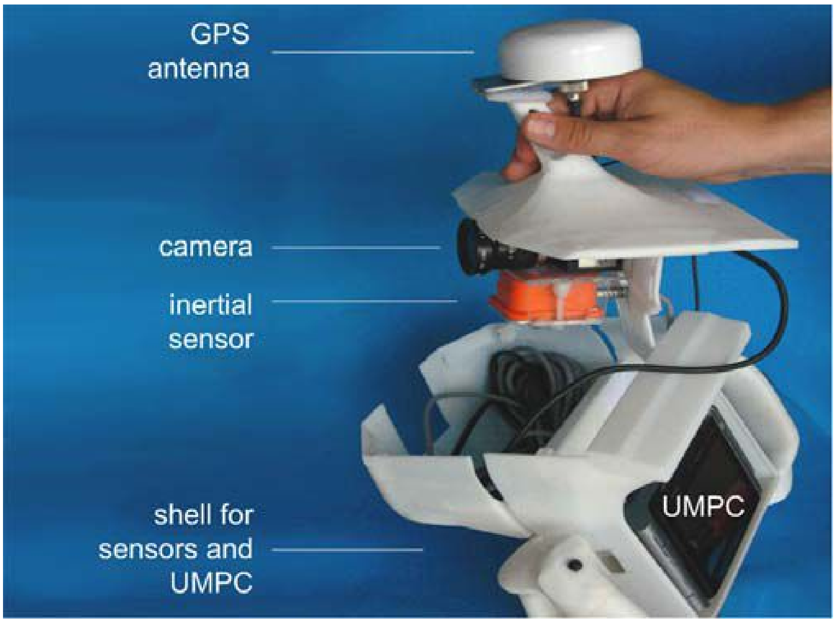


Figure 2.6: Outdoor AR platform. Several sensors are fused to estimate a global camera pose. The augmentation result is displayed on the UMPC. Courtesy of Gerhard Schall [89].

indoor cases [76]. A gravity sensor is sometimes utilized to assist camera pose estimation for a normal camera. The sensor is able to eliminate the degree of freedom, which is called an inclination constraint. By using the constraint, camera pose estimation will be performed from two 2D-3D correspondences [45] or some 2D-3D line correspondences [46].

The sensor fusion is applied to a vehicle navigation [96] and an outdoor augmented reality system [89, 111] as illustrated in Figure 2.6.

2.5 Keypoint Tracking

Keypoint tracking is related with keypoint matching. Because these processes may be confused, the meaning of each process is clarified in this section.

In Figure 2.7(a), camera pose estimation is individually performed for the first and second frames by finding 2D-3D correspondences. No information about the relationship between two frames is utilized. This approach utilizes only keypoint matching. In Figure 2.7(b), the second camera pose is computed using the first

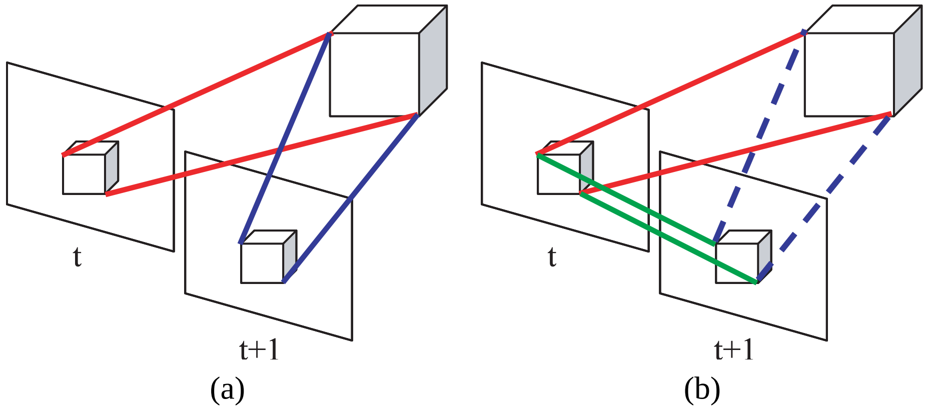


Figure 2.7: Camera tracking. (a) A camera pose is always estimated by finding 2D-3D correspondences. The information of a previous image is not utilized. (b) A camera pose is estimated by tracking 2D-2D correspondences between t and t+1 frames. This approach needs an initial camera pose computed by (a).

camera pose. The first camera pose is computed from 2D-3D correspondences by keypoint matching. Then, a second camera pose is computed by using the first camera pose and 2D-2D correspondences between first and second frames. If keypoint matching is applied to two consecutive frames as described in Figure 2.7(b), it can be called keypoint tracking.

2.6 Thesis Position

From the technical point of view, this thesis investigates two different directions as follows: keypoint tracking based on geometric features and keypoint matching based on temporal features. As described in Figure 2.5, keypoint tracking can be regarded as a part of keypoint matching. In this thesis, two directions described above are summarized in Figure 2.8.

Texture based approaches are becoming a mainstream in feature matching, and have already been well-studied. On the other hand, geometric feature based keypoint matching is not much investigated. For example, this approach has a low tolerance against the change of viewpoints. If a camera is moved, keypoint matching using geometric features is failed. In order to solve this problem, geometric feature based keypoint tracking is developed toward handling the large range of

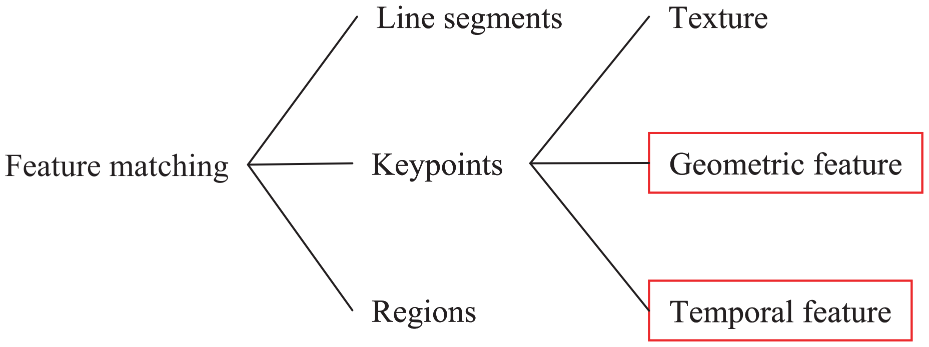


Figure 2.8: Thesis position. This thesis focuses on two different directions: geometric feature and temporal feature.

Table 2.2: Marker vs. natural feature. The advantages and disadvantages of each approach are opposite.

Approaches	Advantages	Disadvantages
Marker	Cost, Stability	Design
Natural feature	Design	Cost, Stability

viewpoint changes.

Also, temporal feature based keypoint matching is investigated as another type of keypoint matching compared with texture or geometric feature based approaches. In this approach, a blinking light is utilized as a keypoint. If it is dark, the methods described in previous sections never work because the objects are not visible. Even though this seems to be natural phenomenon, this indication was given by a construction company as a problem. The company wants to measure the specific parts of the large construction in the dark for monitoring purposes. Because a light emits by itself, it can be extracted in the dark. In order to utilize blinking pattern as a feature, temporal feature based keypoint matching is developed.

From the application point of view, this thesis takes both marker and natural feature based approaches. These approaches have advantages and disadvantages as described in Table 2.2.

The marker based approach has the advantages of computational costs and recognition stability. However, the design of the marker is strictly limited. In contrast, any rich texture can be acceptable in the natural feature based approach. Compared with markers, natural features cannot stably be extracted and recognized with more computational costs. Considering these characteristics, four applications described in this thesis are designed as follows.

Pool supporting system is based on a natural feature based approach because no special equipment should be installed. From natural features included in the environment of a pool game, table corners are selected as pre-selected natural keypoints.

Augmented documents can utilize physical text documents because the method is based on the extended version of a natural keypoint based matching. Toward practical use, normal documents are desirable.

Augmented maps needs paper maps in which intersections are printed due to the stable extraction of intersections. Compared with traditional markers, intersection dots have better visibility. Because the main purpose of this system is map image retrieval, printing markers is selected for the stability.

Photogrametric system using lights is designed for a marker system for measuring specific locations. Using lights as markers for outdoor photogrammetry is a main contribution.

Chapter 3

Texture-Free Keypoint Matching

3.1 Geometric Feature Based Keypoint Tracking

As described in Section 2.3, LLAH as geometric feature based keypoint matching has already developed. However, it cannot be applied to wide range of keypoint matching (keypoint tracking) because the features are not invariant to the large range of views. In order to utilize geometric features for keypoint tracking, on-line geometric feature learning is developed such that the variety of the features is tracked while moving a camera. In other words, the on-line learning process is merged into LLAH. For the explanation of this section, a document is selected as a reference.

3.1.1 Overview

The overall process of keypoint tracking with geometric feature learning is illustrated in Figure 3.1. As a pre-process, an initial index database is prepared from a top view image as a reference. From the image, keypoints and their indices are computed and stored in the database. In the tracking process, keypoint matching between input images and a reference is performed.

First, an initial camera pose with respect to a document is computed using LLAH with the initial index database. After initialization, the on-line geometric feature learning starts. If a camera pose is computed, geometric features (indices) of each keypoint in the image can be re-computed. By updating these re-computed

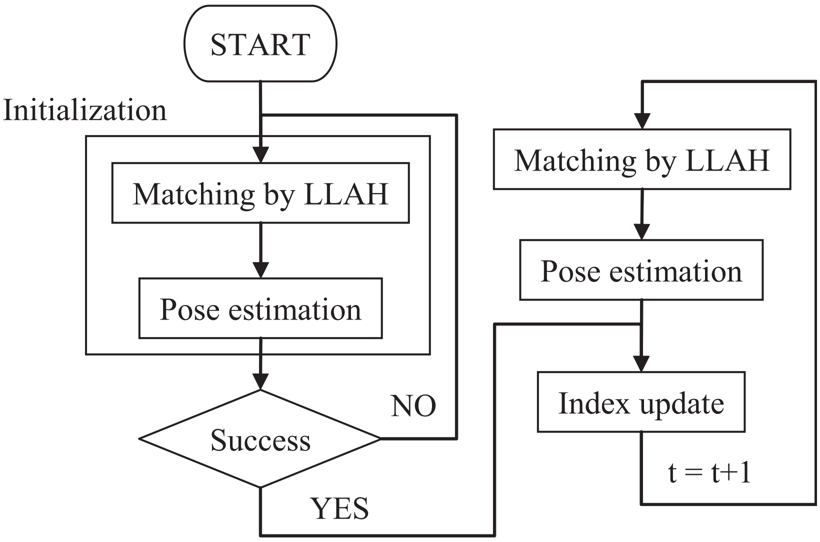


Figure 3.1: On-line geometric feature learning. Initialization is equivalent to the original LLAH for document image retrieval [72] such that camera pose is computed from the matching result of LLAH. After the initialization is done, on-line learning process starts. In index update, the descriptors of each keypoint are recomputed and updated for the next frame.

indices into the index database, the updated database can be utilized from the next frame. The detailed processes are described from the next section.

3.1.2 Fast Collection of Neighbors

In order to compute geometric features, the neighbors of a keypoint are necessary. If the distances between a keypoint with all keypoints are computed, the computational cost is $O(N^2)$, where N is the number of keypoints. The computation of all possible distances would imply larger computational cost. For fast computation of geometric features, the reduction of the cost for collection of neighbor keypoints should be performed. By limiting the candidates for distance computation, the cost is reduced as follows.

As a pre-processing phase, the size of a captured image is divided into square regions by segmenting them at a uniform interval as illustrated in Figure 3.2. When the keypoints are extracted in the captured image, the region to which each keypoint belongs is computed. In addition, each region keeps the list of keypoints

		j	k	l	m	n	
		y	b	c	d	o	
		x	i	a	e	p	
		w	h	g	f	q	
		v	u	t	s	r	

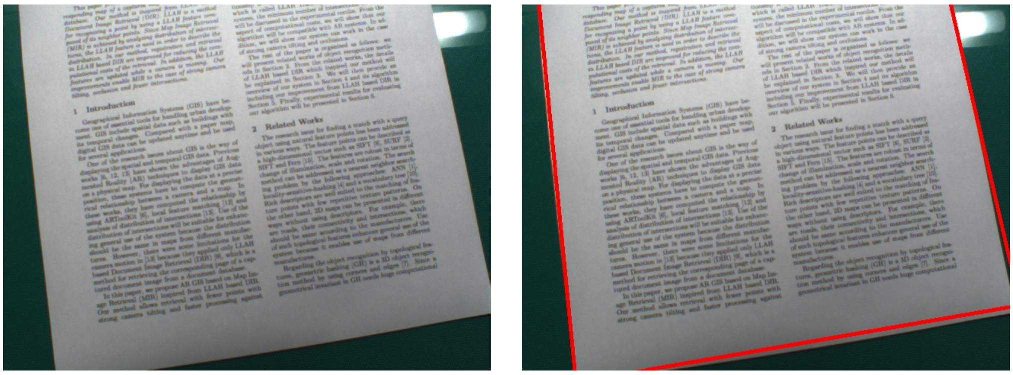
Figure 3.2: Fast collection of neighbors. The image region is divided into square regions beforehand. If a keypoint is extracted in **a**, candidates of neighbor points are collected from region **a** to region **i**. If they are not sufficient, the candidates are collected from region **j** to region **y**.

included in that region.

When the collection of the neighbors of a keypoint is performed, potential candidates are collected from the surrounding regions. For example, the candidates are extracted from region **a** to region **i** when a target keypoint belongs to **a** in Figure 3.2. If the number of candidates is less than the required number, more candidates from more surrounding regions are collected. When the number is more than the required number, the neighbor points are selected from the candidates by computing each distance.

3.1.3 Initialization

In the initialization, a paper document is captured from a nearly top view as illustrated in Figure 3.3(a). Keypoint matching between the captured image and the references in the database is performed by LLAH. From the correspondences, a camera pose is computed as described in Appendix. Because outliers may be included in the correspondences, RANSAC is applied to the process of the computation to eliminate the outliers [26]. If the camera pose is not successfully computed, the initialization is repeated until it succeeds. Figure 3.3(b) illustrates the initialization result by overlaying a document frame.



(a) (b)

Figure 3.3: Initialization. (a) For the initialization, a camera pose should nearly be top view because the initial database is generated from the top view image. (b) For showing that the initialization is succeeded, the colored frame of the document is overlaid.

3.1.4 Index Update

After the initialization is succeeded, the process moves to index update.

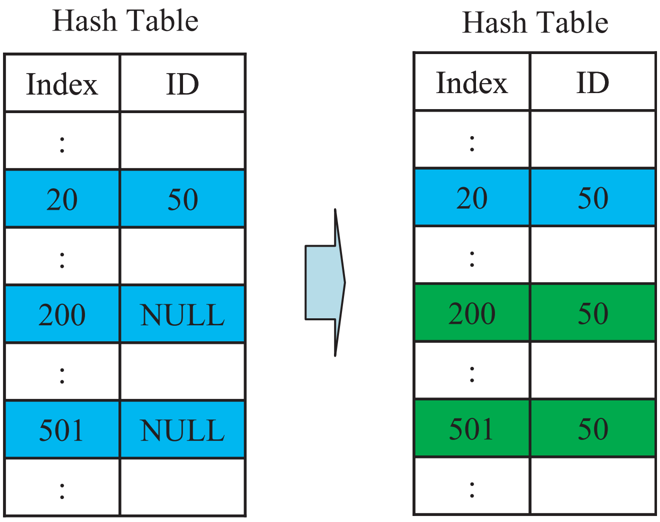
If the update is not included, the range of camera movement is limited to a nearly top view because the index database has the geometric feature of the top view. This means that matching by LLAH cannot be succeeded for a tilted view. For solving this problem, on-line learning process is developed.

Matching by Projection

First, keypoints of the top view image are projected onto the captured image by using the computed camera pose. From the projection, each keypoint in the image can have a correspondence with one of the projected keypoints. If the distance between a keypoint in the image and the nearest projected keypoint is close, the correspondence is judged as matched. This means that keypoints in the image can have a keypoint ID of the projected keypoint in the top view image.

Adding New Indices

In matching by LLAH, each keypoint had ${}_nC_m$ indices. At these indices, there may be NULL when an index is never computed before as illustrated in Figure 3.4.



Hash Table

Index	ID
:	
20	50
:	
200	NULL
:	
501	NULL
:	

Hash Table

Index	ID
:	
20	50
:	
200	50
:	
501	50
:	

Figure 3.4: Index update. For each keypoint, ${}_nC_m$ indices are computed in the descriptor computation of LLAH. For example, 20, 200 and 500 are computed in the left table. If the keypoint gets the keypoint ID of the nearest projected keypoint, the keypoint ID is inserted into NULL as described in the right table.

By inserting the keypoint ID of the projected keypoint at NULL, the empty space is efficiently used. The updated index database is utilized in matching by LLAH from the next frame. This updating process is regarded as on-line learning.

When a camera is close to a top view, the index update does not much occur because the indices of a top view are already stored. In contrast, the update occurs at tilted views because different geometric features can be computed.

If index update is performed for every frame, over-learning may occur because the number of updated indices depends on the number of frames. The influence of index update is discussed in experiments.

3.1.5 Experiments

The implementation is performed on a laptop with Intel Core 2 Duo 2.2GHz, 3GB RAM and 640×480 pixels' camera. The intrinsic parameters of the camera and the radial distortion of the lens are calibrated beforehand.

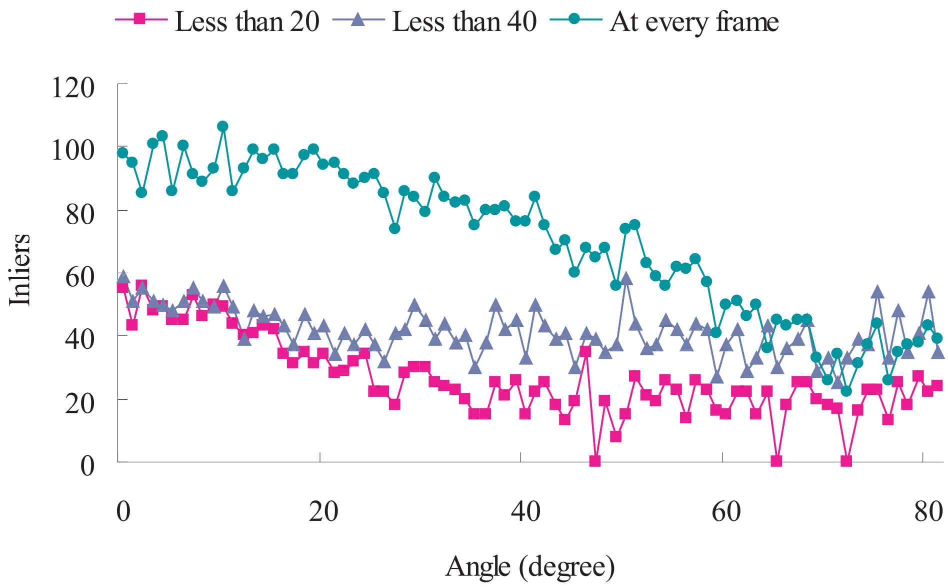


Figure 3.5: Influence of thresholds. Three different thresholds are investigated. Even though the threshold is less than 20, the number of the inliers is almost more than 20, which is enough for computing a camera pose. When a camera is tilted as 80 degrees, a camera pose is still computed.

Influence of Index Update

First, the influence of index update for the index database is investigated. In this experiment, several thresholds for performing index update are tested to judge whether index update at every frame is necessary or not. The parameters for LLAH are as follows: $n = 6$, $m = 5$, $l = 4$, $k = 32$ and $H_{size} = 2^{15} - 1$. A camera was rotated from a top view centering on the center of the document.

Figure 3.5 illustrates the number of the inliers (after RANSAC based camera pose estimation) while moving a camera for each following cases: update when the number of the inliers is less than 20, less than 40, and update at every frame. For all cases, the number of the inliers is almost more than 20, which is enough for computing a camera pose. From the point of camera pose estimation, the difference between all cases was not appeared. However, a major difference appeared for the number of the updated indices for each threshold.

As illustrated in Figure 3.6, the increase of the updated indices depends on the

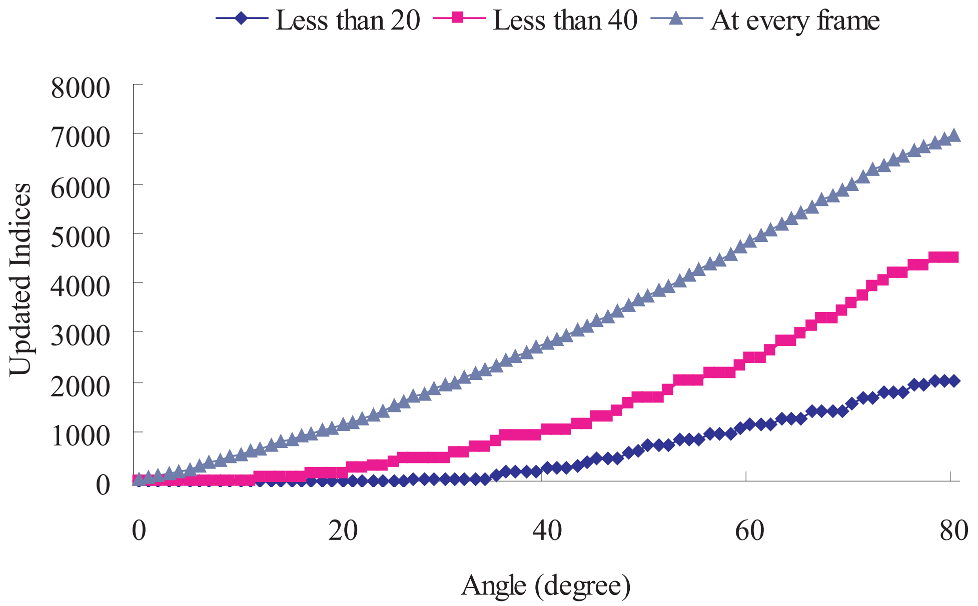


Figure 3.6: Updated indices. The number of the updated indices depends on the number of times of index update. Even though the number of the updated indices for update at every frame is over three times more than that of the case that the threshold is less than 20, a camera pose was computed for the cases.

number of times of index update. The number of the updated indices for update at every frame is over three times more than that of the case that the threshold is less than 20. However, a camera pose is continuously computed for all cases. This means that update at every frame is not always necessary for camera pose estimation. As discussed in [72], the appropriate parameters should be selected from several experiments.

Comparison with LLAH

Compared with LLAH, the method described in this section can be regarded as fusion of LLAH and index update. By comparing two methods, the effectiveness of index update is evaluated.

As illustrated in Figure 3.7, the number of the inliers gradually decreased in LLAH. Camera pose estimation failed around 30 degrees where the number of

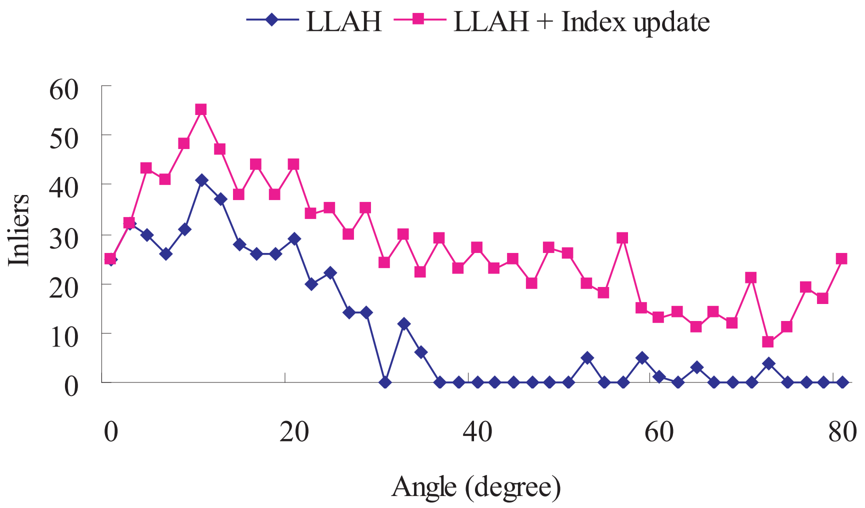


Figure 3.7: Advantage of index update. In LLAH, the range of camera movement is limited to 30 degrees because keypoint matching worked within 30 degrees. By merging index update with LLAH, a camera pose can be computed even under 80 degrees.

the inliers is 0. This means that geometric features of a top view are almost valid from 0 degree to 30 degrees. In the fusion of LLAH and index update, the number also decreased. However, camera pose estimation is successfully performed from 0 degree to 80 degrees because the number of the inliers is always more than the minimum number for camera pose estimation.

Comparison with SURF

For comparison with other methods, SURF implemented on OpenCV [107] is selected because it can run in real-time for AR systems compared with SIFT.

In tracking by SURF, a document image is prepared and printed on an A4 paper. The image resolution is selected as 672×950 because this made the best result compared with other resolutions. For each captured image, correspondences in the document image are established by the descriptors in SURF, which can be regarded as matching between the top view and the captured image.

As illustrated in Figure 3.8, the number of the inliers in SURF is less than that of LLAH even when a camera is set at nearly top view because general local

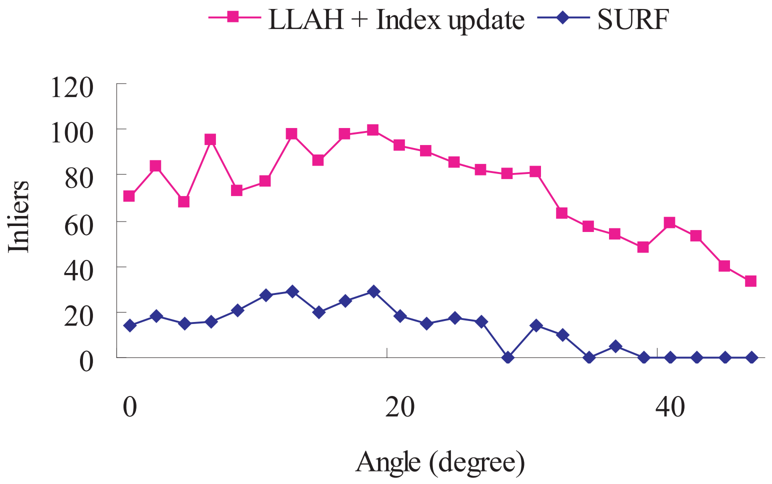


Figure 3.8: Comparison with SURF. Because general local descriptors do not work well for text, the number of inliers is less than that of LLAH. For text, the fusion of LLAH and index update produced the best result for keypoint tracking.

descriptors do not work well for text. Keypoint matching failed after 32 degrees because SURF descriptors are not invariant to the perspective distortion. In order to handle keypoint tracking for text, the fusion of index update and LLAH is the best solution.

3.1.6 Augmentation Results

Because the goal of this method is the applications for augmented reality, the augmentation results are illustrated in Figure 3.9.

Figure 3.9(a) or (b) is utilized for initialization. After the initialization, a camera can freely be moved thanks to the index update. For example a camera pose can be computed under strong tilt as illustrated in Figure 3.9(c), (d) and (e). Even though the half part of the document is overlapped by another object as illustrated in Figure 3.9(f), a camera pose is still computed from the visible part.

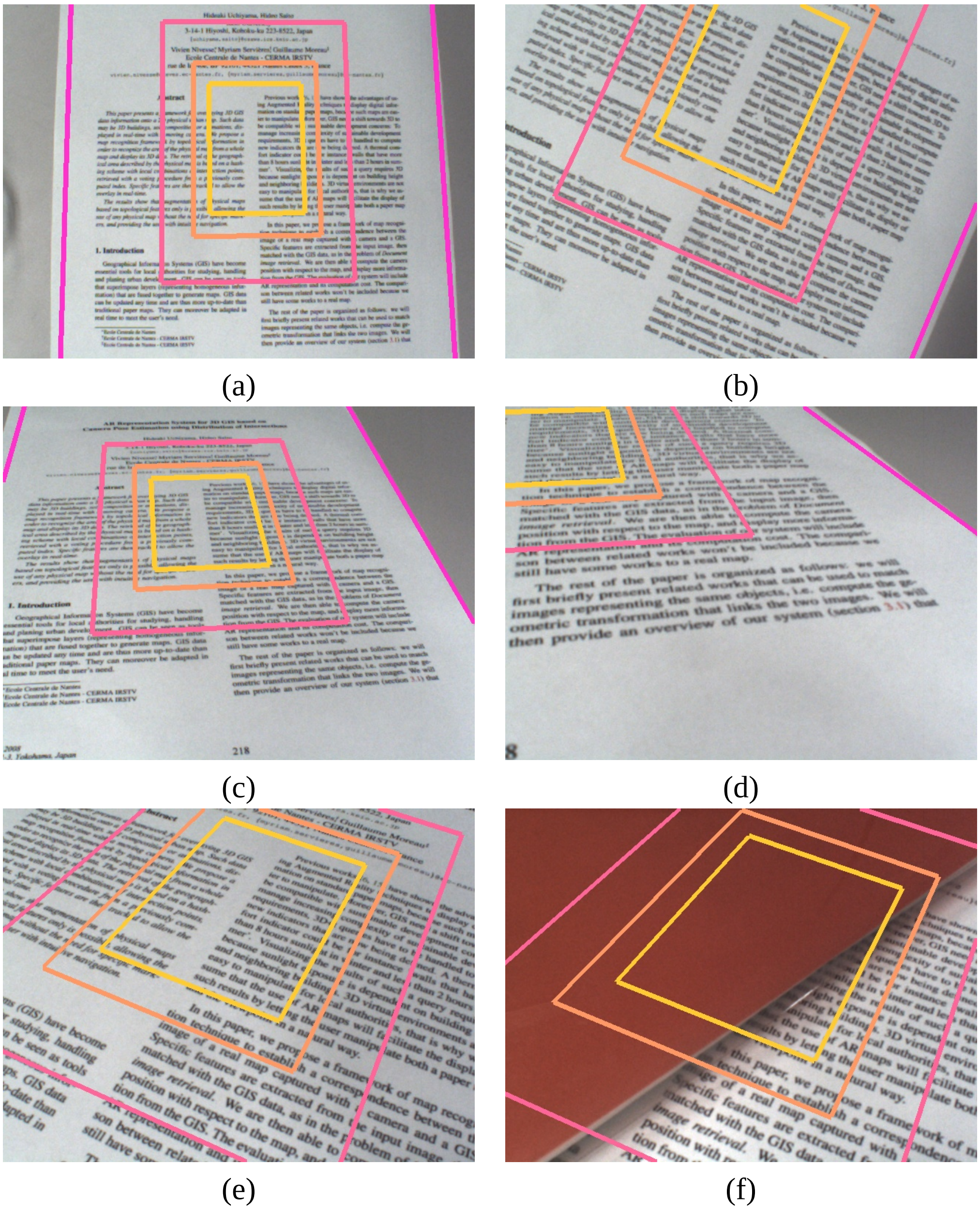


Figure 3.9: Augmentation results. (a) This top view is utilized for initialization. (b) A camera can be rotated. (c) Even though a camera is tilted, a camera pose can be computed. (d) A camera can be moved such that only part of the document is visible. (f) A camera pose can be computed in the presence of occlusion at the view of (e).

3.2 Temporal Feature Based Approach

In keypoint matching methods using texture or geometric features, a target object is implicitly visible. However, the object is not always visible due to a lighting condition such that nothing is visible at night in an outdoor case. For dealing with this case, a light is selected as a marker. Blinking pattern is utilized as a temporal feature.

3.2.1 Overview

A number of images need to be captured at each fixed view because the change of the brightness at each pixel should be received. This approach is similar with visible light communications [44, 77, 63, 10]. Traditional approaches utilized special devices to receive the blinking pattern. Compared with these approaches, this approach can be regarded as image based visible light communications for keypoint matching.

In order to extract the blinking pattern, the blinking speed and capturing speed of a camera should be same. For achieving this, the capturing speed should manually be measured, and set on the light as a pre-process.

From the next section, the design of the blinking pattern and the method for receiving data are described.

3.2.2 Blinking Rule

In order to extract lights and receive their data from a number of images, a signal processing technology is utilized. In other words, a light sends data by blinking. The packet format of the data based on signal processing is described in this section.

For the design of the format, the number of images captured at the same time should be considered. Suppose the number of images is 100. This means that a camera sends 100 samples. A sample is equivalent to 0 or 1 by on/off of a light. In order to receive the data without loss, 50 samples are utilized from 100 samples per packet.

Table 3.1: Representation of 1 bit. 1 bit is composed of 4 samples. Each bit is represented by 0011 and 1100. By using 4 samples, 1 sample error detection and correction are possible.

4 samples	1 bit
0011	0
1100	1

Table 3.2: Data transmission format. The number of bits per packet is 12 bits $(= 50 \div 4)$. The allocation of bits for data and check is designed depending on a purpose.

Component	Bit
Header	3
Data	6
Check	3

In order to represent 1 bit, 4 samples are utilized to detect and correct 1 sample error as described in Table 3.1. For example, a bit is 0 when 4 samples are 1011 because its nearest pattern 0011 is selected. From this representation of 1 bit, the number of bits per packet becomes 12 $(= 50 \div 4)$.

The format of header, data and check bits are defined in Table 3.2. For the header, the pattern should be different from patterns for bits. Because there are 3 bits for the header, the header samples are set as 111111000000. For check bits, cyclic redundancy checking (CRC) is utilized [93] by using following CRC polynomial:

$$x^3 + x + 1 \tag{3.1}$$

An example of the transmitted data is described in Table 3.3.

3.2.3 Data Reception

Ideally, the data of all pixels is analyzed to receive data in parallel. However, it is not sometimes possible because of the lack of memory space. For this reason, the candidates of lights are extracted from the first N images at first (N is discussed later). For only the candidates, the data of all pixels are analyzed.

Table 3.3: Example of transmitted samples. The first part of the transmitted samples is the header. The samples comverted from ID and CRC bits are inserted after the header.

ID	10
ID bits	001010
CRC bits	011
Total samples	111111000000001100111100001111000011001111001100

For each pixel, a threshold for judging the light as lighting or not is computed at each view. If the intensity is higher than a threshold, the pixel is lighting. This way is similar with related works [77, 63]. However, a threshold was fixed beforehand in these works. A fixed threshold is not suitable because a threshold depends on lighting condition. For this reason, a threshold of each pixel is computed from images.

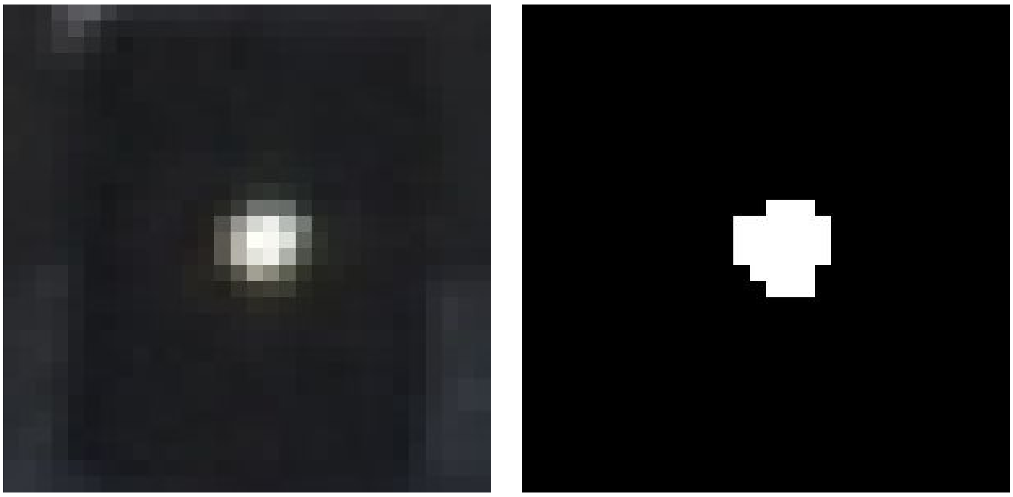
For each pixel in the first N images, N intensities are extracted. A threshold Th_i of each pixel i is computed from the N intensities as follows:

$$Th_i = \frac{Max_i - Min_i}{2} + Min_i \tag{3.2}$$

where Max_i is a maximum intensity and Min_i is a minimum intensity.

Next, the candidates of the lights are extracted from the first N images. For each pixel, blinking pattern is extracted by using Th_i. The pattern is represented by the sequence of 0 and 1. If the pixel corresponds to a light, regular pattern appears. The regular pattern means that switching points of 1 and 0 (01 and 10) appear at even interval because a transmitted data is composed of a header (111111000000) and a bit (0011 or 1100). If the blinking pattern of a pixel follows the regular pattern, the pixel is judged as the candidate of lights.

After the candidates are extracted, the whole blinking pattern of each candidate is extracted from 100 images. In the sequence of 0 and 1, a header is searched to find the beginning of a packet by using template matching at first. After a header is found, 36 samples (= 9 bits$\times$ 4 samples) are converted into 9 bits by 0011 and 1100. For 9 bits, CRC is performed to check a light or not. In addition, each pixel can decode the transmitted data. Finally, only light pixels remain by removing



(a) (b)

Figure 3.10: Light extraction. (a) The smaller transmitter is captured. The rim area of the light is blurred. (b) A light region is extracted by clustering the same ID.

non-lights.

For making light regions as illustrated in Figure 3.10, the same data of the pixels is clustered. For each region, the center is computed as a light position in the image.

3.2.4 Keypoint Matching

The procedure of keypoint matching based on this approach is as follows. First, the lights are set on the measuring positions of a target objects. Next, a number of images are captured at fixed viewpoints. For each view, the images are analyzed to extract the light and blinking pattern. By selecting the same blinking pattern, the same lights in different view images are matched.

3.2.5 Experiments

In the experiment, Nikon D300 is selected as a camera because this camera is actually utilized for the applications described in Chapter 7. The camera parameters of Nikon D300 are described in Table 3.4.

Table 3.4: Camera parameters. These parameters are fixed during the experiments.

Num of Images	100
Focal Length	50 (mm)
F	F/16
ISO	ISO-500
Exposure time	1/100 (sec)

Stability of Light Extraction

First, the stability of light extraction is evaluated.

20 lights are set and captured from 8 views. At each view, the extraction of lights was performed 3 times. This means that a total number of lights is 480 $(= 20 \times 8 \times 3)$.

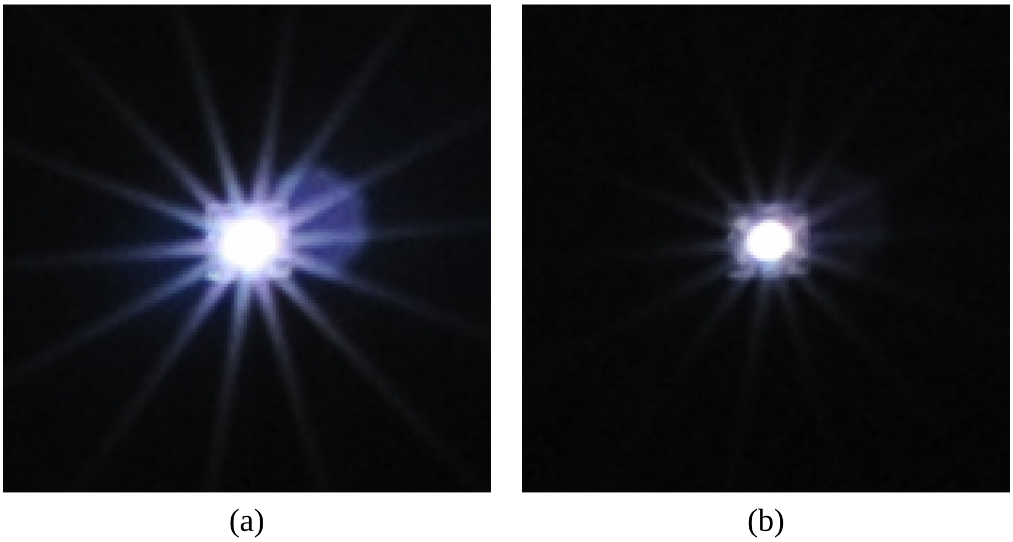
From 480 lights, 459 lights are correctly extracted. No light region was wrongly judged as a light. The reason of failure cases is the incomplete calibration of capturing and blinking speeds. In Table 3.5, the example of the transmitted and received blinking patterns is described. If the starting timing of capturing images is synchronized with that of blinking, the moment of turning on or off a light may be captured as illustrated in Figure 3.11. In this case, one sample may additionally be captured as described in Table 3.5. Error detection and correction by selection of the nearest set (0011 or 1100) do not work for this case.

Table 3.5: Sampling error. 1 sample is sometimes additionally captured when the starting timing of capturing images is synchronized with that of blinking. A light may be captured at the moment of turning on or off.

Transmitted	:	1100110011
Received	:	11001110011

Tolerance for Distance

Next, the tolerance of light extraction against the distance between a light and a camera is evaluated. 10 light are set at 2 meters, 35 meters and 50 meters far



(a) (b)

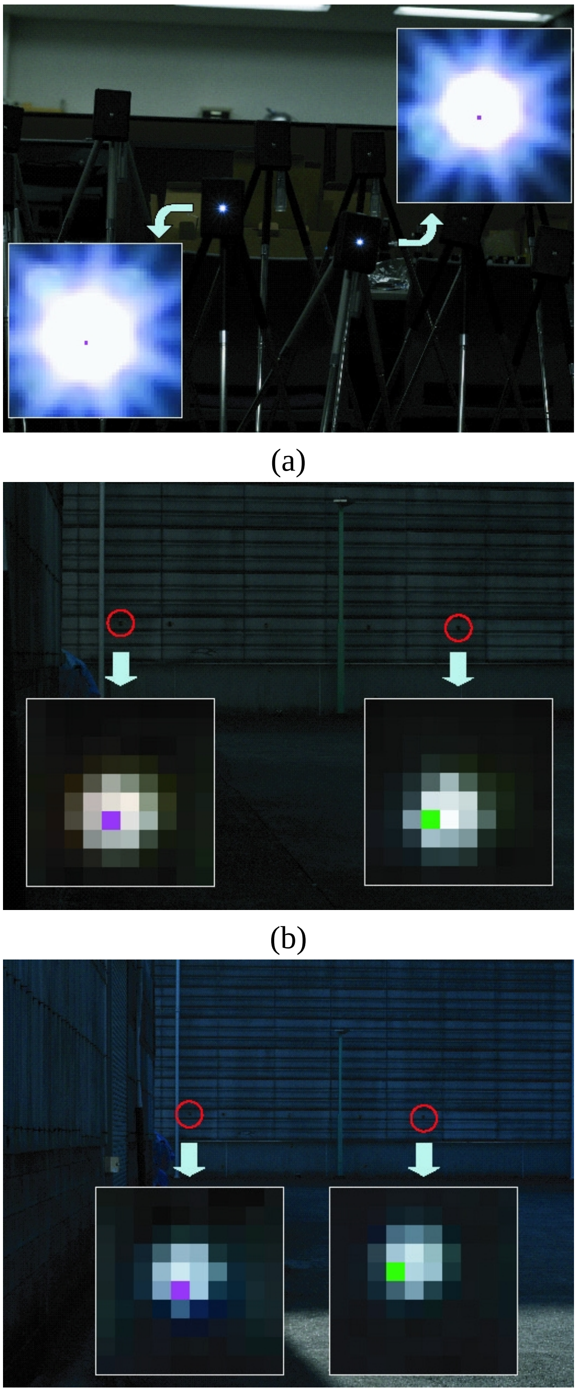
Figure 3.11: Moment of turning off. (a) This image is captured when a light is perfectly on. (b) Compared with (a), the rim area of a light is smaller because this is captured at the moment of turning off.

from a camera as illustrated in Figure 3.12. At each distance, light extraction is performed 3 times. This means that 30 lights are extracted in total at each distance.

In Figure 3.12, the examples of captured lights are illustrated. The image resolution of the lights is 150×150 pixels in (a) and 10×10 pixels in (b) and (c). Table 3.6 describes the relationship between the size of a light in an image and the distance between a camera and a light. If the brightness of a light is enough for capturing a light, the light can be extracted even when the light is set close to or far from a camera.

Table 3.6: Size of captured lights. Even when a camera is 50 meters far from a light, the light can be captured to receive data.

Distance (m)	Pixel
2	2963.6
35	7.4
50	6.5



(a)

(b)

(c)

Figure 3.12: Tolerance for distance. The light is set at (a) 2m (b) 35m (c) 50m for judging whether the transmitted data of a light can be received or not.

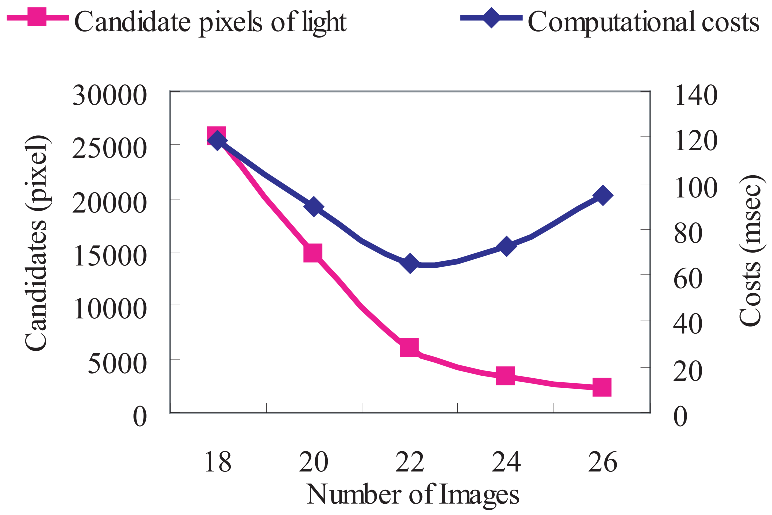


Figure 3.13: N vs. computational costs. The computational costs include light candidate extraction for N images and light extraction from the candidates for $100 - N$ images.

3.2.6 Discussion

In the extraction of the candidates of lights, there is a parameter N, which influences the computational costs. The costs are composed of the extraction of the candidates of lights for N images and the extraction of lights from the candidates for $100 - N$ images. In Figure 3.13, the relationship between N and computational costs is illustrated. If N is larger, the number of light candidates decreases because a non-light pixel does not follow the regular pattern longer. However, the extraction of the candidates of the lights needs more costs if N is larger. For this reason, N should be optimized by the pre-processing.

3.3 Conclusion

Toward texture-free keypoint matching, geometric feature based keypoint tracking and temporal feature based keypoint matching are developed.

In geometric feature based keypoint tracking, on-line learning process of geometric feature is developed. Because the geometric feature is not invariant to the large range of viewpoint, geometric feature based keypoint matching is not enough for relaxing camera movement. In the learning process, the geometric features (descriptors) of each matched keypoint are re-computed to update them. The updated features can be utilized from the next frame. This tracking can be called tracking by descriptor update. Because this can be regarded as a general framework for on-line learning, this approach has a possibility to be applied to other methods.

In temporal feature based keypoint matching, image based visible light communication is developed. Because a light emits by itself, it can be a useful marker in the dark. By blinking, the change of the brightness becomes the temporal feature. For the design of the blinking pattern, a signal processing technology is utilized. The extraction of the light and the reception of the data are based on the technology. This marker system will be helpful for the measurement of outdoor constructions.

Chapter 4

Pool Supporting System

4.1 Introduction

A pool game is one of the popular sports in the world. People of all ages can enjoy several types of pool games such as pocket billiard, carom billiard and snooker. As a characteristic of pool games, players require much skill. They should determine the direction and strength to pocket balls by considering laws of physics such as collision and reflection. Also, they should follow the rule of the games. If a supporting system for pool games is developed, it will be useful for assisting pool players to get skillful.

The studies of pool games can be divided into two categories; the analysis of best shooting way and the display of supporting information.

For computing the best shooting way, several approaches based on fuzzy analysis were proposed. Alian et al. [2] proposed a cost function composed of distances and angles to compute the strength and the direction of the shot. Cheng et al. [13] extended a cost function by including predictable hitting error. Chua et al. [15] used another fuzzy approach called Sugeno Fuzzy. Instead of these fuzzy approaches, Lin et al. [53] used grey decision-making for dealing with uncertainty and incompleteness. For precise analysis, parameters of an actual pool table need to be measured [62].

In the study of the display of supporting information, Jebara et al. [38] proposed a concept of an AR based wearable computer system with a head mounted live video display. A camera-projector system is also utilized as an AR based display [48, 99].

In the previous supporting system, special equipment was necessary and difficult to install in the normal environment. Toward practical use, an AR based pool supporting system using a handheld device is developed. A user has only a camera mounted handheld display to capture images and watch supporting information. When a user captures a pool table, the system automatically estimates ball arrangement on the table and compute supporting information. Then, the user can watch supporting information virtually overlaid on the table by AR while the user captures the table.

In the process of the camera pose estimation, the correspondences of table corners between in the image and the 3D table model need to be established. For this, the geometrical relationship of table corners is utilized.

4.2 System Overview

The usage and Graphical User Interface (GUI) of the system are illustrated in Figure 4.1. First, a user stands beside a pool table, and captures the whole part of the table from an arbitrary view. Then, the system estimates ball arrangement on the table, and analyzes them. After the analysis is finished, several candidates of a next shooting way are provided on the interface. When the user selects one of the candidates, the detailed shooting way is displayed at the upper part of the interface. In addition, the user can watch the shooting way virtually overlaid on the table by AR through the interface when the user captures the whole part of the table again.

4.2.1 Processing Flow

The algorithm is divided into three parts.

First, ball arrangement is estimated from one captured image because the size of a table and balls and their geometrical relationship are known. From a captured image, a green table region is extracted by using thresholding based color segmentation. Then, ball regions are extracted inside the green table by removing a

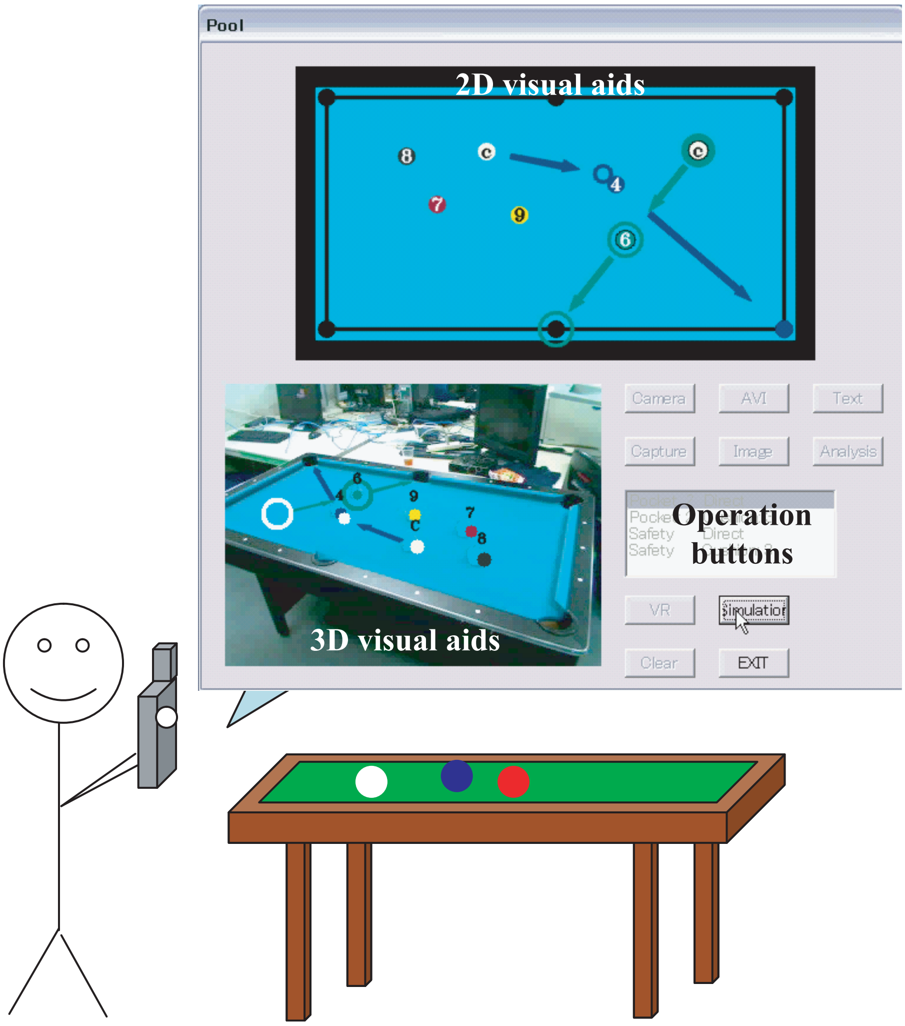


Figure 4.1: System overview. The equipment for a user is only a camera mounted handheld display. The user captures a whole part of the table to compute ball arrangement and watch AR supporting information on the interface.

green color. A ball number for each region is recognized by matching all pixels of the region with the colors of all balls. For computing ball positions on the table, a camera pose with respect to the table is computed by using the planarity of the table. Finally, each ball position is computed from its image coordinate and the camera pose.

Second, supporting information for a next shooting way is computed. By assuming that ball behavior follows particle dynamics, the possibilities of the next shooting way are limited and linearly computed. For each possibility, ball behavior is simulated and evaluated to provide some desirable shooting ways.

Third, supporting information of next shooting ways is displayed by AR. When a user selects one of the shooting ways, the system provides the shooting direction and ball behavior. The supporting information is overlaid on captured images while the user captures the whole part of the table.

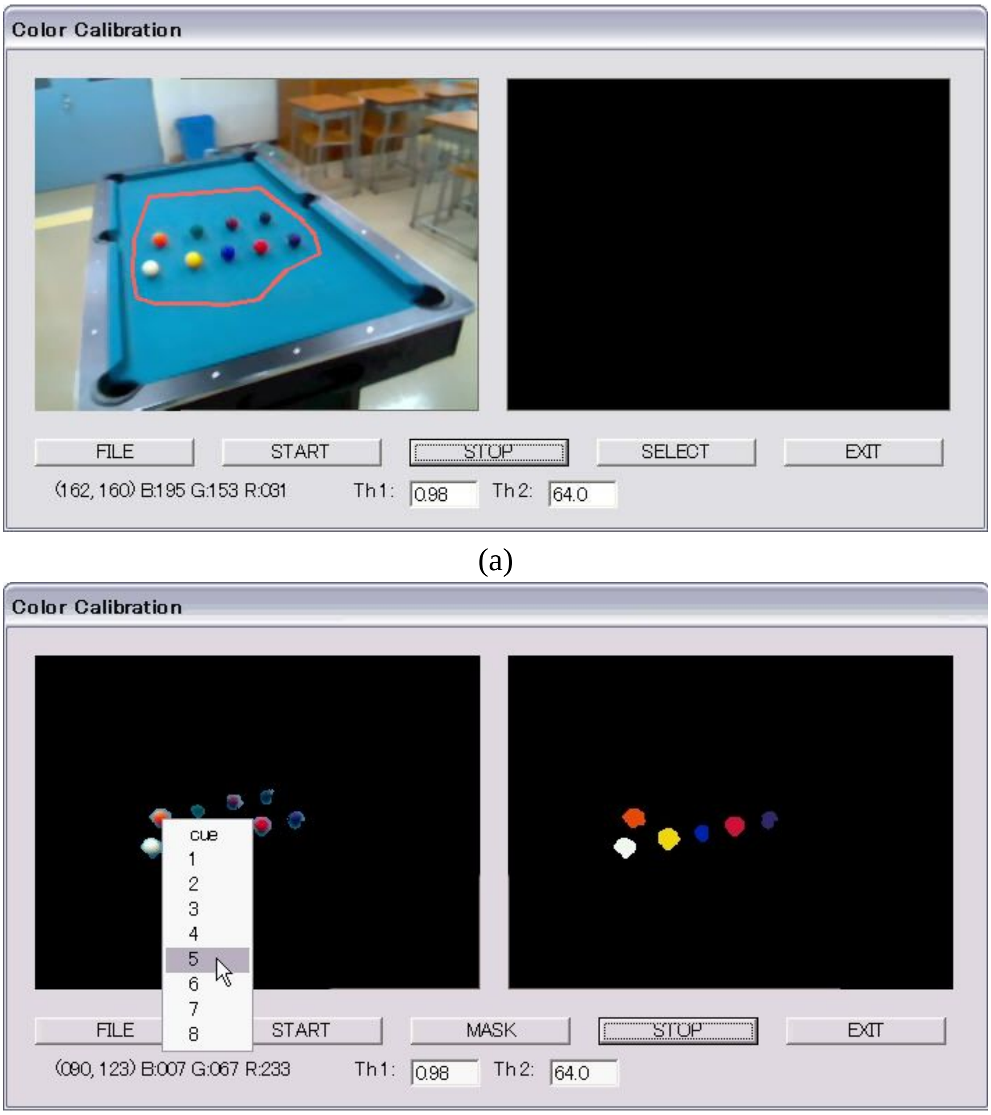
4.2.2 Preparation

The sizes and colors of a table and balls are measured beforehand. The GUI for measuring colors is developed as illustrated in Figure 4.2.

First, a user segments a region including the balls inside the green table as illustrated in Figure 4.2(a). The system computes the color of the table by making RGB histogram of the segmented region and computing a peak of the histogram. Then, the system automatically extracts ball regions by removing the green region, assuming that ball colors are different from the table color. Next, the user specifies a ball number at each region by clicking each ball region as illustrated in Figure 4.2(b).

4.3 Ball Arrangement Estimation

The process for ball arrangement estimation is divided into five parts; corner extraction, ball extraction, ball identification, camera pose estimation and ball position estimation.



(162, 160) B:195 G:153 R:031 Th1: 0.98 Th2: 64.0

(a)

(090, 123) B:007 G:067 R:233 Th1: 0.98 Th2: 64.0

(b)

Figure 4.2: Measuring colors. (a) A user draws a region including all balls inside the table. (b) From users' input, the ball regions are automatically extracted. Then, users specify a ball number by clicking each region.

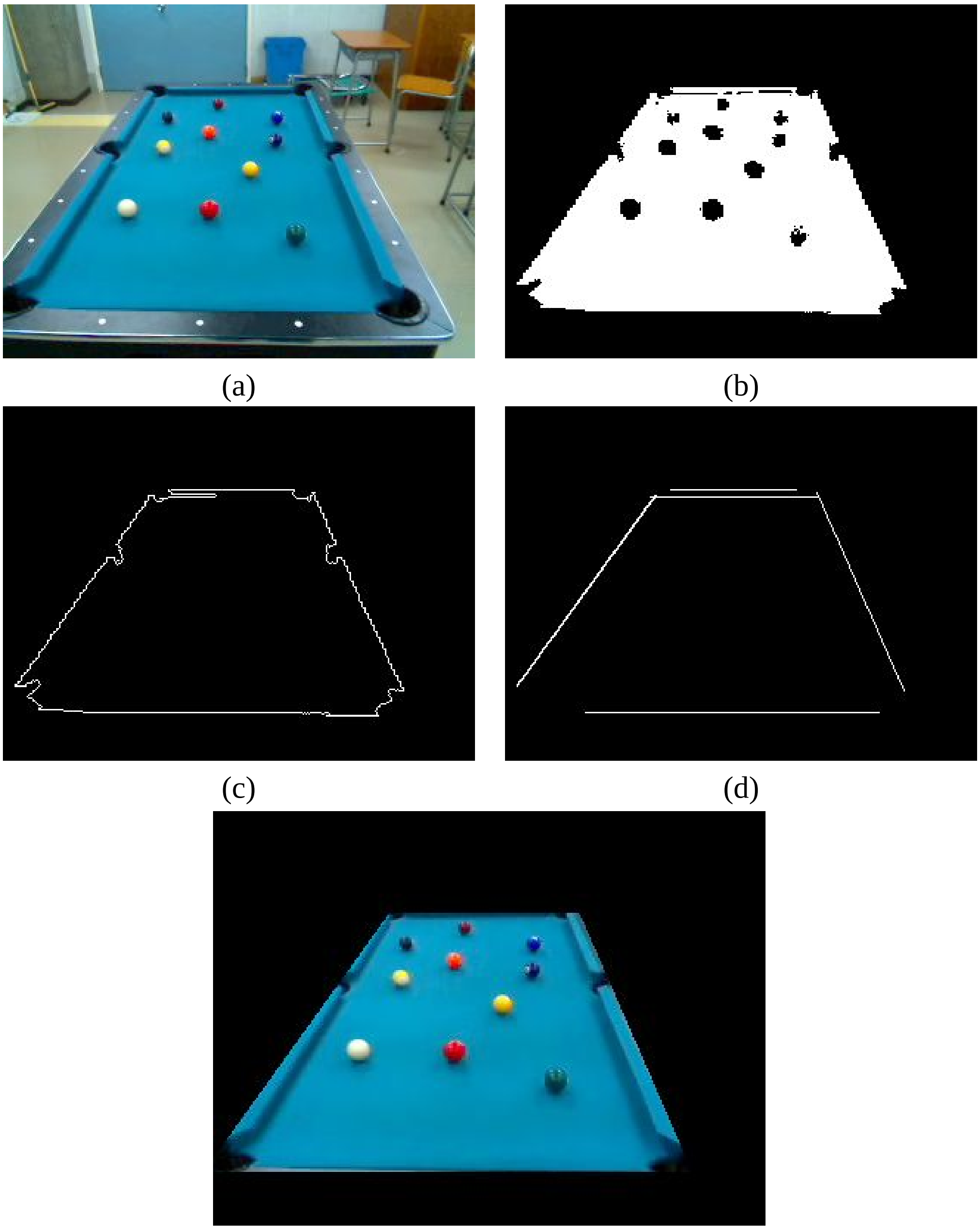
4.3.1 Table Corner Extraction

In order to compute a camera pose with respect to a pool table, 2D-3D correspondences are necessary. In this system, table corners are selected for the correspondences. An example of input images is illustrated in Figure 4.3(a). A user should capture a whole part of the table for extracting four lines of the green mat. In order to extract the green mat, each pixel is compared with the color of the green mat by following equation:

$$Angle = \arccos\left(\frac{\mathbf{C_g} \cdot \mathbf{C_g}}{|\mathbf{C_g}||\mathbf{C_p}|}\right) \tag{4.1}$$

where $\mathbf{C_p}$ is RGB at each pixel, $\mathbf{C_g}$ is RGB of the green mat and $Angle$ is an angle by two colors as a similarity score. The angle is small if two colors are similar. By using a threshold as 10 degrees, each pixel is classified. An extracted green mat is illustrated in Figure 4.3(b).

Next, the contour of the green mat is extracted for computing four line segmentsn as illustrated in Figure 4.3(c). By using Hough transform [61], line segments are extracted from the contour as illustrated in Figure 4.3(d). If there are more than four lines, they are clustered into four line segments. After the segments are extracted, corners are computed by finding intersections. Figure 4.3(e) illustrates the green mat region extracted by four corners.



(a)

(b)

(c)

(d)

(e)

Figure 4.3: Table corner extraction. (a) A whole part of the table is captured. (b) A table mask is generated by finding a green color in (a). (c) The contour of the mask is extracted from (b). (d) Line segments are extracted by Hough transform from (c). (e) By computing four intersections from (d), a green mat area is extracted.

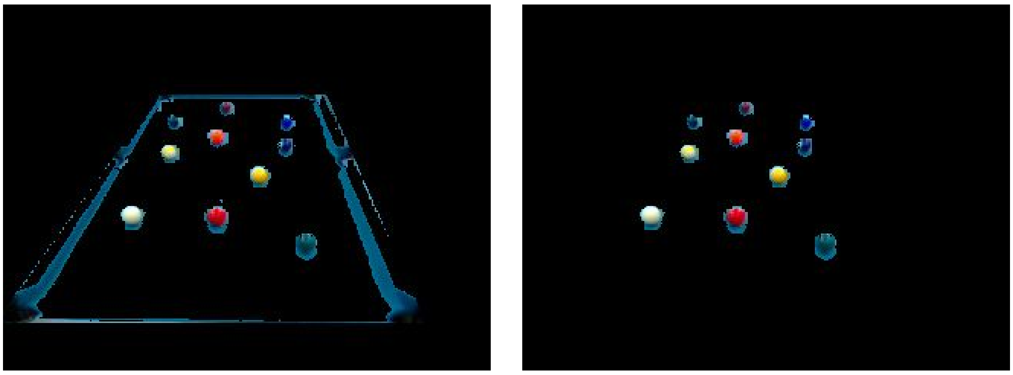


Figure 4.4: Ball extraction. (a) The candidates of balls are extracted from the table area by removing the green mat area. (b) By thresholding the size of the area, balls are extracted.

4.3.2 Ball Extraction

Inside the green mat, balls are extracted because balls always exist there. First, ball candidates are extracted by removing a green color by the following equation:

$$E = a\frac{\mathbf{C_g} \cdot \mathbf{C_p}}{|\mathbf{C_g}||\mathbf{C_p}|} + b\left(1 - \frac{|\mathbf{C_g} - \mathbf{C_p}|}{255}\right) \tag{4.2}$$

where $\mathbf{C_p}$ is RGB at each pixel, $\mathbf{C_g}$ is RGB of the green mat, E is a similarity score and a and b are weights. The first term is the same as Equation 4.1. The second term computes the difference of two intensities. From experiments, the weight of each term is determined as $a = 0.7$ and $b = 0.3$. By setting the thresholding of E as 1.05, the ball candidates are extracted as illustrated in Figure 4.4(a).

In the candidates, pocket areas and shadow cushions are included because these colors are different from the green mat color. These regions are removed by using size thresholding. The ball size in an image is determined from 50 pixels to 400 pixels in 320×240 pixels' image. Figure 4.4 (b) illustrates the result of ball extraction.

4.3.3 Ball Identification

Because the system focuses on the 9-ball game, which uses balls from No.1 to No.9 and a white cue ball, identification of these balls is performed.

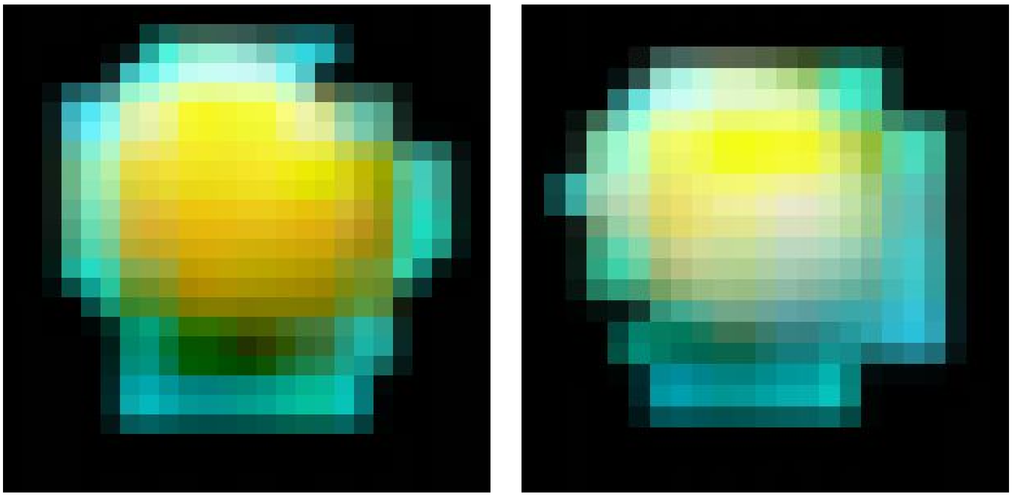


Figure 4.5: Extracted balls. (a) No.1 ball is extracted. (b) No.9 ball with a white stripe is extracted. Compared to (a), more white pixels are included.

The examples of an extracted ball region are illustrated in Figure 4.5. At each pixel, RGB is compared with RGBs of all balls by Equation 4.1. By selecting the nearest color, each pixel is voted into one of balls as described in Table 4.1,

From this voting result, ball identification is performed by the following equations:

$$\frac{b_c + b_1}{b_{sum}} \geq 0.5 \cap 0.7 \geq \frac{b_c}{b_{sum}} \geq 0.3 \tag{4.3}$$

where b_c is the voting for a white ball, b_1 is the voting for No.1 yellow ball and b_{sum} is the sum of voting. The first comparison checks whether the ball is one of three balls (cue, No.1 and No.9 ball) or not. If the ball is not one of them, the maximum count is selected as a ball number.

The second comparison is used for distinguishing three balls. If the ratio is over 0.7, the area is the cue ball. Also, if the ratio is less than 0.3, the area is No.1 ball. Otherwise, the area is No.9 ball. These thresholds are determined from experiments.

4.3.4 Camera Pose Estimation

For computing ball arrangement on the table, a camera pose with respect to the table is estimated.

Table 4.1: Ball identification. At each pixel in a region, RGB is compared with RGBs of all balls. From the result, a ball number is identified.

	c	1	2	3	4	5	6	7	8
Figure 4.5(a)	34	55	0	0	0	13	19	1	1
Figure 4.5(b)	67	34	0	0	0	2	11	0	8

The world coordinate system of the pool table is defined as illustrated in Figure 4.6(a). In the table, there are two sides; long and short as illustrated in Figure 4.6(b) and (c). For making 2D-3D correspondences of table corners, the viewing side should be determined.

If the table is captured from a short side, the difference of the lengths between the short side and the long side is small because of perspective projection. In contrast, the difference is large if the table is captured from a long side. This geometric relationship of the corners is helpful for the identification of the table side. By using thresholding the difference, the viewing side is estimated.

After the viewing side is determined, the correspondences of table corners are established to compute a camera pose by using Appendix.

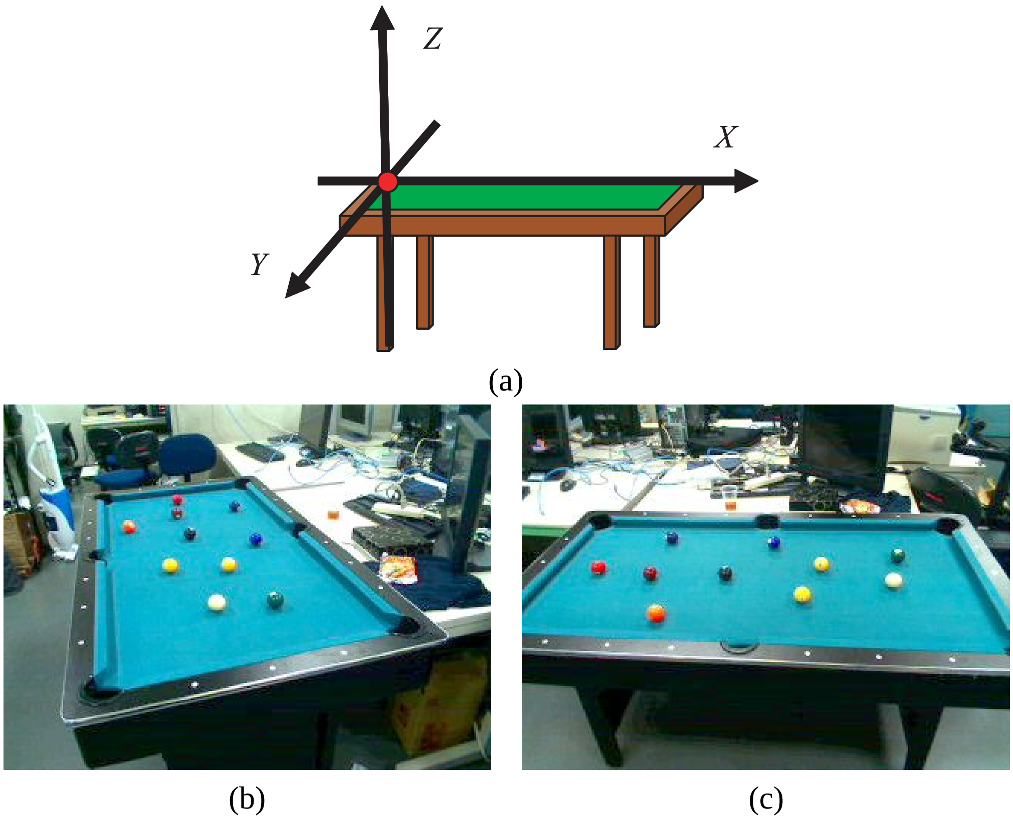
4.3.5 Ball Position Estimation

In the camera pose estimation, $\mathbf{P}$ in Equation A.2 is computed. In the equation, (X, Y) means a ball position on an actual pool table in 3D. If (u, v) and Z are known, (X, Y) is linearly computed.

For each ball, its (u, v) comes from the center of the ball region. Z can be determined beforehand because the sizes of a ball and table are known. In Figure 4.7, r is the radius of the ball and h is the height of the cushion. The center of the ball with respect to Z axis is $Z = -(h - r)$. By using each (u, v) and $Z = -(h - r)$, a ball position (X, Y) is estimated.

4.4 Supporting Information Computation

The system provides the candidates of a next shooting way in the 9-ball game as supporting information. In the rule of the 9 ball game, players should pocket balls



(a)

(b)

(c)

Figure 4.6: Camera pose estimation. (a) A world coordinate system is defined. The origin is one of the corners. (b) An image is captured from a short side. (c) An image is captured from a long side.

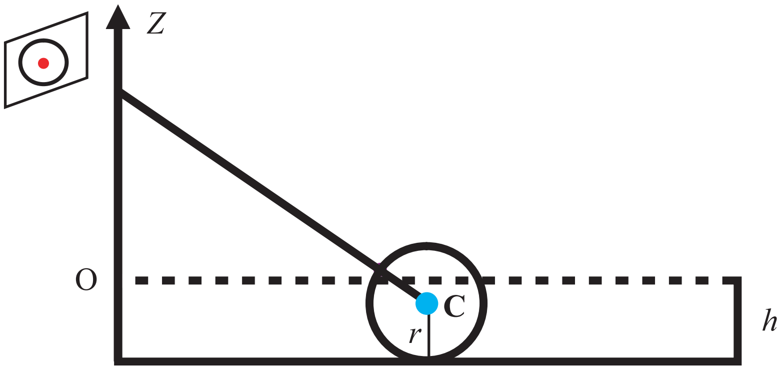


Figure 4.7: Ball position estimation. **C** is a ball position. h is height of a cushion. r is height of a ball. Because a ball is attached on the table, Z of the ball is determined beforehand from h and r.

in the order of ball's numbers. Also, players should switch if they cannot pocket any balls. By considering these rules, there are two types of strategies; pocketing balls (called Pocket) or blocking other players (called Safety).

Pocket

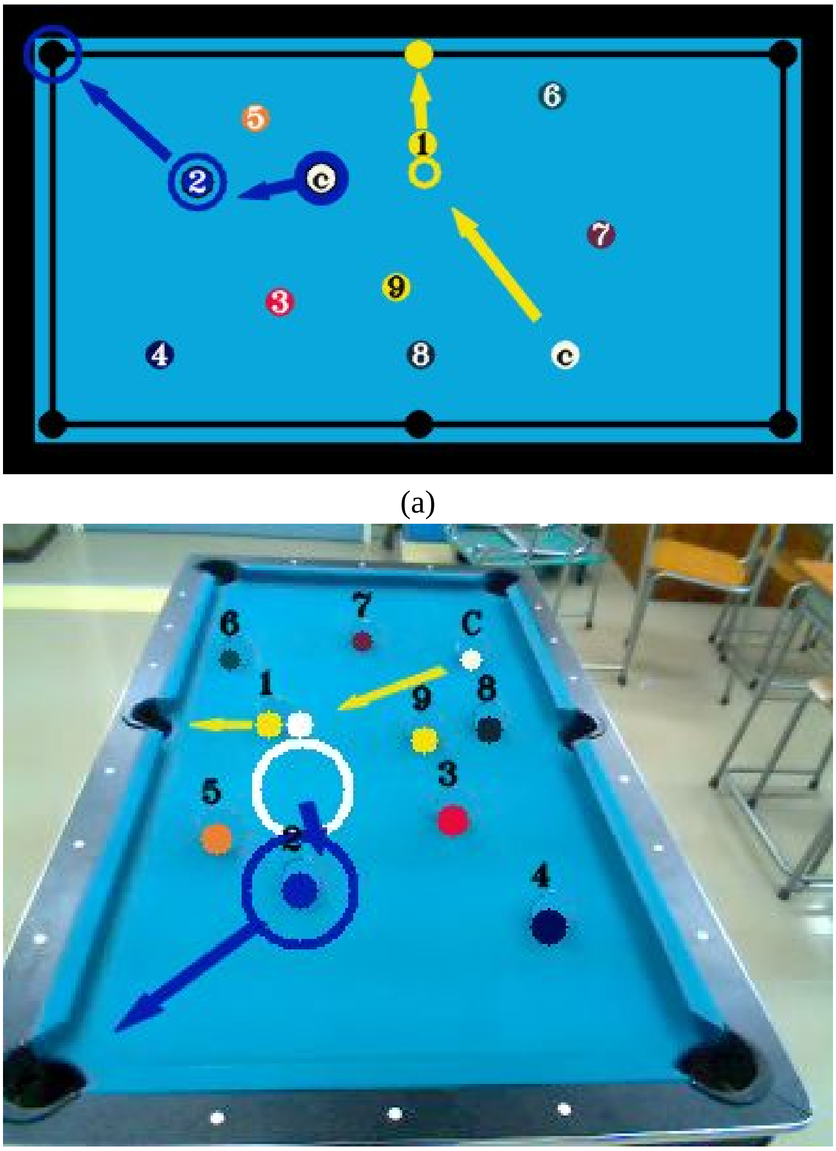
Players aim at pocketing a target ball by considering a position of a next target ball. For computing supporting information for Pocket, ball paths to pocket a target ball are linearly computed because the system assumes that ball behavior follows particle dynamics with friction on the table and reflectance at the cushion. For each path, the system simulates the ball behavior by giving some speeds. If a next shooting path is found after the simulation, the ball path is provided for users.

Safety

If Pocket is not possible, players shoot the cue ball in order to block next players. In this case, several ball paths exist because what players should do is only shooting the cue ball against the target ball. For each possible path, ball behavior is simulated by the same way as Pocket. If a next shooting path is not found after the simulation, this ball path is provided for users. This means that next player will fail in the arrangement.

4.5 Supporting Information Display

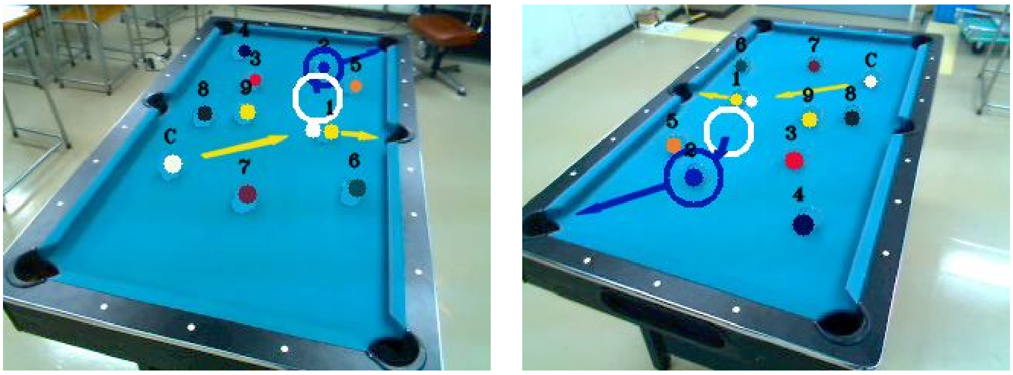
The system provides two types of supporting information displays; 2D style and AR style. The 2D style is equivalent to a top view as illustrated in Figure 4.8(a). In the AR style, a user can watch the supporting information by AR while the user is capturing the table as illustrated in Figure 4.8(b).



(a)

(b)

Figure 4.8: Supporting information display. (a) The 2D style is equivalent to a virtual top view. (b) The user can watch supporting information drawn on the captured image by AR.



(a) (b)

Figure 4.9: Online AR display. (b) is captured from the opposite side of (a). In order to achieve AR for arbitrary views, a camera pose with respect to the table is appropriately computed. In this case, two short sides are identified.

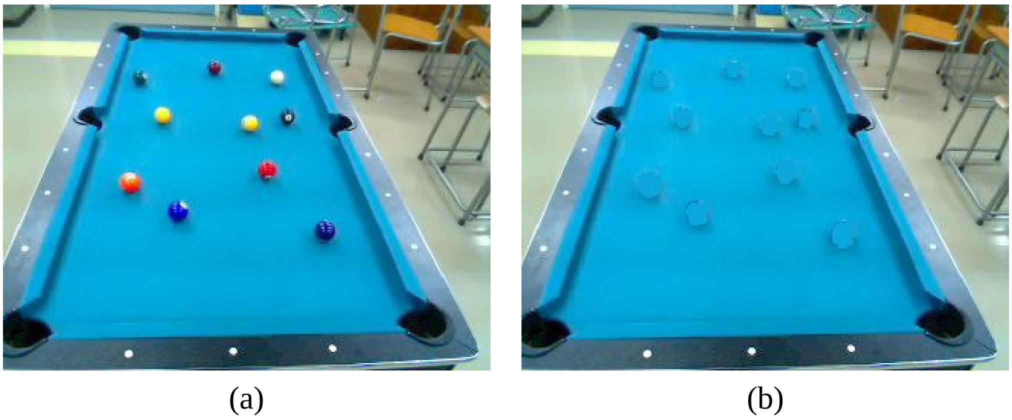
In the AR style, a camera pose should be computed as well as ball arrangement estimation. In order to achieve AR for arbitrary views, 2D-3D correspondences for corners are appropriately established. As illustrated in Figure 4.9, the two short sides should be distinguished.

In this process, the green mat region, ball regions and corners are extracted from a captured image by the same way those of ball position estimation. In order to find the 2D-3D correspondences of corners, the origin of the world coordinate system is firstly selected from four corners. Because the number of corners is four, there are four possibilities for the origin. For each possibility, ball arrangement estimation is tested. If computed ball arrangement is the same as that of ball position estimation, the origin is correctly selected.

For overlaying supporting information, a pool table image without balls should be prepared as a background. A background image is generated from a captured image by replacing ball regions with the green mat color as illustrated in Figure.4.10.

4.6 Experiments

In this section, the accuracy of ball position estimation and reprojection error are evaluated. The size of the pool table is 1330 mm × 700 mm. The radius of a ball is



(a) (b)

Figure 4.10: Background generation. (a) For a captured image, ball regions are extracted. (b) The ball regions are interpolated by green mat color.

24 mm. They are smaller than the official sizes. An Ultra Mobile PC (UMPC) as a handheld device has 1.0GHz CPU, 512MB RAM and 320×240 pixels' camera.

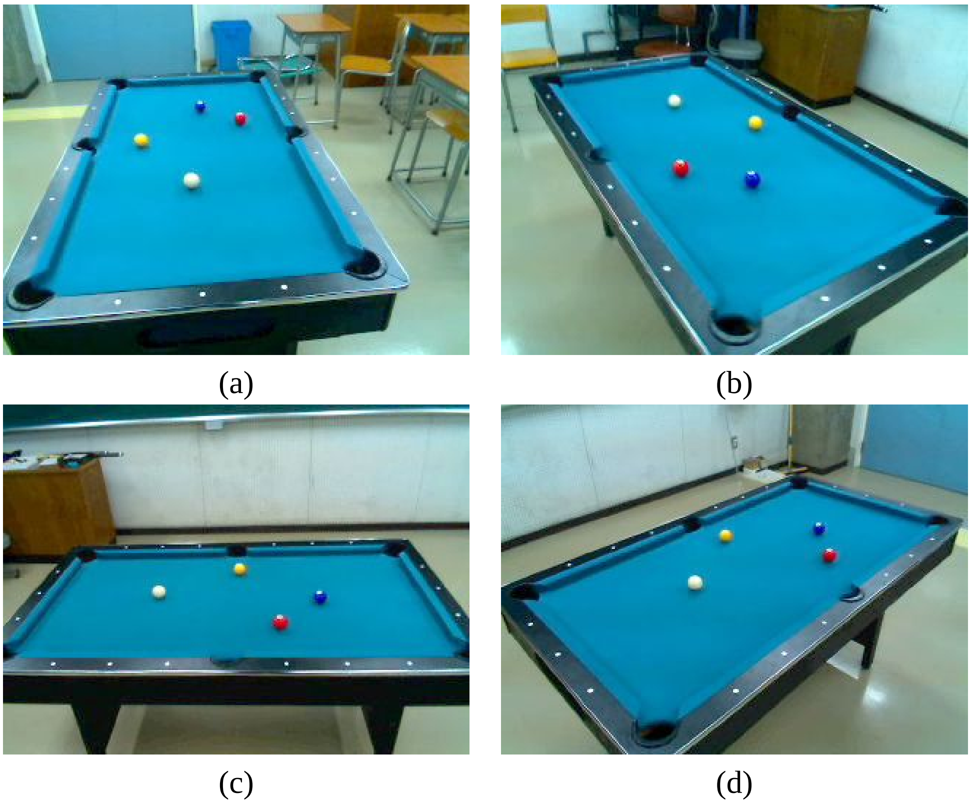
4.6.1 Ball Position

Ball positions measured by the system and ground truths measured manually by a measuring tape are compared. Figure 4.11 illustrates different view images for the experiment.

For each view, ball positions are estimated as described in Table 4.2. The maximum error is 16 mm, and the average of errors is 10 mm. The accuracy depends on the segmentation result of the table and balls. Because simple color segmentation by thresholding is utilized for fast computation, ball regions are not finely extracted. As illustrated in Figure 4.5, a shadow green mat region is also extracted. The center of a ball may not be correctly computed.

4.6.2 Reprojection Error

For evaluating the accuracy of AR display, reprojection error of the ball center is computed. Computed ball positions in Figure 4.11(b) are reprojected onto the captured image by using the camera pose. As a ground truth, each ball center is manually clicked. The distances between the reprojected positions and centers clicked manually are within 5 pixels on an average. The result depends on the



(a) (b) (c) (d)

Figure 4.11: Images for evaluation. Estimated ball arrangement estimation is evaluated. Four different views are evaluated as (a), (b), (c) and (d). The result of each view is in Table 4.2.

accuracy of estimated camera pose and ball arrangement. However, users said that supporting information was finely overlaid on the image and did not care the errors. Because the frame rate of AR Display is 10 fps, the users can watch natural ball behavior drawn on the captured images.

4.7 Conclusion

A handheld AR for supporting pool games is developed. Once a user captures the pool table, the user can watch AR visual aids of a shooting way suggested by the system. The system estimates ball arrangement on the table from one image, and then computes desirable next shooting ways by computer simulation. Recognition of the table and balls is based on color information. The system is implemented

Table 4.2: Error of ball arrangement estimation. The unit is mm. For each ball, the center of the world coordinate is computed. Because Z is fixed, X and Y are evaluated.

Number	c		1	
Coordinate	X	Y	X	Y
Ground Truth	340	404	172	685
(a)	340	411	183	691
(b)	338	408	173	692
(c)	333	398	159	681
(d)	352	394	182	678

Number	2		3	
Coordinate	X	Y	X	Y
Ground Truth	389	956	524	815
(a)	390	954	531	811
(b)	399	948	531	806
(c)	385	954	533	817
(d)	405	959	531	824

on UMPC, and works in real-time.

In the future works, the computation of supporting information is improved. For example, the rotation component is not considered in the simulation. In order to provide supporting information for pool trick shots, the simulation should be more precise. In addition, the segmentation of overlapped balls is necessary. If two balls are close and extracted as one ball, two balls cannot be recognized in the system. For solving this problem, a real-time segmentation method should be included.

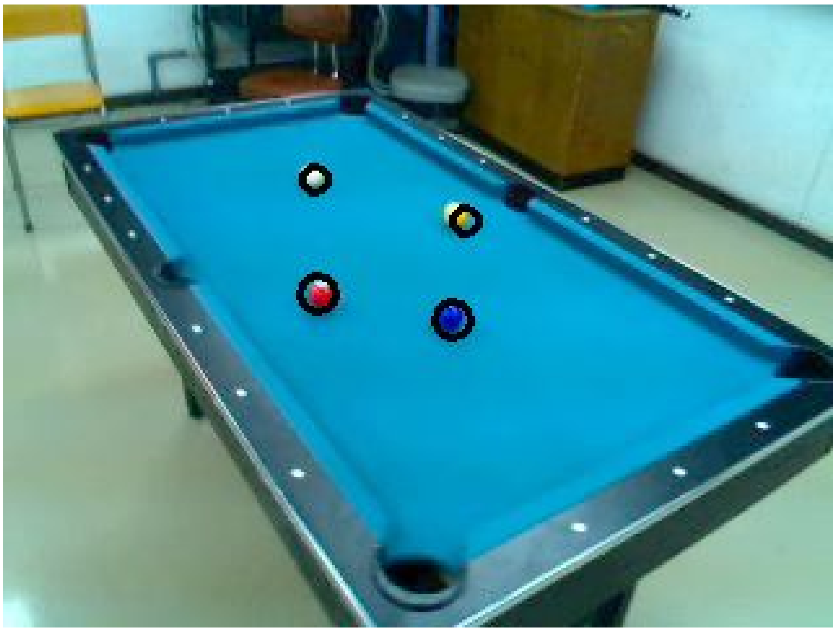


Figure 4.12: Reprojection error. For each ball, the center of a computed ball position in the world coordinate system is reprojected onto the captured image. The distance between the center in the image and the reprojected center is evaluated.

Chapter 5

Augmented Documents

5.1 Introduction

A paper document is getting a lot of attention for practical uses of AR applications. Even though the change of a world trend is going toward paper-less society, there is a great potential to be a novel tangible interface by enhancing its functionality of the document.

For example, Hull et al. developed an augmented reality application for newspapers. In this application, a user has a camera mounted mobile phone, and reads paper documents through the phone. Then, the user can click printed words and logos of documents on the display to connect the web or watch additional information by AR. This application brings document papers to have a novel connection between physical and digital world.

In the community of document analysis, there is a similar research topic called document image retrieval [20]. The purpose of this research is to access the additional information of the printed documents by capturing the documents. By indexing the printed documents, the access to the digital information from the captured document will be possible.

As a novel document AR system, on-line document registering and retrieving system for AR annotation overlay is developed. The system is designed for the people who do not want to write annotations on the documents directly. This can also be considered as an electronical bookmarking. Compared with traditional document AR systems, the system will have the possibility of widespreading because the users can use documents with their hand.

5.2 System Usage

The configuration of the system is only a camera mounted handheld display. A user prepares European text documents in which the user wants to write some annotations electronically. No other special equipment such as markers and sensors is necessary.

5.2.1 Registration

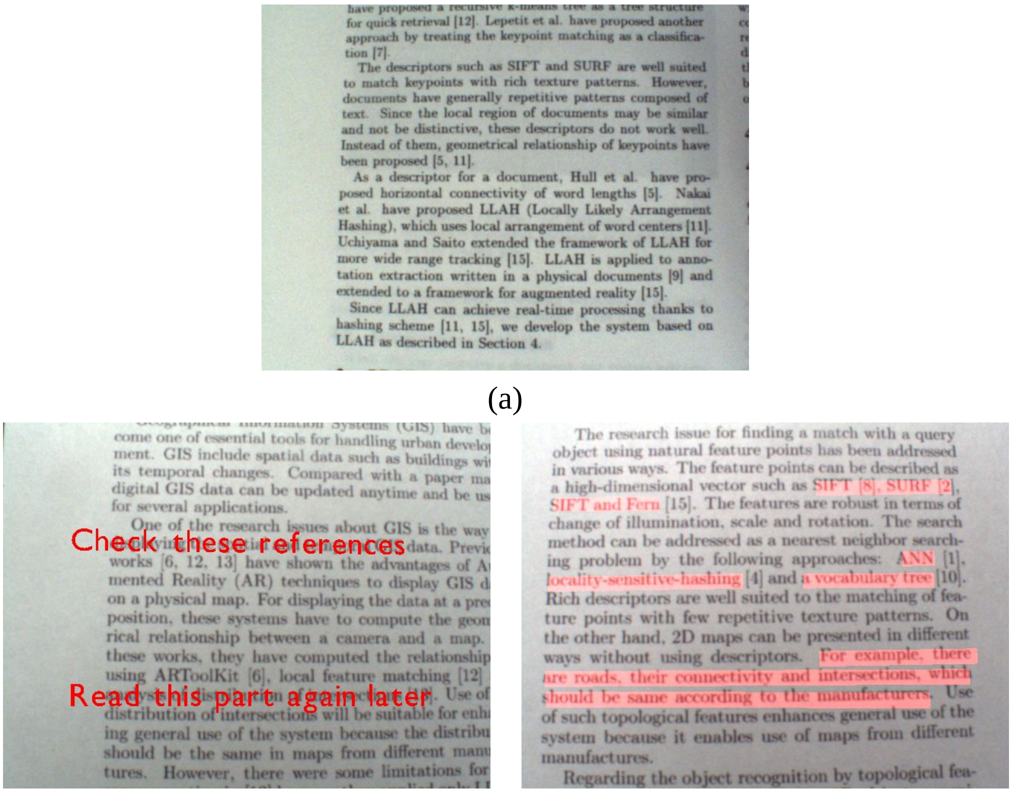
In the registration, the user captures one of the documents from a nearly top view captured as illustrated in Figure 5.1(a). In the captured image, the user can register any annotation on the document. This system provides two types of annotations: text as illustrated in Figure 5.1(b) or highlighter as illustrated in Figure 5.1(c). In addition to these annotations, pictures, movies and URL can be registered. The user virtually writes annotations on a captured document through the display.

The main characteristic of this system is to make a connection between user's physical documents and digital information. In order to avoid registering the same document, the user cannot register a document while the document is already retrieved.

5.2.2 Retrieval

When the registered document is captured again, the annotations are overlaid at the written position. While rotating and translating a camera, the users can watch the overlaid annotations as written on the document. Because a number of documents can be registered in the system, the system can identify each document and overlay its corresponding annotations.

When the multiple registered documents are captured at the same time, the annotations of all documents are overlaid as illustrated in Figure 5.2. In multiple



have proposed a recursive k-means tree as a tree structure for quick retrieval [12]. Lepetit et al. have proposed another approach by treating the keypoint matching as a classification [7].

The descriptors such as SIFT and SURF are well suited to match keypoints with rich texture patterns. However, documents have generally repetitive patterns composed of text. Since the local region of documents may be similar and not be distinctive, these descriptors do not work well. Instead of them, geometrical relationship of keypoints have been proposed [5, 11].

As a descriptor for a document, Hull et al. have proposed horizontal connectivity of word lengths [5]. Nakai et al. have proposed LLAH (Locally Likely Arrangement Hashing), which uses local arrangement of word centers [11]. Uchiyama and Saito extended the framework of LLAH for more wide range tracking [15]. LLAH is applied to annotation extraction written in a physical documents [9] and extended to a framework for augmented reality [15].

Since LLAH can achieve real-time processing thanks to hashing scheme [11, 15], we develop the system based on LLAH as described in Section 4.

(a)

Geographical Information Systems (GIS) have be
come one of essential tools for handling urban develop
ment. GIS include spatial data such as buildings wi
its temporal changes. Compared with a paper ma
digital GIS data can be updated anytime and be us
for several applications.
One of the research issues about GIS is the way
[illegible] data. Previ
works [6, 12, 13] have shown the advantages of A
mented Reality (AR) techniques to display GIS d
on a physical map. For displaying the data at a pre
position, these systems have to compute the geom
rical relationship between a camera and a map.
these works, they have computed the relationship
using ARToolKit [6], local feature matching [12]
[illegible]. Use of
distribution of intersections will be suitable for enh
ing general use of the system because the distribu
should be the same in maps from different manu
tures. However, there were some limitations for

(b)

The research issue for finding a match with a query object using natural feature points has been addressed in various ways. The feature points can be described as a high-dimensional vector such as SIFT [8], SURF [2], SIFT and Fern [15]. The features are robust in terms of change of illumination, scale and rotation. The search method can be addressed as a nearest neighbor searching problem by the following approaches: ANN [1], locality-sensitive-hashing [4] and a vocabulary tree [10]. Rich descriptors are well suited to the matching of feature points with few repetitive texture patterns. On the other hand, 2D maps can be presented in different ways without using descriptors. For example, there are roads, their connectivity and intersections, which should be same according to the manufacturers. Use of such topological features enhances general use of the system because it enables use of maps from different manufactures.

Regarding the object recognition by topological fea-

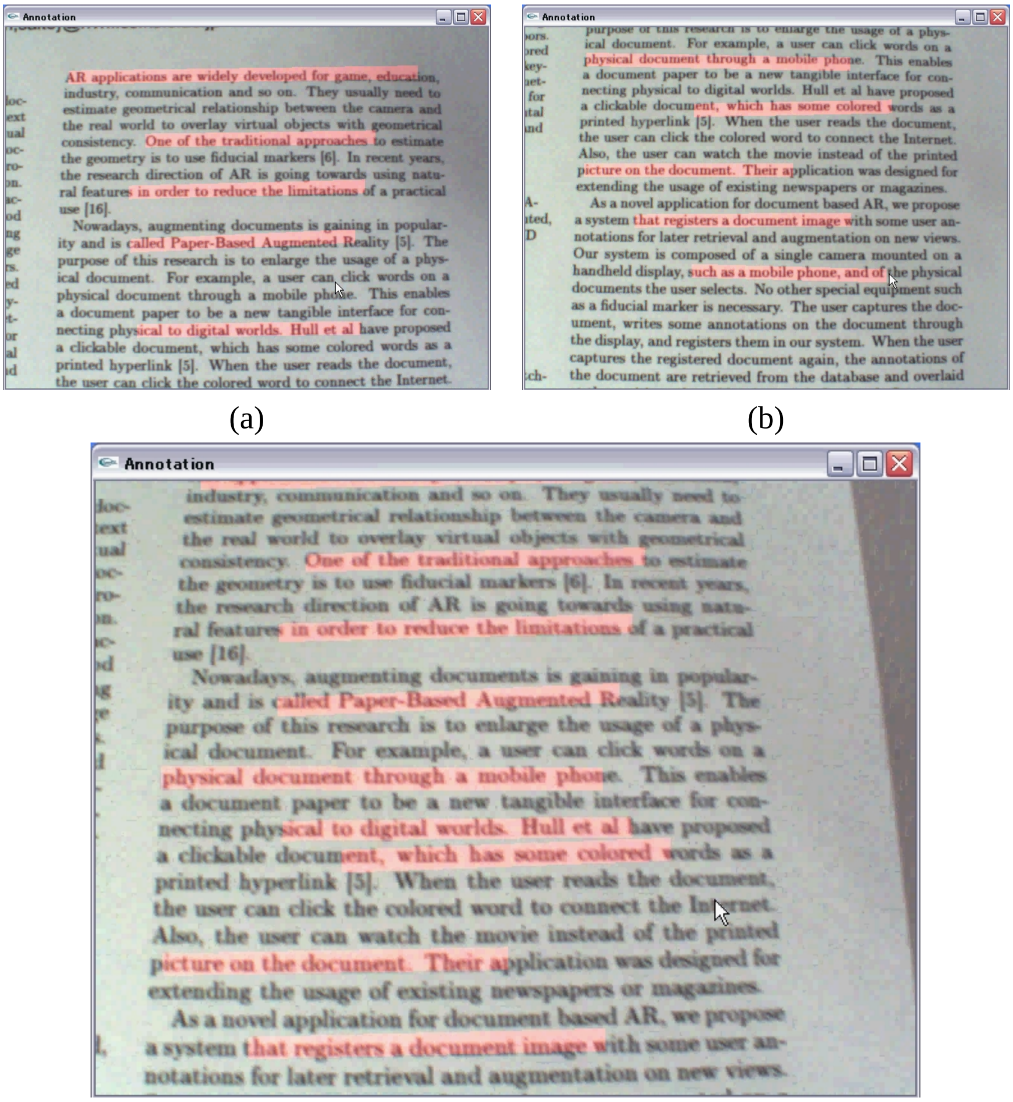
(c)

Figure 5.1: AR annotation. (a) An input image is captured from a nearly top view. (b) Red text is written as a memo. (c) Semi-transparent rectangle is highlighted.

retrieval, the distance between a camera and a document may be a problem as discussed in Section 5.5.

5.2.3 Deletion

This system provides a function to delete a registered document and its annotations. If a user wants to delete the document from the database, the user captures the document again, and clicks a button on a keyboard. Then, the overlaid annotations disappear to show that the document is deleted. Because this process takes less than 1 sec, the user can smoothly delete a document. After a document is removed, the same document can be registered again by capturing it and writing different annotations.



AR applications are widely developed for game, education, industry, communication and so on. They usually need to estimate geometrical relationship between the camera and the real world to overlay virtual objects with geometrical consistency. One of the traditional approaches to estimate the geometry is to use fiducial markers [6]. In recent years, the research direction of AR is going towards using natural features in order to reduce the limitations of a practical use [16].

Nowadays, augmenting documents is gaining in popularity and is called Paper-Based Augmented Reality [5]. The purpose of this research is to enlarge the usage of a physical document. For example, a user can click words on a physical document through a mobile phone. This enables a document paper to be a new tangible interface for connecting physical to digital worlds. Hull et al have proposed a clickable document, which has some colored words as a printed hyperlink [5]. When the user reads the document, the user can click the colored word to connect the Internet.

(a)

purpose of this research is to enlarge the usage of a physical document. For example, a user can click words on a physical document through a mobile phone. This enables a document paper to be a new tangible interface for connecting physical to digital worlds. Hull et al have proposed a clickable document, which has some colored words as a printed hyperlink [5]. When the user reads the document, the user can click the colored word to connect the Internet. Also, the user can watch the movie instead of the printed picture on the document. Their application was designed for extending the usage of existing newspapers or magazines.

As a novel application for document based AR, we propose a system that registers a document image with some user annotations for later retrieval and augmentation on new views. Our system is composed of a single camera mounted on a handheld display, such as a mobile phone, and of the physical documents the user selects. No other special equipment such as a fiducial marker is necessary. The user captures the document, writes some annotations on the document through the display, and registers them in our system. When the user captures the registered document again, the annotations of the document are retrieved from the database and overlaid

(b)

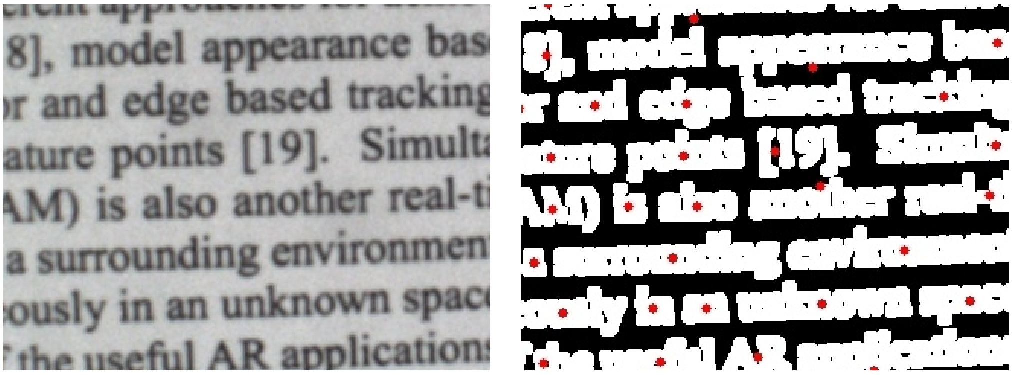
industry, communication and so on. They usually need to estimate geometrical relationship between the camera and the real world to overlay virtual objects with geometrical consistency. One of the traditional approaches to estimate the geometry is to use fiducial markers [6]. In recent years, the research direction of AR is going towards using natural features in order to reduce the limitations of a practical use [16].

Nowadays, augmenting documents is gaining in popularity and is called Paper-Based Augmented Reality [5]. The purpose of this research is to enlarge the usage of a physical document. For example, a user can click words on a physical document through a mobile phone. This enables a document paper to be a new tangible interface for connecting physical to digital worlds. Hull et al have proposed a clickable document, which has some colored words as a printed hyperlink [5]. When the user reads the document, the user can click the colored word to connect the Internet. Also, the user can watch the movie instead of the printed picture on the document. Their application was designed for extending the usage of existing newspapers or magazines.

As a novel application for document based AR, we propose a system that registers a document image with some user annotations for later retrieval and augmentation on new views.

(c)

Figure 5.2: Multiple augmentations. Small parts of the document are individually registered such as (a) and (b). When both parts are simultaneously captured, the annotations of both parts are overlaid such as (c).


8], model appearance bas
or and edge based tracking
ature points [19]. Simult
AM) is also another real-ti
a surrounding environmen
ously in an unknown spac
the useful AR application

(a) (b)

Figure 5.3: Keypoint extraction. (b) is the result of keypoint extraction of (a). Word regions can be segmented by using an adaptive thresholding method with a fixed filter size.

5.3 Algorithm

The algorithm used in this system is geometric feature based keypoint tracking. The detailed procedure for each process is described as follows.

5.3.1 Registration

In the system, document images are not directly stored. When a user captures a document, keypoints in the document are extracted from the image as illustrated in Figure 5.3. By using an adaptive thresholding method, word regions are extracted as blobs in the binary image. In the extracted word regions, some words cannot be segmented and are extracted as one region. In LLAH, such region can be regarded as a noise and does not significantly affect the matching result.

For extracted word regions, the center of each region is computed and dealt as a keypoint. This system stores the keypoints and their descriptors of LLAH included in the captured document image into a document database. The linkage between the document and contents added by a user is also stored.

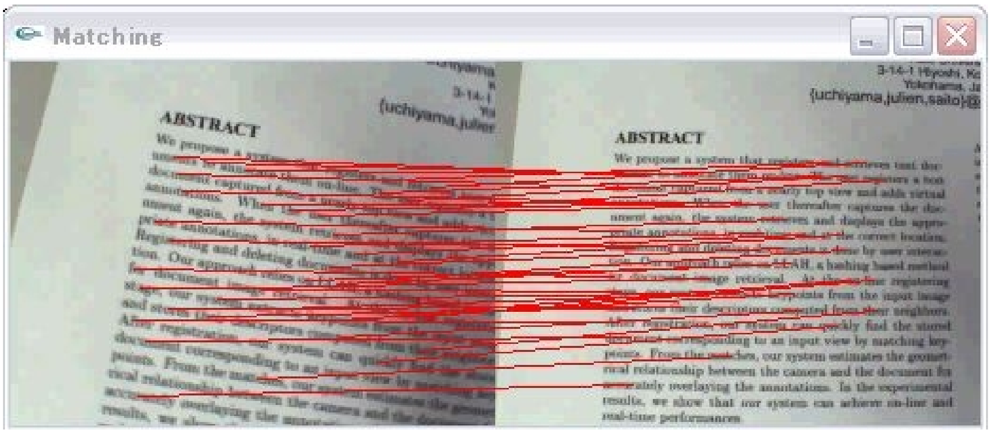


Figure 5.4: Keypoint matching on documents. As a reference, the right image is registered. In retrieval, an input image illustrated in the left has correspondences with the right image.

5.3.2 Retrieval

Retrieval is based on LLAH based keypoint matching between an input image and document images in the database. From a captured image, keypoints and their descriptors are computed by the same way as that in registration. For each keypoint, a correspondence with the keypoints in the document database is established. By performing geometric verification based on a constraint such as correspondences on planar, only correct correspondences are selected to retrieve a corresponding document of the captured document as illustrated in Figure 5.4.

In the retrieval of multiple documents, geometric verification is performed for several documents in the database. When two registered documents are captured in an image, two documents in the database can be retrieved from the correspondences between keypoints in the image and those in the database. If both retrieved documents clear geometric verification, they are judged as visible.

Once a document is retrieved, the keypoints in the image are tracked by geometric feature based keypoint tracking.

5.3.3 Deletion

When the user deletes a registered document, the system needs to removes all data included in the registered document. In this system, each registered document

Table 5.1: Parallel processing. In order to assign the processes to two CPUs equivalently, each processing time was measured.

Process	msec	Process	msec
Image capture	10	Matching by LLAH	23
Keypoint extraction	20	Pose estimation	2
		Index update	4

has the keypoints, their descriptors and annotations. From the database, they are removed.

5.4 Parallel Processing

In the development of AR applications, it is necessary for real-time processing to reduce the computational costs. If there are some CPUs such as Intel Core 2 Duo, they should effectively be utilized.

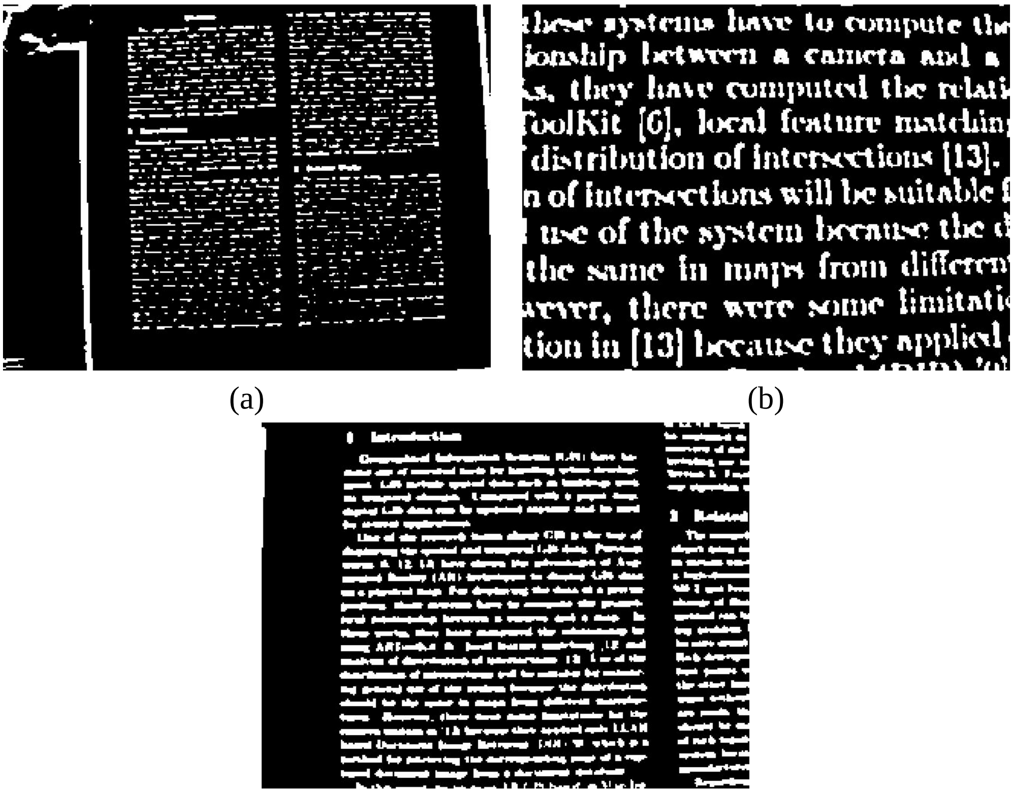
In order to determine the assignment of processes, each processing time is measured from an experiment. The processes are as follows: image capture, keypoint extraction, matching by LLAH, pose estimation and index update as described in Table 5.1. Each processing time was measured by using 100 different images.

From the result, the processes are divided into two parts to assign the processes equivalently. On a first CPU, image capture and keypoint extraction are assigned. On a second CPU, matching by LLAH, pose estimation and index update are assigned. As a result, each CPU will finish the processes within 30 msec to achieve 30 fps.

5.5 Limitation

The distance between a camera and a document is still an issue. Especially, a camera should be far from a document when multiple parts of the document need to be retrieved as described in Section 5.2.

In the case that a camera is far from or very close to a document as illustrated



(a) (b)

(c)

Figure 5.5: Limitation. If a camera is far from or close to a document as (a) and (b), word regions cannot be extracted. The appropriate distance as (c) is determined by character, image and filter sizes.

in Figure 5.5(a) and (b), word regions cannot be extracted because of a fixed filter size in the adaptive thresholding method. The distance is determined by several elements such as character size on a document, the image size and the filter size. In order to solve this problem, the previous camera pose may be helpful for determining these parameters.

5.6 Conclusion

A novel augmented reality system for text documents is developed. Thanks to the augmented reality approach, the user can virtually write any annotations on documents, and watch them later. This system works as a virtual bookmarking.

The characteristic of this system is that the user can use text documents on hands compared to traditional systems using text documents. Any preparation is not necessary for the user. The extension toward many kinds of languages should be considerable. Because the framework for making a connection between user's text document and digital information, this system has a great potentiality to be extended to various types of applications.

From the technical point of view, geometric feature based keypoint tracking can contribute to the wide range of camera movement for augmented reality. In addition, multiple documents can be retrieved by performing geometric verification for several documents in the database.

Chapter 6

Augmented Maps

6.1 Introduction

Geographical Information Systems (GIS) are becoming essential tools for local authorities for studying, handling and planning urban development. Because GIS have functionality as a database, the geographic data are stored in separated layers. In order to generate maps from GIS, several layers are superimposed together depending on the demand of the maps. Compared to printed paper maps, digital geographic data on GIS can be updated anytime and widely utilized for applications such as automotive navigation systems.

In the studies of GIS, geographic data visualization is one of the issues [47], which is called geo-visualization [57]. This visualization is an important issue for geo-spatial data analysis because the data should efficiently be displayed to users. These days, geographic data on GIS are going toward 3D to store building models and the shape of a mountain. From the 3D data, the following query to GIS will be possible; walls that have more than eight hours sunlight in winter and less than two hours in summer. Because 3D geographic data is dealt in GIS, 3D geographic data visualization should be discussed.

Several types of geo-visualization techniques are developed such as web based geo-visualization [36]. Lin et al. discussed about the relationship between multi-dimensional data sets and a client-server structure for real-time interactive and explorative visualization [52]. Steiner et al. developed a tool running on a standard web browsers [98].

AR is another type of visualization. Hedley et al. developed ARToolKit based 3D data visualization system [34]. When a user holds a marker over a camera, 3D mountain was overlaid on the marker. Reitmayr et al. used a camera-projector system for the augmentation [86].

In the previous works of AR based visualization, the semantic relationship between 3D contents and paper maps is not much established. For the meaningful augmentation of the maps, the paper maps and their 3D contents should semantically be connected. For this semantic registration, paper maps with intersections are utilized. Intersections are sort of dot markers. By analyzing the local arrangement of intersections, map image retrieval is performed. In addition, 3D geographic data is also overlaid on the map.

6.2 System Overview

The system focuses on visualization of geographic data extracted from GIS to assist users to watch geographic data for geospatial analysis by AR.

6.2.1 Preparation

First, a user selects several regions for watching geographic data on GIS. Then, GIS automatically outputs 2D maps of selected regions and the coordinates of intersections. Finally, the user prints maps, and input the data exported from GIS into the system.

6.2.2 Usage

A user has a camera mounted handheld display as illustrated in Figure 6.1. The maps are set on a table. When the user watches one of the maps through the display, the user can watch its corresponding geographic data overlaid on the map. The geometric consistency between the map and the geographic data is established. When the user selects another map, the user can watch the geographic data of the selected map.

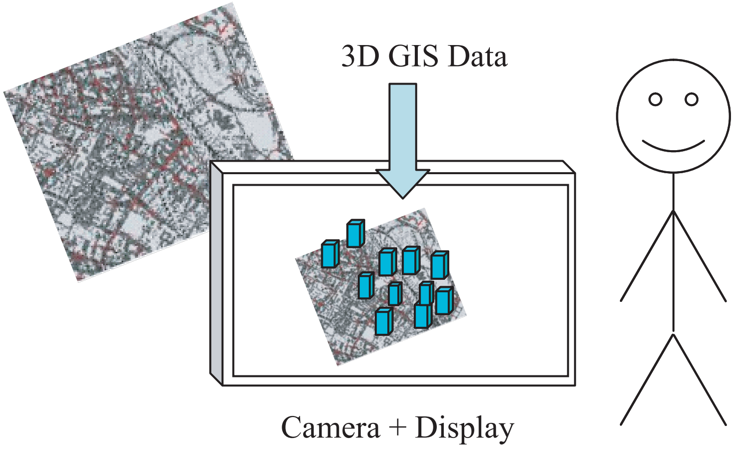


Figure 6.1: AR geo-visualization. A user has a camera mounted handheld display. When the user watches one of paper maps generated from GIS, the user can watch the geographic data on the map with geometric consistency.

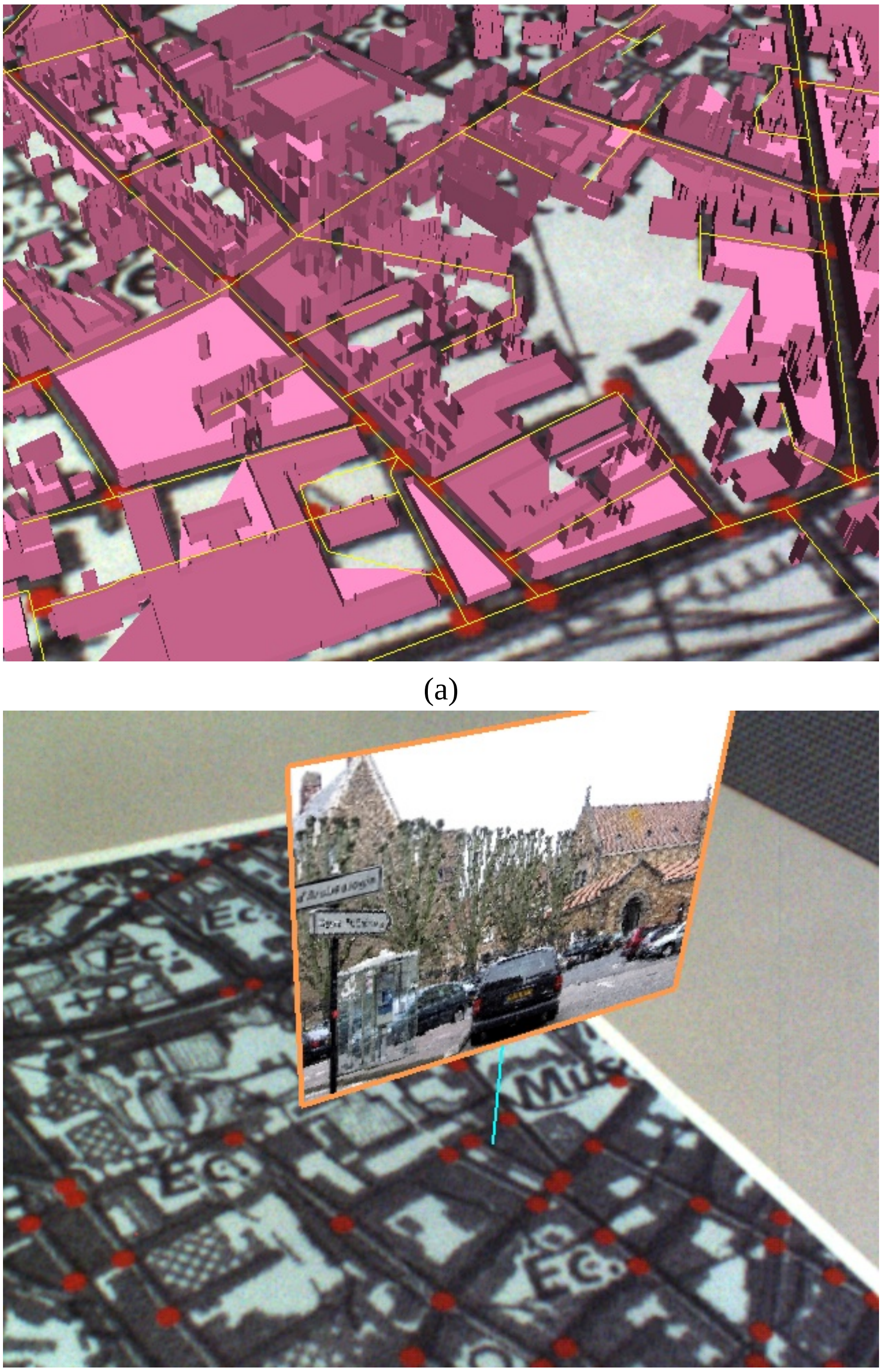
6.2.3 Contents

GIS have different types of geographic data. For example, 3D geographic data such as buildings are stored in GIS. In Figure 6.2(a), 3D buildings are overlaid on the map.

In the future, the connection with web applications for geography such as Google map might be considerable. Pictures and movies in Google map have already had geo-tagged data. Figure 6.2(b) shows the concept of the application. The pictures extracted from Google map are overlaid. Because each picture has shooting location information, the picture can be overlaid at the location.

6.3 Map Image Retrieval

Map image retrieval means the retrieval of a corresponding map from a database using a captured map as a query. Compared with document image retrieval [73], the main difference is the target object.



(a)

(b)

Figure 6.2: Contents. (a) 3D building models (pink) and the road network (yellow) of the map are overlaid. (b) A picture is overlaid at its shooting location. This is a concept to connect with Google map.

6.3.1 Features in Maps

In the studies of object retrieval, discriminative features are necessary to distinguish objects. In map image retrieval, the features should be selected from the geographic data on GIS.

In the layers on GIS, there is a 2D raster map image, which is equivalent to a normal map with roads, buildings and map symbols. If this map is dealt as normal texture, a solution may be a texture based approach. However, a different approach is sought in this work because novel knowledge for geographic data matching dedicated to GIS may be found. At the beginning of this work, the local arrangement of intersections is assumed as a distinctive feature because intersections seem to be randomly distributed.

6.3.2 Maps with Intersections

First, intersections need to be extracted from a captured map. There is an approach to extract intersections from 2D raster maps [14]. However, it is too difficult to apply to AR systems because the method needs huge computational costs. Because the main purpose of this work is to achieve map image retrieval, intersections are printed on the maps to extract them by color segmentation.

The example of maps with intersections is illustrated in Figure 6.3. When a 2D raster map is extracted from GIS, intersections are also extracted from a road network by using a SQL query.

6.3.3 Retrieval by LLAH

In order to utilize the local arrangement of intersections as a feature, the problem is the way of describing the arrangement. Because the local arrangement is regarded as geometric feature, LLAH is selected as a descriptor.

In a pre-processing, the world coordinates of intersections included in maps are extracted from GIS to make the index database of LLAH. In map image retrieval, one of the maps is firstly captured from a nearly top view. Because intersections are printed with a specific color, they can simply be extracted by extracting its color. The center of each extracted blob is dealt as a keypoint. For

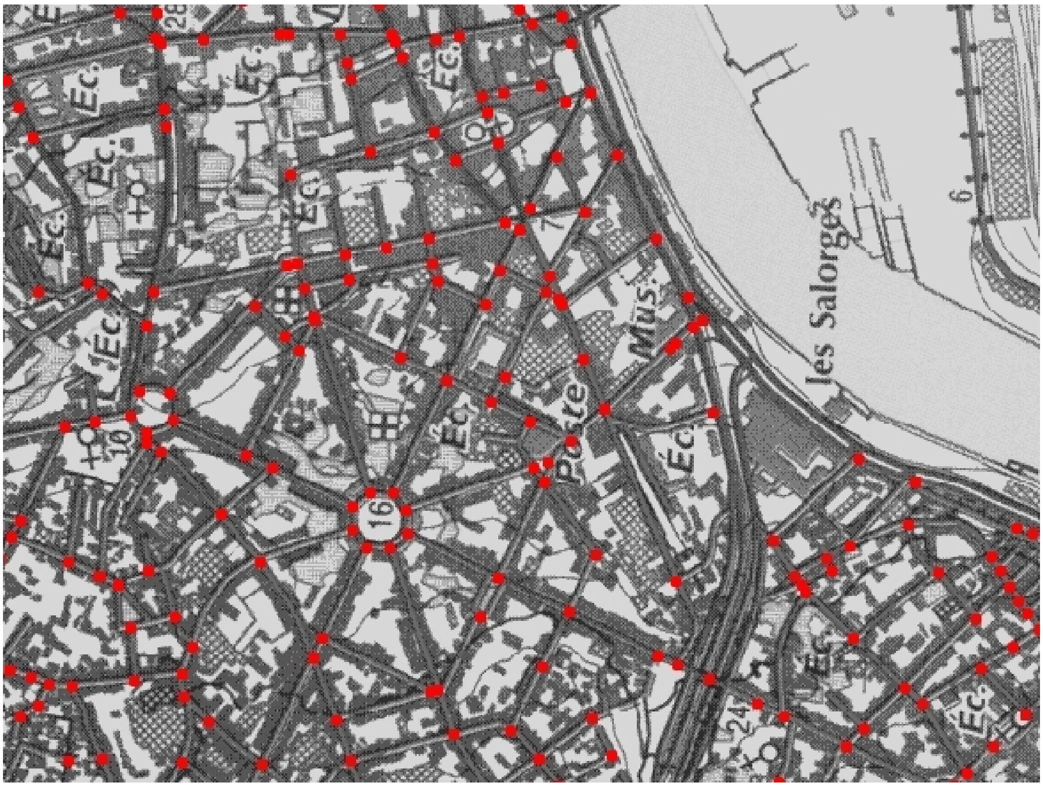


Figure 6.3: Map with red intersections. The map is generated from GIS by using a SQL query to extract intersections from a road network.

each keypoint, the indices of LLAH are computed to make the correspondences with intersections in the database as illustrated in Figure 6.4. After the matching, RANSAC based camera pose estimation described in Appendix is performed. By using RANSAC, the outliers can be removed from the correspondences.

6.3.4 Augmentation

After the map is retrieved, the contents related with the map are overlaid on the map. Because correspondences of intersections are established, the geometric registration between the map and the contents is performed.

After the retrieval is succeeded, geometric feature based keypoint tracking starts for stable augmentation of the contents. A camera can be moved close to the map or tilted.

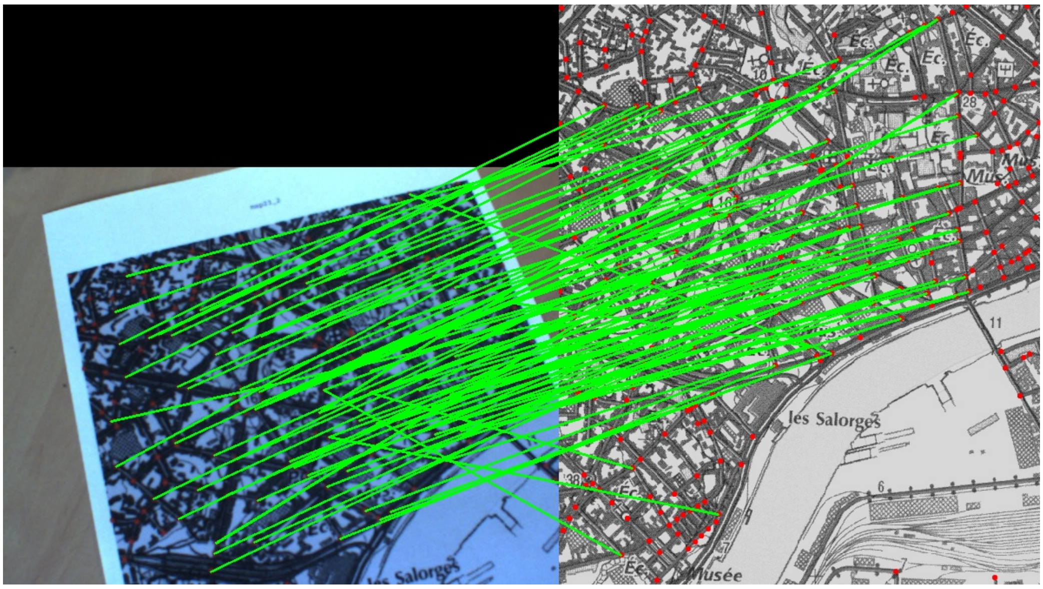


Figure 6.4: Matching intersections. In order to establish correspondences of intersections between a captured image (left) and a database image (right), LLAH is utilized to describe the geometrical relationship of intersections.

6.4 Experiments

This experiment utilizes real GIS data in Nantes [78]. In Nantes, there are 3760 intersections. This application is implemented on a laptop with 2.2GHz CPU, 3GB RAM and 640×480 pixels' camera. The camera is calibrated beforehand.

6.4.1 Map Image Retrieval

First, effectiveness of map image retrieval by matching intersections is evaluated.

As illustrated in Figure 6.5(a), a rectangle region around Nantes is segmented into 10×10 maps at an equal interval. In 100 maps, 48 maps include intersections. The number of intersections included in each map is listed in Table 6.1. For example, 3 maps include from 21 to 30 intersections. In this experiment, each map is printed on an A4 paper. Then, each map is captured from a top view.

As described in Table 6.1, the number of success counts is 44 maps (91.7%). This represents that geometrical relationship of intersections is useful for retrieving maps of a city. If the number of intersections is more than 40, map image

Table 6.1: Effectiveness of map image retrieval. For 48 maps generated from GIS, map image retrieval is performed. 44 maps are successfully retrieved.

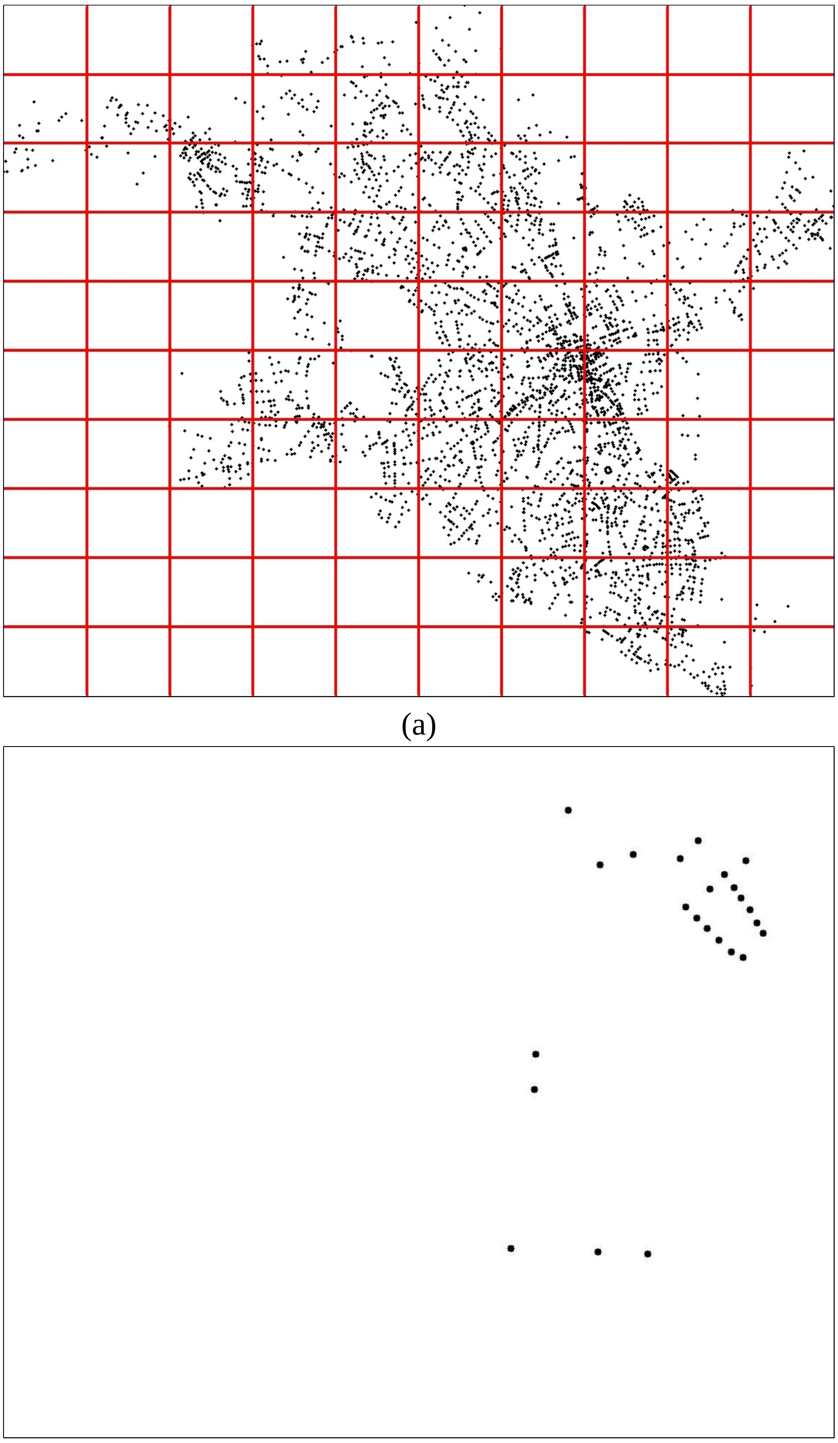
Intersections	Maps	Successes
21-30	8	5
31-40	3	2
41-50	2	2
51-60	9	9
61-70	3	3
71-80	6	6
81-90	3	3
91-100	3	3
101-	11	11
Sum	48	44

retrieval is successfully performed. The failure cases are 3 maps including from 21 to 30 intersections and 1 map including from 31 to 40 intersections. The example of failure cases is illustrated in Figure 6.5(b). In this case, the arrangement of some intersections is a line. Because a ratio of two triangles utilized in LLAH does not work for a line, matching intersections fails. In order to handle this case, the solution is to use several sizes or colors of intersections to make new discriminative relationship.

6.4.2 Processing Time

Next, a processing time is evaluated to prove that geometric feature based approach is applicable to AR systems.

The processes are mainly divided into two parts: intersection extraction, map image retrieval and model rendering for AR. The average processing time is described in Table 6.2. For 48 maps, it took 25 msec. The total computational time for intersection extraction and map image retrieval is less than 40 msec. This is enough for AR systems. However, building rendering needed 130 msec. Because buildings from GIS were detailed such as 7000 buildings, each building has many vertices. For fast rendering, the simplified models should be generated.



(a)

(b)

Figure 6.5: Intersections in Nantes. (a) By segmenting a whole Nantes map, 48 maps are generated. (b) Map identification fails when intersections are in a line.

Table 6.2: Processing time. Map image retrieval is performed in real-time. However, the 3D building rendering need a computational cost because the buildings on GIS are detailed.

Process	msec
Intersection extraction	9
Map image retrieval	25
Building rendering	130

6.5 Conclusion

AR based geographic visualization system for GIS using paper maps is developed. The system is based on the framework of map image retrieval based on matching intersections. The local arrangement of intersections is utilized as a feature for the retrieval.

In this system, maps with intersections are generated From GIS. When a map is captured, the corresponding map of the captured map is retrieved from the database by matching intersections by LLAH. From the matching result, 3D geographic data can be overlaid on the map with geometric consistency. The experiment proved that the local arrangement of intersections was enough distinctive for map image retrieval.

In future works, grid arrangement of intersections should be handled because the intersection arrangement is sometimes regular grid in Japan. In this case, several colors or sizes should be applied to intersections to make new arrangement depending on each color or size. A link with Google map or other geographic data is also established. This framework of map image retrieval has a potential to be the normal way to watch geographic data on maps. In addition, printed intersections can be utilized for user interaction such that a user selects the location by clicking them.

Chapter 7

Photogrammetric System Using Lights

7.1 Introduction

Digital photogrammetry is one of remote sensing technologies for estimating geometric shapes from images [16, 106]. It is applied to monitoring systems for measuring the degree of the distortion of large constructions because it is a non-contact and safe [29, 56].

For example, Moraa et al. developed a monitoring system for a landslide by using aerial images [67]. Lim et al. used a active laser scanner for monitoring cliff evolution [51]. Jiang et al. reported a research history and the state of the art technologies about bridge monitoring systems [40].

In order to measure the 3D world coordinates of specific locations, point marker based photogrammetric systems are utilized [50]. In these systems, a user usually puts markers at measuring locations. As a demand, a construction company indicates two points as follows.

- Matching markers in two views is manual or semi-automatic.
- Extracting markers from an image captured in the dark is difficult.

Especially, measurement of expansion and contraction of a bridge should be performed for a whole day including night. When the measurement is performed in the dark, many lights are set to make enough brightness for extracting markers.

In order to meet the demand mentioned above, a light marker based photogrammetric system is developed. In this system, a light blinks because the change of the brightness is utilized as a temporal feature. In addition, it is extractable from images captured in the dark because it emits by itself. From these properties, a light is desirable marker for outdoor photogrammetry.

7.2 System Overview

The system is composed of a digital camera, light markers and a computer. Because target objects are outdoor large constructions, a high resolution digital camera is necessary. As a camera, Nikon D300 is selected, which resolution is 4288×2848. This camera can capture 100 images in the continuous shoot mode.

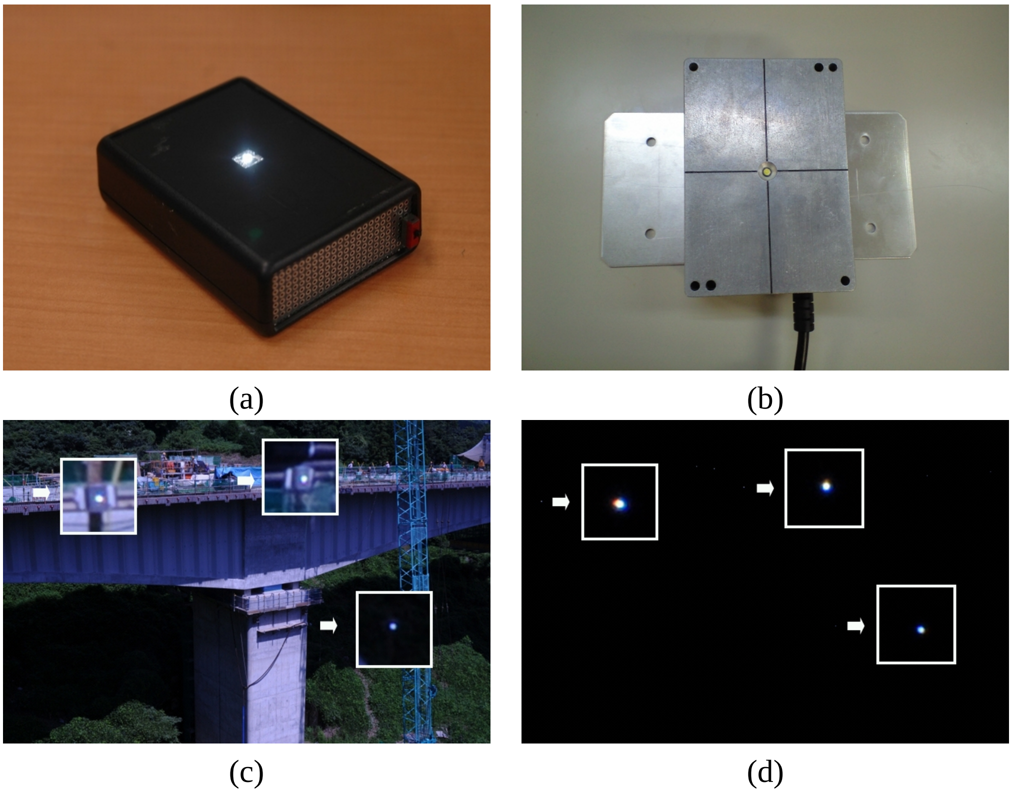
7.2.1 Light

For light markers, LED lights are selected because they are widespread and becoming cheaper. Their blinking patterns are controllable such that lights have set arbitrary blinking patterns at arbitrary constant interval. Each light has independent blinking pattern as a unique number. In order to respond several cases, two different lights are developed as illustrated in Figure 7.1. The smaller one works with a battery for one-time measurement. The larger one needs a wired power source for a whole day measurement. As illustrated in Figure 7.1(c) and (d), the brightness of LED lights is enough for extracting lights in the daytime and the dark.

7.2.2 Usage

Before using the system, the serial shoot speed of a camera need to be measured. The capturing speed is set on the blinking speed of each light. The capturing and blinking speeds should be same for receiving blinking pattern.

First, lights are set on both reference locations and measuring locations as well as traditional photogrammetry. For the reference locations, their 3D coordinates



(a) (b)

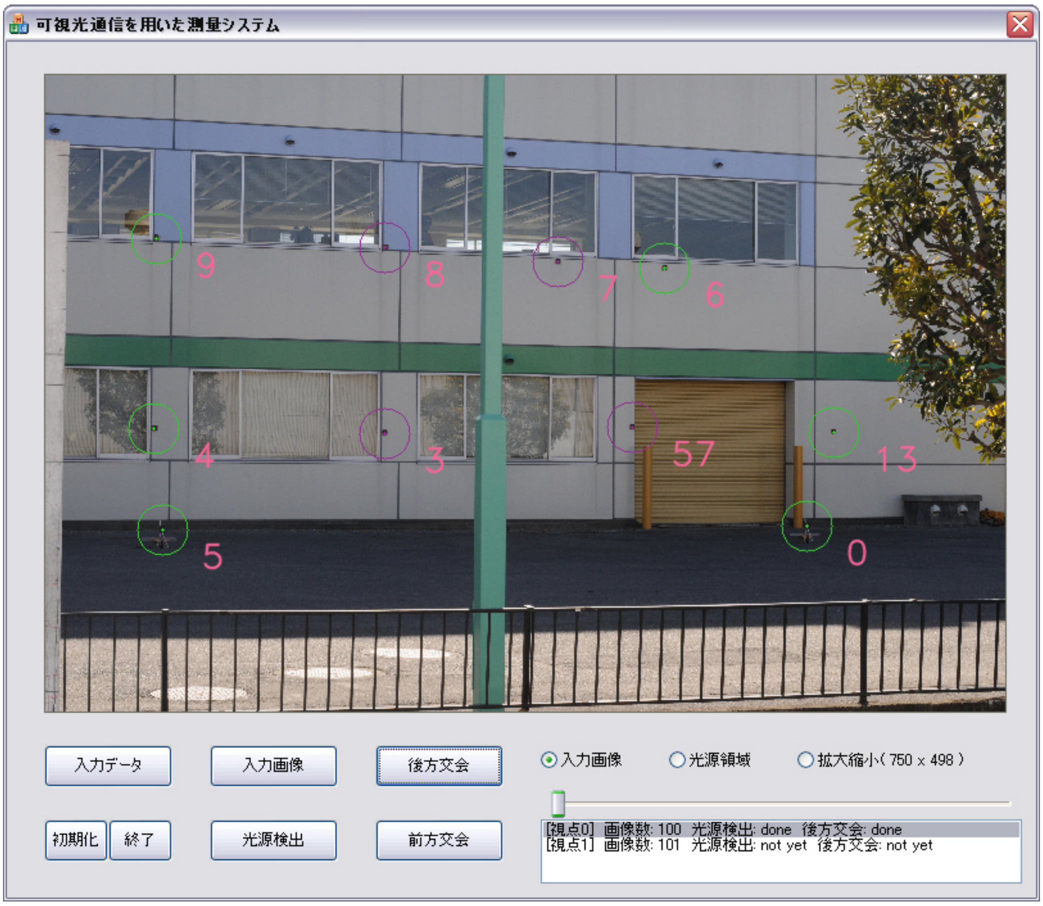
(c) (d)

Figure 7.1: LED lights. (a) The smaller LED light has a battery. (b) The larger LED light needs a wired power source. (c) In the daytime, LED lights of (b) were extractable. (d) In the night, LED lights of (b) were extractable.

are measured with a range finder such as a total station by making a world coordinate system. The unique number and world coordinate of a light at each reference location are input into the system.

After lights are set on the site, they are captured from more than two views. At each fixed view, 100 images are consecutively captured by a camera. The total capturing time by Nikon D300 is 16 seconds.

Next, images captured at each view are input into the system as illustrated in Figure 7.2. For each view, the system automatically extracts lights and their unique numbers. From the number, each light is judged as a reference location or measuring location. By using the reference locations, a camera pose of each view is computed as camera pose estimation. For measuring locations, matching lights in different views is done by finding the same number. Finally, the 3D world



```
[視点0]  画像数: 100  光源検出: done  後方交会: done
[視点1]  画像数: 101  光源検出: not yet  後方交会: not yet
```

Figure 7.2: GUI. First, a user stores captured images in the camera, and then input them into the system. The user operates the extraction of lights, camera calibration and triangulation on GUI.

coordinate of a light at each measuring location is computed by triangulation.

7.3 Actual Uses

This system was actually applied to the outdoor photogrammetry of two large constructions: building and tunnel. The result of each measurement is described in this section.

Table 7.1: Accuracy of photogrammetry for a building. The average error is 6mm. The accuracy of Y was worse than that of X and Z because of the arrangement of lights.

	Ground truth (m)			Estimated (m)			Error (mm)		
ID	X	Y	Z	X	Y	Z	X	Y	Z
6	59.227	50.000	25.020	59.225	50.004	25.018	2	4	2
7	57.164	49.998	25.064	57.164	50.003	25.060	0	5	4
8	53.998	50.002	25.207	53.998	50.006	25.208	0	4	1
9	49.991	49.996	25.234	49.992	50.005	25.230	1	9	4

7.3.1 For Building

As a first trial of the system, the measurement of the shape of a building was performed as illustrated in Figure 7.3. In this case, smaller LED lights are utilized.

As well as traditional photogrammetry, the world coordinates of all light are measured by using a range finder. For photogrammetry, lights from No.0 to No.5 are utilized as reference points. The world coordinates of lights from No.6 to No.9 are estimated by triangulation. The estimated coordinates of the lights from No.6 to No.9 are compared with the coordinates measured by the range finder.

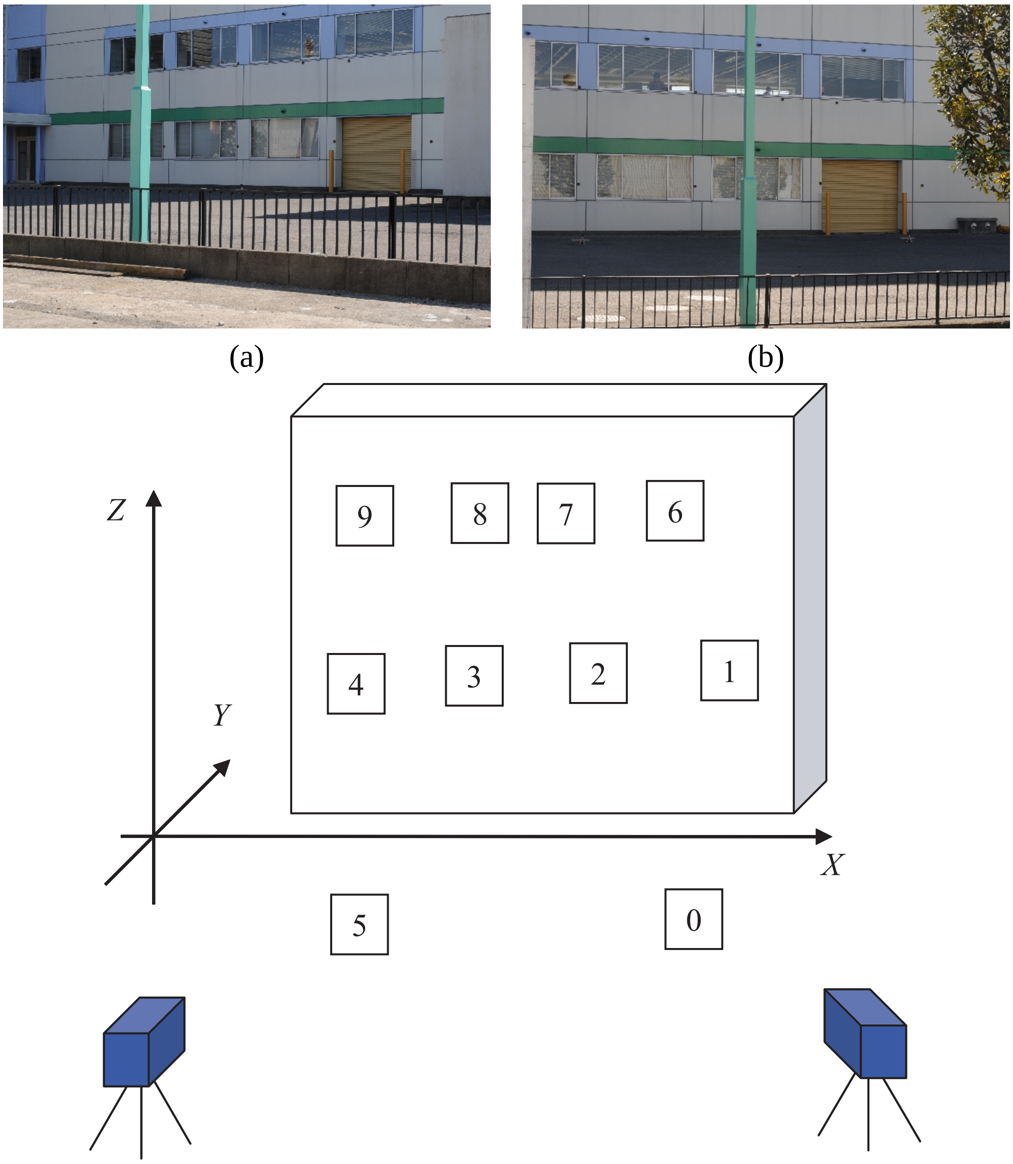
As described in Table 7.1, the average error is 6 mm. The accuracy of Y was worse than that of X and Z because lights from No.1 to No.4 were on the same plane. If the lights can be distributed, the accuracy will be better. From this experiment, the company decided to apply to another larger construction site because the accuracy was acceptable.

7.3.2 For Tunnel

This system was applied to the measurement of the shape of a tunnel under construction. Because the illumination lamps were not installed, the tunnel was dark. For this case, lights are really useful as markers.

As illustrated in Figure 7.4, lights are set on the wall of a tunnel at an actual construction site.

The detailed procedure of the measurement in as follows. The world coor-

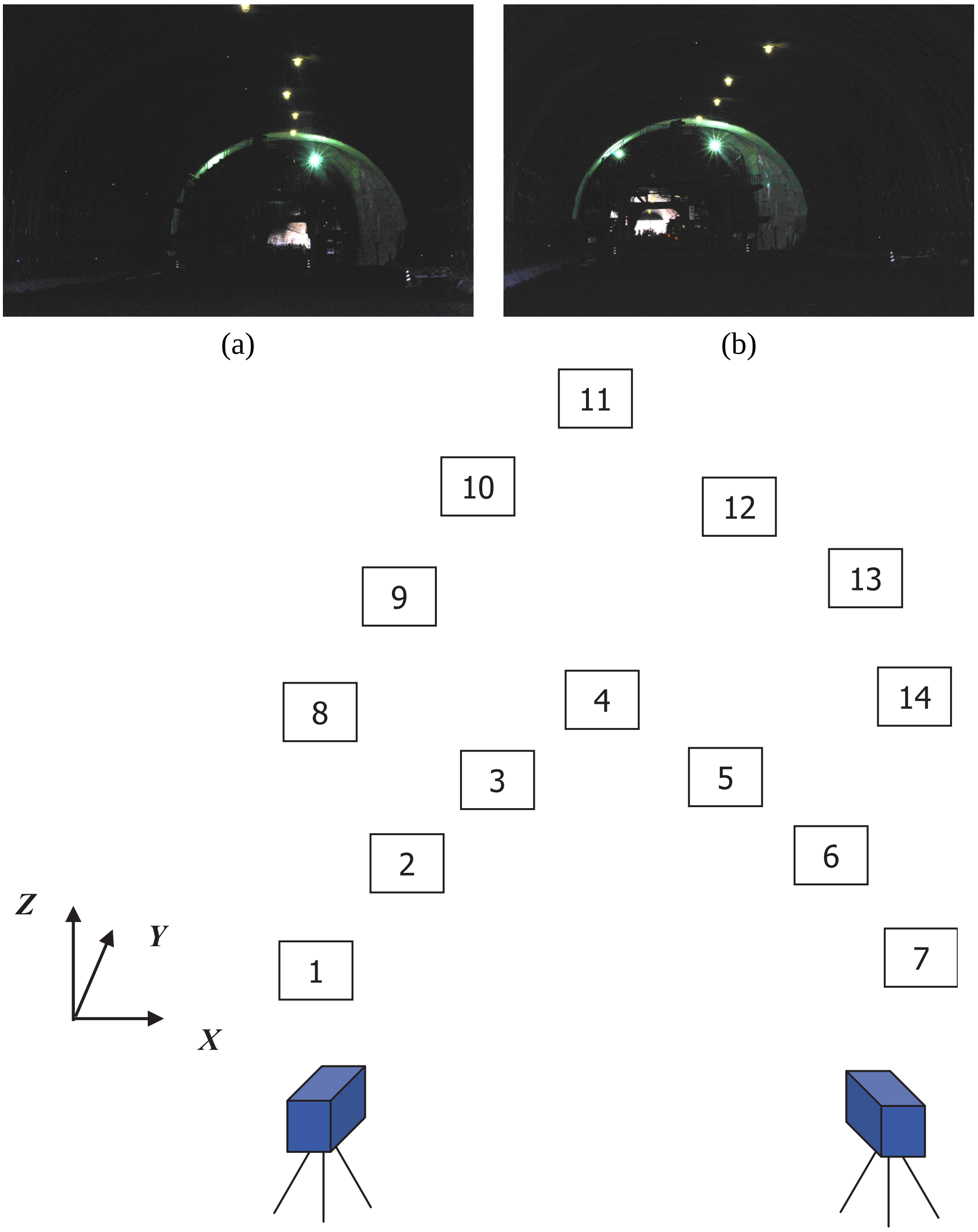


(a)

(b)

(c)

Figure 7.3: Photogrammetry for a building. (a) The image is captured from a right side. (b) The image is captured from a left side. (c) The lights are set on the wall and in front of the building.


(a)

(b)

(c)

Figure 7.4: Photogrammetry for a tunnel. (a) The image is captured from a right side. (b) The image is captured from a left side. (c) The lights are set on the wall of a tunnel.

Table 7.2: Accuracy of photogrammetry for a tunnel. The average error is 7mm.

	Ground truth (m)			Measured (m)			Error (mm)		
ID	*X*	*Y*	*Z*	*X*	*Y*	*Z*	*X*	*Y*	*Z*
8	42.622	76.178	11.072	42.620	76.179	11.068	2	-1	4
9	42.623	80.724	11.163	42.620	80.725	11.156	3	-1	7
10	42.369	80.822	14.689	42.370	80.923	14.682	-1	-1	7
11	48.976	80.907	19.286	48.972	80.908	19.279	4	-1	7
12	57.540	80.969	14.687	57.532	80.970	14.683	8	-1	4
13	57.317	80.966	11.046	57.305	80.961	11.042	12	5	4
14	57.339	76.470	11.089	57.333	76.470	11.088	6	0	1

dinates of all lights are measured by using the range finder at first. For photogrammetry, lights from No.1 to No.7 are utilized as reference points. The world coordinates of lights from No.8 to No.14 are estimated by triangulation. The estimated coordinates of lights from No.8 to No.14 are compared with the coordinates measured by the range finder.

As described in Table 7.2, the average error is 7mm and the maximum error is 13 mm. Compared to the case of a building, the accuracy is almost same because the size of the target construction is almost same. Because the accuracy depends on image resolution and the size of a target construction, a camera should be selected by considering a demand for the accuracy.

7.3.3 Processing Time

First, capturing 100 images at all viewpoints are performed. Because the capturing is 16 sec per viewpoint, the total capturing time is a few minutes.

Next, the recognition of lights at each view and triangulation are performed in the system. For each view, the extraction of lights and the receipt of their data take about 2 mins. Because the triangulation takes less than 1 sec, the total time will be about 10 mins. The construction company can accept this time for on-site measurement.

7.4 Conclusion

An outdoor photogrammetric system using light markers is developed. This system can be applied in the dark because lights emit by themselves to be visible. In addition, matching markers in different views is automatically performed by embedding unique number into the blinking patterns. For developing the system, the technology of image sensor based visible light communication is utilized.

In future works, the implementation will be on Field Programmable Gate Array (FPGA) for acceleration. Because the process for each pixel is same, the process can be parallelized by using special units. Also, the combination with GPS will be interesting. If GPS is attached with a light, the global position can be provided from GPS. As a result, the pre-process for making a world coordinate system will be disappeared.

Chapter 8

Conclusions

8.1 Summary

This thesis investigated two different approaches about texture-free keypoint matching. Even though texture based approaches is a mainstream in the studies of feature matching, there are still problems which the approaches cannot solve. By taking different approaches such as geometric feature and temporal feature, novel technical insights were found. The contributions of this thesis are summarized in Table 8.1.

There are two technical contributions using geometric feature and temporal feature. In the study of geometric feature, geometric feature based keypoint tracking was developed. Because the geometric feature is not invariant to the large range of views, the feature changes depending on camera movement. This means

Table 8.1: Summary of contributions. This thesis has two contributions for technical problems and four contributions for novel applications.

	Contribution
Methods	On-line geometric feature learning for keypoint tracking
	Blinking light extraction and blinking pattern reception
Applications	Pool supporting system for a handheld device
	Framework of on-line document registering and retrieving
	Map image retrieval using local arrangement of intersections
	A lights as a marker for outdoor photogrammetry

that geometric feature based keypoint matching is valid within a narrow range. In order to widen the range of camera movement, on-line learning of the feature was developed. When geometric feature based keypoint matching is succeeded, the geometric features at each keypoint are re-computed because some of the features are changed. By updating the changed features at every frame, continuous keypoint matching using geometric features was achieved.

In the study of temporal feature, blinking pattern of a light is recognized from temporal images for keypoint matching. This approach can be regarded as image sensor based visible light communication. The blinking pattern is designed using a signal processing technology to extract the light efficiently and receive the pattern correctly. From temporal images, the candidates of the lights are extracted by checking the rule of the blinking pattern. For only the candidates, all samples from the images are extracted to reduce the use of the memory. For the samples of each pixel, error checking is performed to extract only lights and receive the transmitted data. The blinking light can be utilized for keypoint matching in the dark.

In this thesis, four novel applications using the two approaches were also described. Depending on each purpose, the approaches based on natural features or markers are designed.

An augmented reality based pool supporting system using a handheld device was developed. Because the pool table does not have rich texture, texture based approaches cannot be applied. In order to use table corners for camera pose estimation, their geometrical relationship is utilized. It is important to implement a natural feature based AR system actually working in real-time toward practical use.

In augmented documents, the framework for registering and retrieving documents on-line was developed. This framework can make a connection between text documents a user has and digital information the user adds. Based on this framework, a virtual bookmarking system was developed. Because the text documents are binary, geometric feature based keypoint tracking is utilized for stable augmentation.

Augmented maps took the same approach as that of augmented documents for AR visualization. In addition, geometric feature is also utilized for map image retrieval toward semantic registration between 3D geographic data and paper maps. The maps utilized in this system have red intersections generated from GIS. In the map image retrieval, the local arrangement of intersections is assumed as a discriminative feature. From the experiments, it was proved that the feature was enough for handling the maps of one city.

Photogrametric system using lights were developed to meet the demand from a construction company. The company needs to measure the shape of a large construction in the dark because the change of the shape should be tracked all day including night. A light is selected as a marker to capture it in the dark. Automatic light extraction and matching in different views were achieved by making the lights blinking.

From these contributions, novel directions different from traditional texture based approaches were opened up.

8.2 Future Works

This thesis solved problems which traditional texture based approaches could not solve. However, the solutions in this thesis cannot be applied to normal texture because it is difficult to extract keypoints from the texture stably for utilizing geometric features. This means that the approach of this thesis is to design a method depending on a target object. This situation may not be desirable. The ultimate goal of feature matching is to deal with all objects by all in one solution. For example, feature matching for objects illustrated in Figure 8.1 will be one solution in the future. A two-stage approach is one possibility for the solution. The first stage deals with the object recognition. Depending on the recognized object, the method for keypoint matching is selected in the second stage.

In augmented maps and augmented documents, there is an assumption such that a paper is planar. Because a paper is non-rigid surface, it can be changed into any shape such as curved, folded and cut. For the next step of these researches, the recognition of paper manipulation will be interesting. By using the manipulation as a query, novel interaction will be designed.

A blinking light in the photogrametric system may be utilized in another application. In this system, a light can be regard as sending data. Because the frame

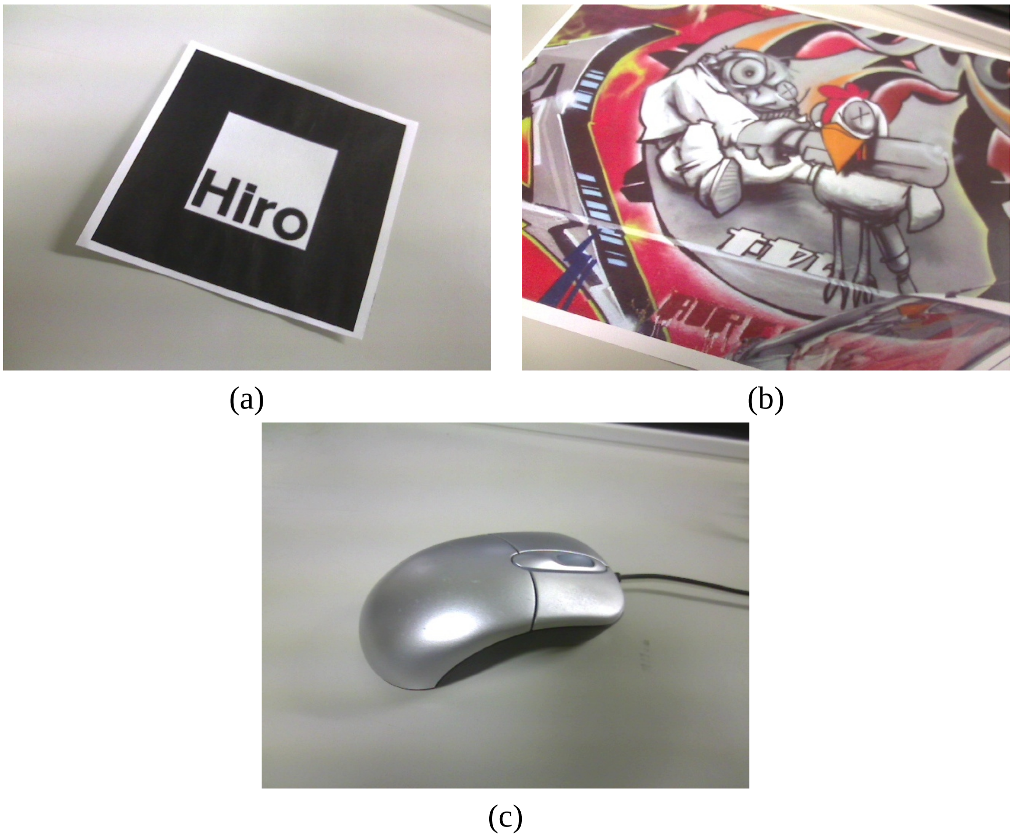


Figure 8.1: Future works. Can the same feature matching work for all objects?

rate of the camera used in this system was quite slow, the speed of transmission was also slow. If the speed of transmission becomes 100 kbps, lights can send large data such as pictures and movies. In this case, light markers can be applied to augmented reality. If a user captures lights, the contents for augmented reality is directly downloaded from lights. Because the acquisition of the contents from images is not much discussed, downloading contents from lights will be an interesting approach.

In the studies on augmented reality, applications for semi-transparent objects should be discussed. From a technical point of view, tracking semi-transparent objects is not solved by computer vision technologies because existing methods such as using texture and geometric feature cannot track the semi-transparent objects. If the tracking is achieved, it is possible to overlay virtual objects inside the semi-transparent objects such that virtual fishes are swimming in an actual aquarium. This may open a novel future in augmented reality.

One of the interesting research domains utilizing feature matching is visual servoing, which is known as vision based robot control [59]. The robot is controlled by feedback information from visual cues based on image analysis. The methods described in this thesis can be directly applied to visual servoing. For example, the localization of a robot using illumination lumps will be achieved by image based visible light communication. Because the lumps are set at both indoor and outdoor locations, they are useful for providing their location information. If the robot captures 6 lumps for camera pose estimation, it can know its precise pose compared to traditional wire less sensor based approaches. When a method for visual servoing is designed, the response time should strictly be cared for smooth control. By developing all in one solution for feature matching as mentioned above, novel feature of visual servoing will also be opened.

Appendix A

Camera Pose Estimation

A.1 Mathematical Preparation

For expressing a camera in geometry, there are several types of camera models such as affine and projective models [28, 33]. From these models, this thesis focuses on the perspective model (pinhole camera model).

As a reference coordinate system, a world coordinate system is first prepared as illustrated in Figure A.1. Camera pose estimation is to compute a camera position and orientation with respect to the world coordinate system.

The geometry between a homogeneous image coordinate $\mathbf{m} = (u, v, 1)$ and a homogeneous world coordinate $\mathbf{X} = (X, Y, Z, 1)$ is formulated as

$$\begin{aligned} \mathbf{m} &= \mathbf{A}\,[\mathbf{R} \mid \mathbf{t}]\,\mathbf{X} \quad &(A.1) \\ \mathbf{m} &= \mathbf{P}\mathbf{X} \quad &(A.2) \end{aligned}$$

where $\mathbf{A}$ is a 3×3 matrix for internal camera parameters, $\mathbf{R}$ is a 3×3 rotation matrix and $\mathbf{t}$ is a 3×1 translation matrix. $[\mathbf{R} \mid \mathbf{t}]$ means a transformation from a world coordinate to a camera coordinate. An image coordinate is computed by multiplying the camera coordinate with $\mathbf{A}$. $\mathbf{A}\,[\mathbf{R} \mid \mathbf{t}]$ can be also expressed as $\mathbf{P}$, which is a 3×4 projection matrix. Since $\mathbf{A}$ is composed of a focal length and a principal point of an image sensor, $\mathbf{A}$ can be measured beforehand as camera calibration using a 2D checker board [108] or a 1D object [109].

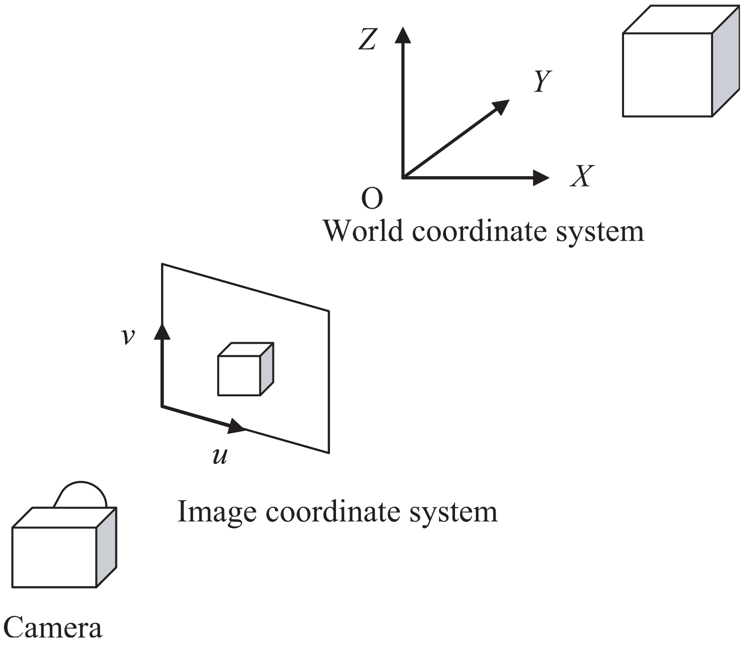


Figure A.1: Image and world coordinate systems. Camera pose estimation is the computation of a geometrical relationship between an image coordinate system and a world coordinate system.

In order to compute a camera pose with respect to the world coordinate system, 2D-3D correspondences between the image coordinate system and the world coordinate system should be established as discussed in this thesis. If these correspondences are established, there are two types of solutions; computing $\mathbf{R}$ and $\mathbf{t}$ in Equation A.1, or $\mathbf{P}$ in Equation A.2. For the former case, many mathematical solutions have already been proposed if $\mathbf{A}$ is known from camera calibration [31, 84, 4]. In Equation A.2, $\mathbf{P}$ is linearly computed if there are more than six 2D-3D correspondences.

A.2 Using Homography

Homography represents a geometrical relationship between 2 planes [33]. The procedure of the derivation of a homography from Equation A.1 is explained using a $Z = 0$ plane.

First, Equation A.1 is re-described as

$$\mathbf{m} = \mathbf{A}\left[\mathbf{r1}\ \mathbf{r2}\ \mathbf{r3} \mid \mathbf{t}\right]\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{A.3}$$

where the elements of $\mathbf{R}$ is described as $(\mathbf{r1}\ , \mathbf{r2}\ , \mathbf{r3})$.

When the geometrical relationship between the image plane and the XY plane is considered, $\mathbf{r3}$ can be ignored. Therefore, Equation A.3 is transformed into

$$\begin{aligned} \mathbf{m} &= \mathbf{A}\left[\mathbf{r1}\ \mathbf{r2} \mid \mathbf{t}\right]\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} && \text{(A.4)} \\ \mathbf{m} &= \mathbf{H}\mathbf{X}' && \text{(A.5)} \end{aligned}$$

where $\mathbf{H} = \mathbf{A}\left[\mathbf{r1}\ \mathbf{r2} \mid \mathbf{t}\right]$ is a 3×3 homography. In order to compute a homography, four correspondences are minimally necessary.

Next, camera pose estimation from a homography is explained. When a homography is computed, a part of the rotation matrix and the translation matrix are computed as

$$\left[\mathbf{r1}\ \mathbf{r2} \mid \mathbf{t}\right] = \mathbf{A}^{-1}\mathbf{H} \tag{A.6}$$

Due to orthogonal, the rest of the rotation matrix is computed as

$$\mathbf{r3} = \mathbf{r1} \times \mathbf{r2} \tag{A.7}$$

Because $\mathbf{r1}\ , \mathbf{r2}\ , \mathbf{r3}$ and $\mathbf{t}$ are estimated, camera pose estimation is completed.

Bibliography

[1] E. H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt, and J. M. Ogden. Pyramid methods in image processing. *RCA Engineer*, 29(6):33–41, 1984.

[2] M. E. Alian, S. B. Shouraki, M. T. M. Shalmani, P. Karimian, and P. Sabzmeydani. Roboshark: a gantry pool player robot. In *International Symposium on Robotics*, 2003.

[3] T. L. Andersen, S. Kristensen, B. W. Nielsen, and K. Gronbak. Designing an augmented reality board game with children: the battleboard 3D experience. In *Conference on Interaction Design and Children: Building a Community*, pages 137–138, 2004.

[4] A. Ansar and K. Daniilidis. Review and analysis of solutions of the three point perspective pose estimation problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:578–589, 2003.

[5] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM*, 45:891–923, 1998.

[6] R. Azuma. A survey of augmented reality. *Presence*, 6(4):355–385, 1997.

[7] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre. Recent advances in augmented reality. *IEEE Computer Graphics and Applications*, 21:34–47, 2001.

[8] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. SURF: Speeded up robust features. *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[9] O. Bimber and R. Raskar. *Spatial augmented reality: merging real and virtual worlds*. A K Peters LTD, 2005.

[10] H. Binti, S. Haruyama, and M. Nakagawa. Visible light communication with LED traffic lights using 2-dimensional image sensor. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E89-A(3), 2006.

[11] J. Borenstein, H. R. Everett, L. Feng, and D. Wehe. Mobile robot positioning: Sensors and techniques. *Journal of Robotic Systems*, 14(4):231–249, 1997.

[12] P. J. Burt and E. H. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, COM-31,4:532–540, 1983.

[13] B. Cheng, J. Li, and J. Yang. Design of the neural-fuzzy compensator for a billiard robot. In *International Confrence on Networking, Sensing and Control*, volume 2, pages 909–913, 2004.

[14] Y. Y. Chiang and C. A. Knoblock. Automatic extraction of road intersection position, connectivity, and orientations from raster maps. 2008.

[15] S. C. Chua, E. K. Wong, and V. C. Koo. Intelligent pool decision system using zero-order sugeno fuzzy zystem. 44:161–186, 2005.

[16] T. A. Clarke, M. A. R. Cooper, J. Chen, and S. Robson. Automated 3-D measurement using multiple CCD camera views. *Photogrammetric Record*, 15:315–322, 1994.

[17] M. Datar, P. Indyk, N. Immorlica, and V. S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Symposium on Computational Geometry*, pages 253–262, 2004.

[18] A. Davison, I. Reid, N. Molton, and O. Stasse. MonoSLAM: real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1167, 2004.

[19] P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach. In *ACM SIGGRAPH*, pages 11–20, 1996.

[20] D. Doermann. The indexing and retrieval of document images: a survey. *Computer Vision and Image Understandinge*, 70:287–298, 1998.

[21] F. Doil, W. Schreiber, T. Alt, and C. Patron. Augmented reality for manufacturing planning. In *Workshop on Virtual environments*, pages 71–76, 2003.

[22] V. Ferrari, T. Tuytelaars, and L. Van Gool. Integrating multiple model views for object recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 105–112.

[23] M. Fiala. Vision guided control of multiple robots. In *Canadian Conference on Computer and Robot Vision*, pages 241–246, 2004.

[24] M. Fiala. Structure from motion using SIFT features and the PH transform with panoramic imagery. In *Canadian conference on Computer and Robot Vision*, pages 506–513, 2005.

[25] J. Fischer, D. Bartz, and W. Straser. Occlusion handling for medical augmented reality using a volumetric phantom model. In *ACM Symposium on Virtual Reality Software and Technology*, pages 174–177, 2004.

[26] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24:381–395, 1981.

[27] P. Forssen and D. Lowe. Shape descriptors for maximally stable extremal regions. In *IEEE International Conference on Computer Vision*, pages 1–8, 2007.

[28] D. A. Forsyth and J. Ponce. *Computer vision: a modern approach*. Prentice Hall Professional Technical Reference, 2002.

[29] C. S. Fraser and B. Riedel. Monitoring the thermal deformation of steel beams via vision metrology. *ISPRS Journal of Photogrammetry and Remote Sensing*, 55(4):268–276, 2000.

[30] C. Geiger, C. Reimann, J. Sticklein, and V. Paelke. JARToolKit - a java binding for ARToolKit. In *IEEE Augmented Reality Toolkit Workshop*, 2002.

[31] R. M. Haralick, C.-N. Lee, K. Ottenberg, and M. Nolle. Review and analysis of solutions of the three point perspective pose estimation problem. *International Journal of Computer Vision*, 13:331–356, 1994.

[32] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, pages 147–151, 1988.

[33] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, second edition, 2004.

[34] N. R. Hedley, M. Billinghurst, L. Postner, R. May, and H. Kato. Explorations in the use of augmented reality for geographic visualization. *Teleoperators and Virtual Environments*, 11:119–133, 2002.

[35] B. K. P. Horn. *Robot vision*. MIT Press, 1986.

[36] B. Huang and H. Lin. GeoVR: a web-based tool for virtual reality presentation from 2D GIS data. *Computers and Geosciences*, 25:1167–1175, 1999.

[37] N. Inamoto and H. Saito. Virtual viewpoint replay for a soccer match by view interpolation from multiple cameras. *IEEE Transactions on Multimedia*, 9(6):1155–1166, 2007.

[38] T. Jebara, C. Eyster, J. Weaver, T. Starner, and A. Pentland. Stochastics: augmenting the billiards experience with probabilistic vision and wearable computers. In *International Symposium on Wearable Computers*, pages 138–145, 1997.

[39] J. R. Jensen. *Introductory digital image processing: A remote sensing perspective*. Prentice Hall, second edition, 1995.

[40] R. Jiang, D. V. Jauregui, and K. R. White. Close-range photogrammetry applications in bridge measurement: literature review. *Measurement*, 41:823–834, 2008.

[41] B. Kamgar-Parsi. Line matching: solutions and unsolved problems. In *International Conference on Image Processing*, pages 905–908, 2001.

[42] H. Kato and M. Billinghurst. Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In *IEEE and ACM International Workshop on Augmented Reality*, pages 85–94, 1999.

[43] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 225–234, 2007.

[44] T. Komine and M. Nakagawa. Integrated system of white LED visible-light communication and power-line communication. *IEEE Transactions on Consumer Electronics*, 49(1):71–79, 2003.

[45] D. Kotake, K. Satoh, S. Uchiyama, and H. Yamamoto. A hybrid and linear registration method utilizing inclination constraint. In *IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 140–149, 2005.

[46] D. Kotake, K. Satoh, S. Uchiyama, and H. Yamamoto. A fast initialization method for edge-based registration using an inclination constraint. In *IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 239–248, 2007.

[47] E. L. Koua, A. M. Maceachren, and M. J. Kraak. Evaluating the usability of visualization methods in an exploratory geovisualization environment. *International Journal of Geographical Information Science*, 20:425–448, 2006.

[48] L. B. Larsen, R. B. Jensen, K. L. Jensen, and S. Larsen. Development of an automatic pool trainer. In *ACM SIGCHI International Conference on Advances in computer entertainment technology*, pages 83–87, 2005.

[49] V. Lepetit and P. Fua. Keypoint recognition using randomized trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:1465–1479, 2006.

[50] M. Lightfoot, G. Bruce, and D. Barber. The measurement of welding distortion in shipbuilding using close range photogrammetry. In *Annual Conference of the Remote Sensing and Photogrammetry Society*, 2007.

[51] M. Lim, D. N. Petley, N. J. Rosser, R. J. Allison, A. J. Long, and D. Pybus. Combined digital photogrammetry and time-of-flight laser scanning for monitoring cliff evolution. *The Photogrammetric Record*, 20:109–129, 2005.

[52] H. Lin, J. Gong, and F. Wang. Web based three dimensional geo referenced visualization. *Computers and Geosciences*, 25:1177–1185, 1999.

[53] Z. M. Lin, J. S. Yang, and C. Y. Yang. Grey decision-making for a billiard robot. In *International Confrence on Systems, Man and Cybemetrics*, volume 6, pages 5350–5355, 2004.

[54] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[55] G. S. M. Bertalmio, V. Caselles, and C. Ballester. Image inpainting. In *ACM SIGGRAPH*, 2000.

[56] H. G. Maas and U. Hampel. Photogrammetric techniques in civil engineering material testing and structure monitoring. *Photogrammetric Engineering and Remote Sensing*, 72:39–45, 2006.

[57] A. M. MacEachren and M.-J. Kraak. Research challenges in geovisualization. *Cartography and Geographic Information Science*, 28:3–12, 2001.

[58] S. Mahamud and M. Hebert. The optimal distance measure for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.

[59] E. Marchand and F. Chaumette. Stable visual servoing of camera-in-hand robotic systems. *Robotics and Autonomous Systems*, 52:53–70, 2005.

[60] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *British Machine Vision Conference*, pages 384–393, 2002.

[61] J. Matas, C. Galambos, and J. Kittler. Progressive probabilistic hough transform for line detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 554–560, 1999.

[62] S. Mathavan, M. R. Jackson, and R. M. Parkin. A theoretical analysis of billiard ball dynamics under cushion impacts. In *Institution of Mechanical Engineers*, 2010.

[63] N. Matsushita, D. Hihara, T. Ushiro, S. Yoshimura, J. Rekimoto, and Y. Yamamoto. ID CAM: a smart camera for scene capturing and ID recognition. In *IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 227–236, 2003.

[64] C. Matysczok, R. Radkowski, and J. Berssenbruegge. AR-bowling: immersive and realistic game play in real environments using augmented reality. In *ACM SIGCHI International Conference on Advances in computer entertainment technology*, pages 269–276, 2004.

[65] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60:63–86, 2004.

[66] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65:43–72, 2005.

[67] P. Moraa, P. Baldib, G. Casulac, M. Fabrisd, M. Ghirottia, E. Mazzinie, and A. Pescic. Global positioning systems and digital photogrammetry for the monitoring of mass movements: application to the ca' di malta landslide. *Engineering Geology*, 68:103–121, 2003.

[68] Y. Motokawa and H. Saito. Online tracking of guitar for playing supporting system by augmented reality. In *IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 243–244, 2006.

[69] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. Generic and real-time structure from motion using local bundle adjustment. *Image and Vision Computing*, 27:1178–1193, 2009.

[70] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Applications*, pages 331–340, 2009.

[71] T. Nakai, M. Iwamura, and K. Kise. Real-time retrieval for images of documents in various languages using a web camera. In *International Conference on Document Analysis and Recognition*, pages 146–150, 2009.

[72] T. Nakai, K. Kise, and M. Iwamura. Hashing with local combinations of feature points and its application to camera-based document image retrieval—retrieval in 0.14 second from 10,000 pages—. In *International Workshop on Camera-Based Document Analysis and Recognition*, pages 87–94, 2005.

[73] T. Nakai, K. Kise, and M. Iwamura. Use of affine invariants in locally likely arrangement hashing for camera-based document image retrieval. In *IAPR Workshop on Document Analysis Systems*, pages 541–552, 2006.

[74] T. Nakai, K. Kise, and M. Iwamura. Camera based document image retrieval with more time and memory efficient LLAH. In *International Workshop on Camera-Based Document Analysis and Recognition*, pages 21–28, 2007.

[75] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2161–2168, 2006.

[76] M. Ocana, L. M. Bergasa, and M. A. Sotelo. Robust navigation indoor using wifi localization. In *IEEE Internacional Conference on Methods and Models in Automation and Robotics*, pages 581–586, 2004.

[77] Y. Oike, M. Ikeda, and K. Asada. A smart image sensor with high-speed feeble ID-beacon detection for augmented reality system. In *IEEE European Solid-State Circuits Conference*, pages 125–128, 2003.

[78] OrbisGIS. http://orbisgis.cerma.archi.fr/.

[79] M. Ozuysal, M. Calonder, V. Lepetit, and P. Fua. Fast keypoint recognition using random ferns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(3):448–461, 2010.

[80] K. Pentenrieder, C. Bade, F. Doil, and P. Meier. Augmented reality-based factory planning - an application tailored to industrial needs. In *IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 1–9, 2007.

[81] J. Pilet, V. Lepetit, and P. Fua. Augmenting deformable objects in real-time. In *IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 134–137, 2005.

[82] J. Pilet and H. Saito. Virtually augmenting hundreds of real pictures: An approach based on learning, retrieval, and tracking. In *IEEE Virtual Reality*, 2010.

[83] J. Princena, J. Illingwortha, and J. Kittlera. A hierarchical approach to line extraction based on the hough transform. *Computer Vision, Graphics, and Image Processing*, 52:57–77, 1990.

[84] L. Quan and Z. Lan. Linear n-point camera pose determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:774–780, 1999.

[85] S. Ramalingam, S. K. Lodha, and P. Sturm. A generic structure-from-motion framework. *Computer Vision and Image Understanding*, 103:218–228, 2006.

[86] G. Reitmayr, E. Eade, and T. Drummond. Localisation and interaction for augmented maps. In *IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 120–129, 2005.

[87] J. Rekimoto. Matrix: a realtime object identification and registration method for augmented reality. In *Asia Pacific Conference on Computer Human Interaction*, pages 63–68, 1998.

[88] E. Rosten and T. Drummond. Machine learning for high speed corner detection. In *European Conference on Computer Vision*, pages 430–443, 2006.

[89] G. Schall, D. Wagner, G. Reitmayr, E. Taichmann, M. Wieser, D. Schmalstieg, and B. Hofmann-Wellenhof. Global pose estimation using multi-sensor fusion for outdoor augmented reality. In *IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 153–162, 2009.

[90] D. Schmalstieg and D. Wagner. Mobile phones as a platform for augmented reality. In *IEEE Virtual Reality Workshop on Software Engineering and Architectures for Realtime Interactive Systems*, pages 43–44, 2008.

[91] C. Schmid and A. Zisserman. Automatic line matching across views. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 666–671, 1997.

[92] J. Shi and C. Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[93] C. Shi-yi and L. Yu-bai. Error correcting cyclic redundancy checks based on confidence declaration. In *International Conference on TS Telecommunications*, pages 511–514, 2006.

[94] T. Sielhorst, T. Obst, R. Burgkart, R. Riener, and N. Navab. An augmented reality delivery simulator for medical training. In *International Workshop on Augmented Environments for Medical Imaging*, 2004.

[95] S. N. Sinha, J. michael Frahm, M. Pollefeys, and Y. Genc. GPU-based video feature tracking and matching. In *Workshop on Edge Computing Using New Commodity Architectures*, 2006.

[96] I. Skog and P. Handel. In-car positioning and navigation technologies: a survey. *IEEE Transactions on Intelligent Transportation Systems*, 10:4–21, 2009.

[97] S. M. Smith and J. M. Brady. SUSAN - a new approach to low level image processing. *International Journal of Computer Vision*, 23:45–78, 1995.

[98] E. Steiner, A. M. MacEachren, and D. Guo. Developing lightweight, data-driven exploratory geovisualization tools for the web. In *International Symposium On Spatial Data Handling*, pages 487–500, 2002.

[99] A. Suganuma, Y. Ogata, A. Shimada, D. Arita, and R.-i. Taniguchi. Billiard instruction system for beginners with a projector-camera system. In *ACM International Conference on Advances in Computer Entertainment Technology*, pages 3–8, 2008.

[100] S. Uchiyama, K. Takemoto, K. Satoh, H. Yamamoto, and H. Tamura. MR platform: a basic body on which mixed reality applications are built. In *IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 246–253, 2002.

[101] D. Wagner, T. Langlotz, and D. Schmalstieg. Robust and unobtrusive marker tracking on mobile phones. In *IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 121–124.

[102] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg. Pose tracking from natural features on mobile phones. In *IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 125–134, 2008.

[103] D. Wagner and D. Schmalstieg. ARToolKit on the PocketPC platform. In *IEEE Augmented Reality Toolkit Workshop*, pages 14–15, 2003.

[104] D. Wagner and D. Schmalstieg. ARToolKitPlus for pose tracking on mobile devices. In *Computer Vision Winter Workshop*, pages 139–146, 2007.

[105] P. Wang, Q. Ji, and J. Wayman. Modeling and predicting face recognition system performance based on analysis of similarity scores. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:665–670, 2007.

[106] T. Werner, F. Schaffalitzky, and A. Zisserman. Automated architecture reconstruction from close-range photogrammetry. In *International Symposium: Surveying and Documentation of Historic Buildings – Monuments – Sites, Traditional and Modern Methods*, 2001.

[107] Willowgarage. OpenCV 2.0. `http://opencv.willowgarage.com/wiki/`, 2010.

[108] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:1330–1334, 2000.

[109] Z. Zhang. Camera calibration with one-dimensional objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:892–899, 2004.

[110] S. Zhao, K. Nakamura, K. Ishii, and T. Igarashi. Magic cards : A paper tag interface for implicit robot control. In *ACM Conference on Human Factors in Computing Systems*, pages 173–182, 2009.

[111] Z. Zhou, J. Karlekar, D. Hii, M. Schneider, W. Lu, and S. Wittkopf. Robust pose estimation for outdoor mixed reality with sensor fusion. In *International Conference on Universal Access in Human-Computer Interaction*, pages 281–289, 2009.

Publications

Journal Articles (in English)

1. Hideaki Uchiyama and Hideo Saito, “AR supporting system for pool games using a camera-mounted handheld display,” *Advances in Human-Computer Interaction*, vol.2008, Article ID 357270, 2008.

Journal Articles (in Japanese)

1. Hideaki Uchiyama, Hideo Saito, Myriam Servières and Guillaume Moreau, “A geovisualization framework based on on-line geographical data matching between a map with intersections and its intersection database from GIS,” *The Journal of the Institute of Image Information and Television Engineers*, 64:563–569, April 2010.

2. Hideaki Uchiyama, Masaki Yoshino, Shin-ichiro Haruyama, Hideo Saito, Masao Nakagawa, Takao Kakehashi and Naoki Nagamoto, “A photogrammetric system based on visible light communications using light markers,” *The Journal of the Institute of Image Electronics Engineers of Japan*, 38:703–711, September 2009.

3. Hideaki Uchiyama and Hideo Saito, “AR display for strategy based pool supporting system from handy camera,” *Transactions of the Virtual Reality Society of Japan*, 12:159–169, June 2007.

International Conferences

1. Hideaki Uchiyama, Julien Pilet and Hideo Saito, “On-line document registering and retrieving system for AR annotation overlay,” In

Proceedings of the 1st ACM Augmented Human International Conference, April 2010.

2. Hideaki Uchiyama and Hideo Saito, “Augmenting text document by on-line learning of local arrangement of keypoints,” In *Proceedings of the 8th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 95–98, October 2009.

3. Ryo Mitsumori, Hideaki Uchiyama, Hideo Saito, Myriam Servierès and Guillaume Moreau, “Change detection based on SURF and color edge matching,” In *Proceedings of the Invited Workshop on Vision and Control for Access Space (in 9th Asian Conference on Computer Vision)*, September 2009.

4. Hideaki Uchiyama, Hideo Saito, Myriam Servierès and Guillaume Moreau, “AR GIS on a physical map based on map image retrieval using LLAH tracking,” In *Proceedings of the 3rd International Conference on Virtual and Mixed Reality (in 13th International Conference on Human-Computer Interaction) (LNCS 5622)*, pages 128–135, July 2009.

5. Hideaki Uchiyama, Hideo Saito, Myriam Servierès and Guillaume Moreau, “AR city representation system based on map recognition using topological information,” In *Proceedings of the 11th IAPR Conference on Machine Vision Applications*, pages 382–385, May 2009.

6. Naoyuki Yazawa, Hideaki Uchiyama, Hideo Saito, Myriam Servierès and Guillaume Moreau, “Image based view localization system retrieving from a panorama database by SURF,” In *Proceedings of the 11th IAPR Conference on Machine Vision Applications*, pages 118–121, May 2009.

7. Hideaki Uchiyama and Hideo Saito, “Rotated image based photomosaic using combination of principal component hashing,” In *Proceedings of the 3rd IEEE Pacific-Rim Symposium on Image and Video Technology (LNCS 5414)*, pages 668–679, January 2009.

8. Hideaki Uchiyama, Hideo Saito, Vivien Nivesse, Myriam Servierès and Guillaume Moreau, “AR representation system for 3D GIS based on camera pose estimation using distribution of intersections,” In *Proceedings of the 18th Annual International Conference on Artificial Reality and Telexistence*, pages 218–225, December 2008.

9. Hideaki Uchiyama, Masaki Yoshino, Hideo Saito, Masao Nakagawa, Shin-ichiro Haruyama, Takao Kakehashi and Naoki Nagamoto, “Photogrammetric system using visible light communication,” In *Proceedings of the 34th Annual Conference of the IEEE Industrial Electronics Society*, pages 1771–1776, November 2008.

10. Hideaki Uchiyama and Hideo Saito, “AR display of visual aids for supporting pool games by online markerless tracking,” In *Proceedings of the 17th Annual International Conference on Artificial reality and Telexistence*, pages 172–179, November 2007.

11. Hideaki Uchiyama and Hideo Saito, “Position estimation of solid balls from handy camera for pool supporting system,” In *Proceedings of the 1st IEEE Pacific-Rim Symposium on Image and Video Technology (LNCS 4319)*, pages 393–402, December 2006.