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

Augmented Reality on Geometrically Changeable Paper

March 2013

Graduate School of Science and Technology
Keio University

Sandy Eggi Martedi

Abstract

This research explores a technique for enhancing the integration between physical paper and digital information using augmented reality. Conventional augmented reality systems overlay virtual information onto planar objects such as paper. A piece of paper can be rotated and translated in front of a camera in order to view virtual contents in 6 degrees of freedom. However, people usually handle a piece of paper by folding or bending. For instance, people fold a piece of paper for holding it easily or bend it to follow the shape of their hands. Conventional augmented reality systems do not consider the change in the geometric property of paper such as folding and bending as natural interaction. Therefore, it is necessary to implement such interactions in order to enhance augmented reality.

Firstly, this work includes the modeling and recognition of folding, bending and cutting-based interactions on physical paper by applying a matching method. This work proposes an automatic recognition of the folding applied to physical papers and the transition between folded and planar condition. Secondly, the folding is extended into a bending interaction. Thirdly, regions recognition on paper is extended to allow the user to cut a piece of paper and track the pieces independently.

The system setup is then extended using projector-camera setup to allow the user to view the visualization directly on physical papers. In this case, the system can be implemented in larger area. By using the random dot marker technique, automatic content alignment is proposed. It allows the arbitrary movement of projector, camera and physical paper. As a result, it is not necessary to fix the projector, the camera and the paper beforehand and the time-consuming pre-calibration procedure can be avoided.

The proposed method is applied in order to realize the visualization of geographical information on physical paper maps. User can fold, bend, and cut a piece of paper map for interacting with virtual contents. A system architecture for building augmented maps applications that retrieves the geographical information from the Internet is presented. Using the proposed system architecture, the paper map and the virtual contents can be retrieved on demand so that the augmented maps application of any location can be made. Moreover, in order to support the user mobility and recent devices, the implementation of augmented maps on mobile phones is explored. Furthermore, the interaction of augmented maps is also explored by realizing pointing and tapping gestures.

Acknowledgements

I would like to express my gratitude to Prof. Hideo Saito for supervising me all these years. He supervised me since I started research as research student until I completed my study. He always encourage me for achieving many publications and grabbing many opportunities in conferences.

I would like to thank to Prof. Bruce Thomas from University of South Australia. He always suggests interesting ideas in our collaboration project. He gave many valuable comments in order to improve my thesis.

I would like to express my appreciation to the thesis committee members: Prof. Ken-ichi Okada and Prof. Yasue Mitsukura for kindly reviewed this thesis. All of their comments and suggestions are very important factors for completing this thesis.

Special thanks to Prof. Maki Sugimoto who has been a nice mentor, collaborator and friend. To my greatest inspirations, Dr. Hideaki Uchiyama, I thank him for teaching me so much knowledge and showing me overwhelming passion towards research. My thanks is also for Prof. Guillaume Moreau, Prof. Myriam Servières and Dr. François de Sorbier, who kindly gave me helpful advices as my collaborators.

I would like to thank to Keio University and my sponsor MEXT for supporting my research life in Japan.

I would like to say thank you to all of my lab mates in Hideo Saito Laboratory since I joined the lab in 2007. To all friends who kindly help me during my study: Dr. Yuko Uematsu, Dr. Tomoaki Teshima, Dr. Julien Pilet, Dr. Vincent Nozick, Dr. Songkran Jarusirisawad, Dr. Chutisant Kerdvibulvech, Dr. Yuji Oyamada, Dr. Ismael Daribo, Sebastien Callier, Dao Hu Hung and Dissaphong Thachasongtham. I specially thank to my best friends who always be my great support: Seiji Suzuki, Hiroshi Tanaka, Takayuki Nakamura, Yusuke Nakamura and Erwin Harahap. My thanks to all of young friends and fellows in Hideo Saito Laboratory.

Finally, I would like to thank my family and my friends for their support during my studies.

February 2013
Sandy Eggi Martedi

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivations

This thesis discusses a technique for enhancing the functionality of physical paper. In recent years, the digital technology has improved the way to transfer and store information. The digital media has partially replaced physical paper because the information in digital media is instantly and easily replicable and transmittable. On the other hand, in order to copy and transmit information on physical paper instantly, conversion into digital version is necessary. Researchers have explored technology so called augmented reality for merging the benefits of physical paper and digital media. Using augmented reality, one can view virtual information on top of physical paper via video see-through device.

This chapter initially discusses about paper in general and the recent digital paper. The integration between physical paper and digital content is then discussed. Augmented reality applications using physical paper and digital content as the main motivations in this thesis are described. In order to realize those applications, some problems need to be solved. This thesis proposes solutions for those problems in upcoming chapters.

1.1.1 History of Paper

Paper was invented in 105 AD in China during the Han Dynasty and spread to the west via Samarkand and Baghdad. The word ”paper” is taken from the Ancient Greek word papyros to mention the Cyperus papyrus plant. The papyrus was used in ancient Egypt and other Mediterranean cultures for writing before the making

of paper in China (see Figure 1.1).

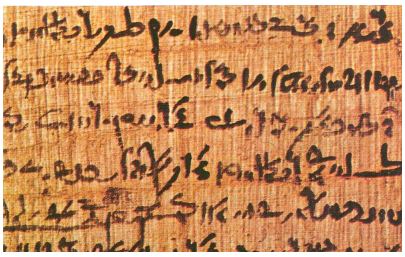


Figure 1.1: Papyrus fragment. http://commons.wikimedia.org/wiki/Papyrus

For many years, paper has been used for making books through handwriting or printing process. Even though other materials such as plastic and metal are difficult to age that is suitable for keeping information longer, paper is widely used for making books due to the lightness and thinness.

Electronic or digital version of paper has been invented and it has been sold in the market widely in recent years. A digital paper reader combines electronic paper with a computation and storage device to provide a portable device for users. Digital paper reader is an electronic device that can display text and images in certain intensity and color similar to physical paper [6, 91]. The reader can read the contents even in outdoor environment as if reading a real book. Unlike the ordinary display, digital paper does not reflect the light as spectral highlights that blocks the text. As a result, the displayed content is readable in a wide range of lighting condition. Digital paper will become an alternative to physical books in the future.

1.1.2 Paper-based Augmented Reality

Despite the electronic paper and e-book reader having attracted a lot of attention, physical paper can not be replaced entirely with digital paper. One of the reasons is digital paper requires a specific device for viewing. Moreover, a part of readers are more accustomed to physical paper. Readers interact with papers and books naturally by folding and flipping. Generally, the user reads on paper faster than on screen, especially for long content reading [36]. In addition, conventional distribution of newspapers, magazines, and books has been steadily established. Therefore, paper as printing media will exist even in the future.

Instead of replacing physical paper with digital one, there are efforts to enhance the functionality of physical paper by adding digital contents [18, 31, 46,

84]. Paper-based augmented reality is one of them. It is an application for displaying digital information virtually on top of physical paper using video see-through device as illustrated in Figure 1.2.

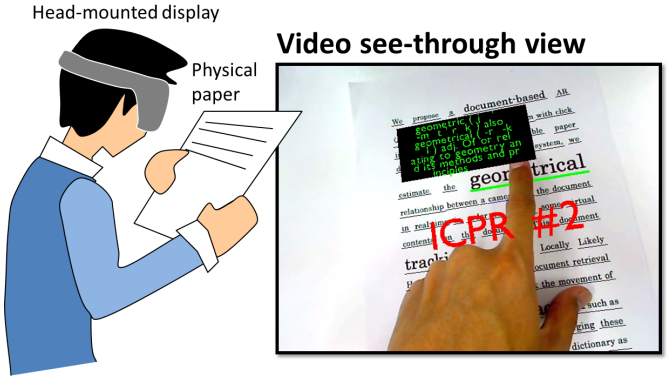


Figure 1.2: Paper-based augmented reality. Virtual labels are displayed over the physical paper.

Paper-based augmented reality has been widely used in industries [28, 45]. Many products contain an augmented reality marker in their packages in order to give more interactive information. Magazines and books put also the marker in the cover or inside the contents to provide multimedia contents. In those fields, paper-based augmented reality can offer many benefits. Unlike conventional media that contains predefined and static contents, augmented reality uses a camera that brings interactivity and customizable multimedia contents.

Meanwhile, the quality of display for augmented reality is also improving [34]. Video see-through displays such as smartphones, tablets and head mounted displays are the current technology for displaying augmented reality contents. All of them perform well and they are already available in the market. Augmented reality has been used for supporting the realization of ubiquitous computing because augmented reality can visually merge the real world with the virtual world [8, 35, 56]. In ubiquitous concept, the user can access system using every day object and activities without even realizing it [93]. Therefore, recognition of every object in the real world and augmentation of information onto it becomes the major issues. Surface and object recognition is the key technologies in order to realize this idea.

Paper is one the surfaces that can be easily found everywhere. Books, magazines, newspapers or maps contain much potential information for recognition. In the near future, we can imagine bringing only one physical book and read

many novels from inside its covers. One can also write or draw something on a white piece of paper and the next words or strokes will be recommended virtually. Furthermore, we can imagine bringing a piece of paper map on the street and geographic information from the Internet such as 3D models will pop-up as illustrated in Figure 1.3.

Figure 1.3: Augmented maps application. Displaying 3D models on top of paper map. The image is provided by Dai Nippon Printing company, Japan.

There are two current trends in augmented reality field that can achieve those future images. Firstly, one could view numerous virtual contents on video see-through display attached on the human head such as head mounted display or near eye display. New versions of glasses-like display have been produced recently. Additionally, research on contact lens display has been started. In this case, the difficult part in this trend is improving the current registration and tracking techniques, developing novel applications and enriching the content. In order to entirely merge visually the real world with virtual world, the recognition and tracking technique are the main issues.

Secondly, one could view information directly on any surface in the real world without using video see-through display. In this case, any surface can act as the display. We can imagine that virtual information pop out in any object and visible to our bare eyes. For instance, the price of a product can be displayed on the package in real time. The focus of this direction is the recognition and how to display the information on the physical surface.

1.2 Problems

In order to realize natural interaction in paper-based augmented reality, the folding, bending and cutting actions on physical paper should be recognized. That recognition requires the modeling of the geometrical property of the paper and the information extraction from the paper. In contrast to ordinary paper-based augmented reality, modeling and detecting a geometrically changeable paper requires continuous detection and shape reconstruction in real time. Moreover, on monocular augmented reality setup, the reconstruction must be performed using limited number of features.

Furthermore, in order to realize augmented reality in a wide range of surface types and locations, the current setup is extended into a projector-camera setup. The major issue in the projector-camera setup is the alignment of the projected content on the physical surface in real time. Alignment method technique usually requires the time-consuming calibration process by calculating the relative position between camera, projector and surface beforehand. The automatic alignment becomes more difficult especially when the camera, the projector and the surface are moving arbitrarily that generally occurs on mobile setup and surveillance setup.

1.3 Contributions

The main contributions in this thesis are listed as follows:

- Paper manipulation for enriching the functionality of physical paper in augmented reality is explored. The folding-, bending-, and cutting-based interaction are implemented.
- Feature-based alignment method for projector-camera setup by partially projecting the surface content is presented.
- A system architecture for retrieving maps and virtual contents from the Internet is proposed in order to make on demand augmented maps application. The system architecture is also used in order to implement the first and second contribution.

1.4 Thesis Organization

The rest of this thesis is organized as follows: related works of this research are explained in Chapter 2. Chapter 3 describes the recognition of the changeable geometrical property of paper including folding, bending, and cutting. Chapter 4 explains the implementation of alignment method for projector-camera setup using keypoint matching method. Chapter 5 describes the system architecture for building the augmented maps application using geographical data from the Internet and used in the proposed method explained in Chapter 3 and Chapter 4. Chapter 6 concludes this thesis with a summary and discussion of future improvement.

Chapter 2

Related Works

This chapter discusses the state of the arts in enhancing physical paper using augmented reality. Firstly, paper-based augmented reality investigations are stated. Augmented reality using projector-camera setup are then explained. Related works on augmented reality applications and system architecture that deal with geographic information are described. Furthermore, the explanation of the random dot marker technique is presented because this thesis applies this technique in order to solve the problems mentioned in the previous chapter. Finally, the position of this thesis is illustrated to give a broad picture about the problem domain in this thesis compared to other relevant works.

2.1 Paper-based Augmented Reality

The geometrical properties of paper can be changed in order to add interactivity for augmented reality. Firstly, paper-based augmented reality methods are described briefly.

2.1.1 Planar

Paper-based augmented reality can be implemented using fiducial marker, texture or geometric features extracted from the paper. Marker-based augmented reality initially appeared in the form of color coded pattern in 1995 [75] and it was popularized by ARToolkit [42] and ARTag [27]. Generally, it uses a planar paper as a fiducial marker as illustrated in Figure 2.1. Template matching is usually used for detecting the marker. The camera pose can be estimated using detected corners in

the border. Using the estimated camera pose, the virtual object can be overlaid as illustrated in Figure 2.2.

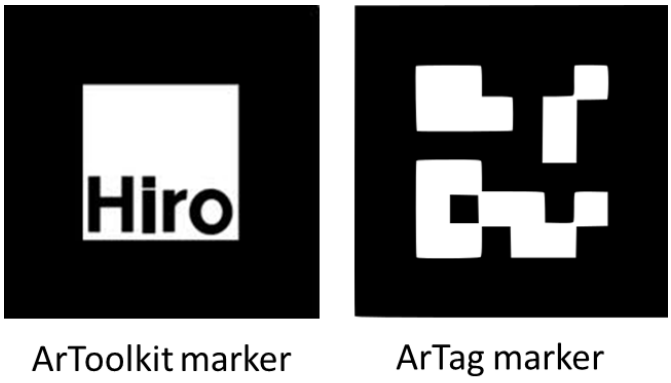


Figure 2.1: Fiducial markers. A fiducial marker is composed a pattern for recognition and a rectangle for camera pose estimation.

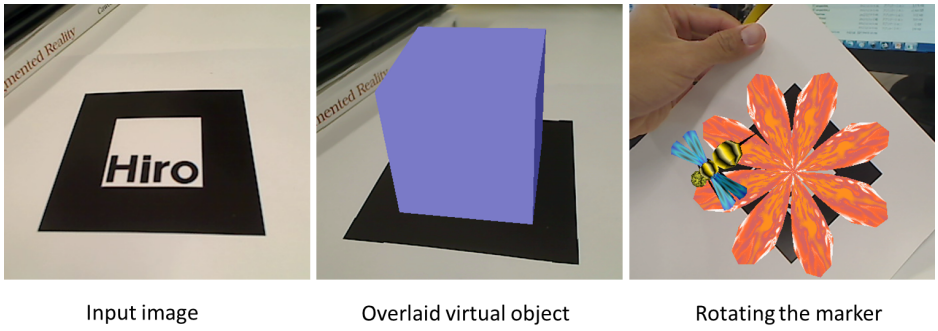
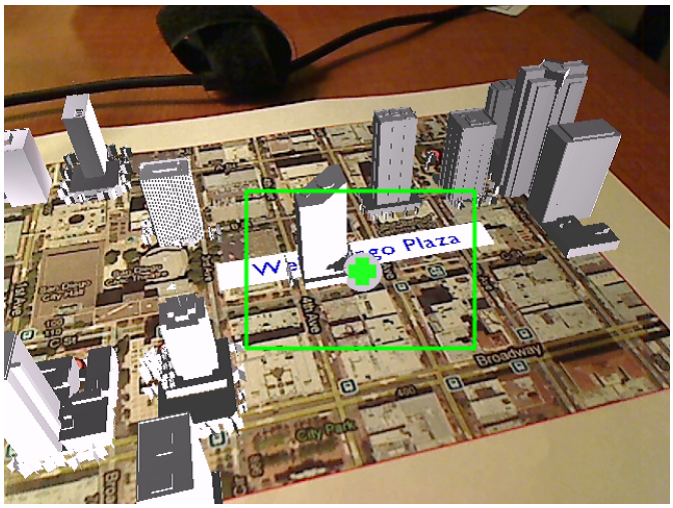


Figure 2.2: ARToolkit-based application. The example visualizations on ARToolkit. The interaction is limited to viewing, translating, and rotating the planar marker.

Texture-based augmented reality replaces the fiducial marker with ordinary images that contain rich texture as features. It uses feature and descriptor extraction and matching such as SIFT [55], randomized tree [50], SURF [11], Random Ferns [68], BRISK [53], and CARD [9]. Many applications have been developed using such techniques because it uses an ordinary image as marker that looks natural for users as illustrated in Figure 2.3.



©2011 Google - Imagery ©2011 Google, Map data ©Google

Figure 2.3: Satellite map as texture for tracking. An augmented map application that uses Random Ferns [68] for tracking a piece of paper map.

2.1.2 Deformation

The planar paper-based augmented reality is extended by deforming the paper. In this case the geometrical property of the paper is changed. The deformable paper modeling was introduced initially by Kergosien et al. [43]. They modeled a piece of paper in mathematical term and simulate the bending and creasing interaction using boundary points. Bo and Wang used geodesic of paper instead of boundary points to estimate the shape more accurately [16].

Paper interaction such as holding, collocating, collating, flipping, and rubbing have been demonstrated in PaperWindows [40]. In technical point of view, PaperWindows facilitates natural interactions on physical paper by recognizing the shape of the paper using multiple cameras based vision tracking system. The interaction in PaperWindows is illustrated on Figure 2.4. More detailed user study on manipulating surfaces including paper for interaction have been performed by Lee et al. [49]. They showed that folding, bending and stretching gestures are intuitive for users in order to build a deformable interactive device. Another technical approach attached LED markers on the paper and applied LED tracking to detect the shape of a paper [48].

Modeling and detection of deformable surfaces including paper and clothes

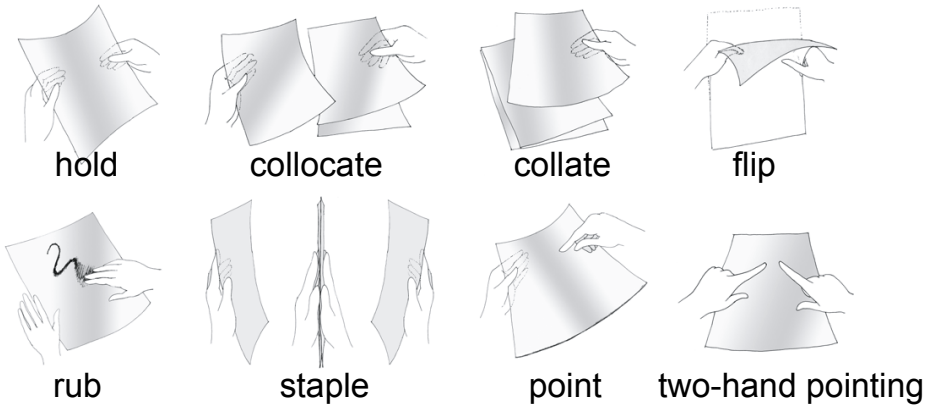


Figure 2.4: Gestures in PaperWindows. The gestures that related with the change of geometric properties are hold, collocate, collate, flip, and staple.

are being extensively explored in many applications [19, 32, 70, 71]. Deformable surface is usually modeled by a collection of triangles [20, 29, 67, 72, 79, 80, 81].

Recent method has succeeded in detecting deformation of surface and simultaneously computing the camera pose and hence it is able to overlay virtually 3D model on top of the surface [23]. Another approach separated the deformed part and the planar part in order to improve the accuracy of existing deformable detection method [62]. RANSAC method also can be applied to detect deformable surface as proven by Tran et al. [87]. Interactive folding for origami application is introduced by Zhu et al. using texture on the paper [96]. They recorded features in a piece of paper into overlapping regions that will trigger an action when the user folds the paper and the camera captures and detects any visible region. Similarly, Gesture recognition on deformed paper has been explored by Ziegler and Belongie [97] by applying deformation model proposed by Pilet et al. [72].

2.1.3 Separation

Performing separation or cutting action on augmented reality requires detection of separated regions and the separation line. Taylor et al. have investigated the cutting on a deformed material such as clothes or paper [86]. Their work is based on 3D deformation that handles reconstruction of material by registering the material using local features. Their work uses the structure from motion that uses multiple frames.

Recognizing a separation line has been explored in several works. Bergig et al.

have developed an application for augmented reality that recognize handwriting al. [13]. Their method recognizes the 2D drawing and reconstructs its 3D shape. Another investigation on arbitrary shape for planar registration is done by Hagbi et al. [33]. They used a classification method for searching region template in database. Donoser et al. proposed a method for tracking regions using a region detector method so called maximally stable extremal region (MSER) [24].

Nishizaka et al. presented a separating technique by detecting two divided parts after the separation action [65]. They use random Ferns [68] as natural feature to track the planar objects. They prepare ferns database and remove the detected part in database during separation and track each region individually. However, they only depend on features inside each divided part that reduces the tracking accuracy because the features decreases as well. Recent investigation so-called progressive shape models considers the deformation and topological changes of 3D object simultaneously using learning method on temporal meshes [51]. The method applied a geometric estimation without considering the texture or photometry information. Taking into account the texture information may increase the accuracy of the method.

2.2 Alignment Method for Projector-camera Setup

In order to allow users to move freely, projector is used instead of monitor display. There are several types of projector-camera system based on the configuration of the projector and the camera. Conventional setup uses static configuration of projector and camera. Projector and camera are fixed firmly at particular configuration and the position changes are not allowed. However, with the help of a depth camera, interactive applications can be developed using this configuration [12]. Other type of systems allow projector and camera to move together [37]. However, the relative position between projector and camera should be static. These kinds of systems are usually developed for wearable system.

Recent systems do not restrict the configuration between projector and camera. Such system can compensate the change of the relative position between projector and camera during run time using automatic calibration or alignment method. Generally, projector-camera systems use camera or sensors for registering a movable surface and aligning projected contents. Marner and Thomas proposed a method using a sensor attached on a digital brush for coloring 3D objects by projecting colors on the objects [57]. Similarly, Lee et al. put a sensor on each corner of a surface and automatically register the position of the surface [48]. Sensor

based method is accurate and fast but sensors must be attached on the surface beforehand.

Ehnes and Hirose used an AR marker for tracking a movable surface [25]. Takao et al. detected surface corners and corrected the projection using a homography matrix [85]. Recently, Audet and Okutomi proposed a method that uses a surface texture and projected pattern for aligning virtual contents [10]. Similarly, a feature based approach using border tracking was proposed by Leung et al. [52].

A combination of a particle filter method and a motion sensor method has been proposed [83]. This method calibrates the camera and projector in each frame so that the mapping between the camera coordinate and projector coordinate can be estimated during runtime.

In order to compute the projector position relative to camera, active registration is applied. Active registration involves the projection of known patterns before projecting the content. The active registration using binary coded patterns [98] and the simultaneous projection of structured light pattern [7] are investigated. Recently, McIlroy et al. have implemented the alignment method using projected dots from Kinect [60].

2.3 System Architecture

2.3.1 Application

Paper maps are used in augmented reality as marker because they contain plenty features such as points, polygons, and textures that are potential for registration. Augmented reality application that uses paper maps as marker is called augmented maps. There have been some researches on utilizing ARToolkit for developing augmented maps application. Hedley et al. combined the augmented reality with geographical data visualization [39]. They also equipped the system with fingertip detection and interaction. Bobrich et al. also used the ARToolkit to track a planar map [17]. McGee at al. developed a collaboration system for augmented maps by placing four AR markers near the printed map [59]. The user can draw annotation on a piece of paper map by using digital pen and share their modifications with the other users. However, ARToolkit is not robust against occlusions. The virtual contents are also limited to predefined data. On top of that, the marker obstructs the appearance of the map.

In order to keep the appearance of the system and the map, recent investigations replaced fiducial marker with the map itself using map features for tracking.

Recently, a fast keypoint detection using random ferns had also been developed for map tracking [68]. Another approach used mutual information between two map images for tracking [22]. Reitmayr et al. developed augmented maps used natural features tracking in table-top system equipped with a projector [74]. Their system could project the additional information on top of the map using projector. Moreover, the user can select information using PDA device as a pointer. Similarly, Rohs et al. used patches in a piece of paper map as the visual descriptor for detection and tracking [77]. Furthermore, they used mobile devices for displaying additional information.

2.3.2 Virtual Contents

Besides tracking as the main issue, researchers have drawn attentions to collaboration on augmented reality. Generally, augmented reality application contains predefined 3D models which are not reusable for another application. Instead of creating 3D models from scratch, some AR applications retrieve them from the Internet. For instance, AR sights system allows users to download available markers from its website and 3D models from Google 3D Warehouse [1].

Live videos augmentation on aerial map was explored by Kim et al. [44]. They also applied the real lighting condition estimated directly from the maps image. Similarly, Bing from Microsoft integrates maps, panorama pictures, and live videos submitted by users in [3]. Both works use collaboration among users. However, they only merge on the digital map digitally in the application instead of using paper maps and augmented reality setup.

Morrison et al. used natural features to track a printed Google map and visualized additional information on top of it [63]. However, they did not augment the 3D city models onto the printed map. Similarly, Paelke et al. integrated a piece of paper map and additional information on mobile devices [69]. Gruber et al. provided a dataset for tracking using city models from Google 3D warehouse [30]. However, their work only covered virtual contents preparation and omitted map detection and tracking.

2.4 Random Dot Marker Technique [90]

Random dot marker technique is an extension of the Locally Likely Arrangement Hashing method [64]. LLAH and random dot marker technique use the geometric relationship between feature as the descriptor. The descriptor is stored into a

hash table for fast access during matching in runtime. The benefit of LLAH is the robustness against occlusion and the fast matching performance thanks to the hash table. Random dot marker extends the advantage of LLAH. It updates the hash table during runtime that makes the tracking more stable and capable to track the planar surface even in the extreme camera orientation. Random dot marker technique is also capable to detect multiple markers.

2.4.1 Initialization

The random dots are generated as the keypoints for matching. These keypoints are then stored to a file that can be loaded when the system starts.

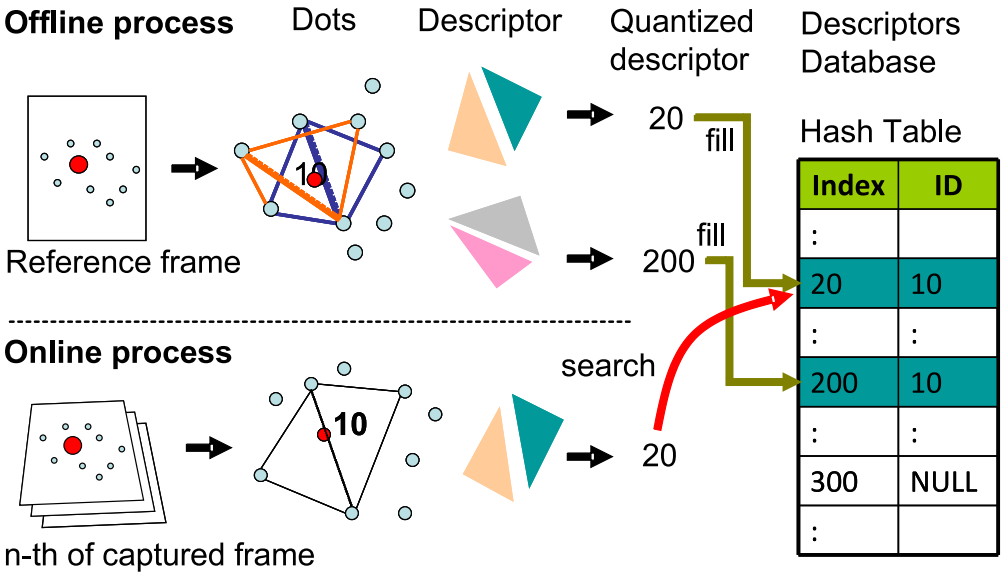


Figure 2.5: Keypoints matching. There are two main processes on the matching. Offline process estimates descriptors of all dots/keypoints and store them to a descriptor database as the index of dots id. The online process extracts dots from captured image and compute the descriptor of the dot followed by searching the dots id based on the descriptor.

Keypoints are matched in the initialization step as illustrated in Figure 2.5. In offline phase, a descriptor database is created from a text file that contains the

location of buildings (dots) in geographic coordinate. The descriptors of a point are computed by estimating its relationship to the neighboring points.

In online phase where a camera captures the printed paper, dots are extracted using color detection. The descriptor for each dots are then calculated followed by matching the calculated descriptors with the descriptors in the database. At this stage, many matches are established and the planar is detected.

2.4.2 Tracking

After the matches are established in the initialization, planar tracking is started on succeeding frames. The descriptors of detected dots are inserted into the database. The updated database increases the possibility of detection because the previous information are kept. The random dot marker technique is extended so in order to detect deformable surfaces [88].

2.5 Thesis Position

The trend in paper-based augmented reality researches can be mapped into two axis directions as illustrated in Figure 2.6. There are two axes: the shape complexity axis and the space axis in augmented reality. The horizontal axis describes the shape complexity in the registration method. The left side represents the simplest shape that can be registered in augmented reality that is planar. Recent works in augmented reality is able to register complex shapes such as deformed and arbitrary shapes. This thesis focuses on the shape between planar and deformed shape: folded, bended and separated surface (cut surface).

The vertical axis describes the space scale that is built as the result of the augmented reality setup. The desktop setup such as web camera, monitor and desktop computer creates a private space because it is limited to limited users. Moreover, the video see-through display such as head mounted display or glasses-like display create private space because the user will only see the virtual information privately. The highest position in the vertical axis creates a public space. In this area, the system can be used for multiple users in larger area. This thesis uses projector-camera setup which can be categorized in this area of research.

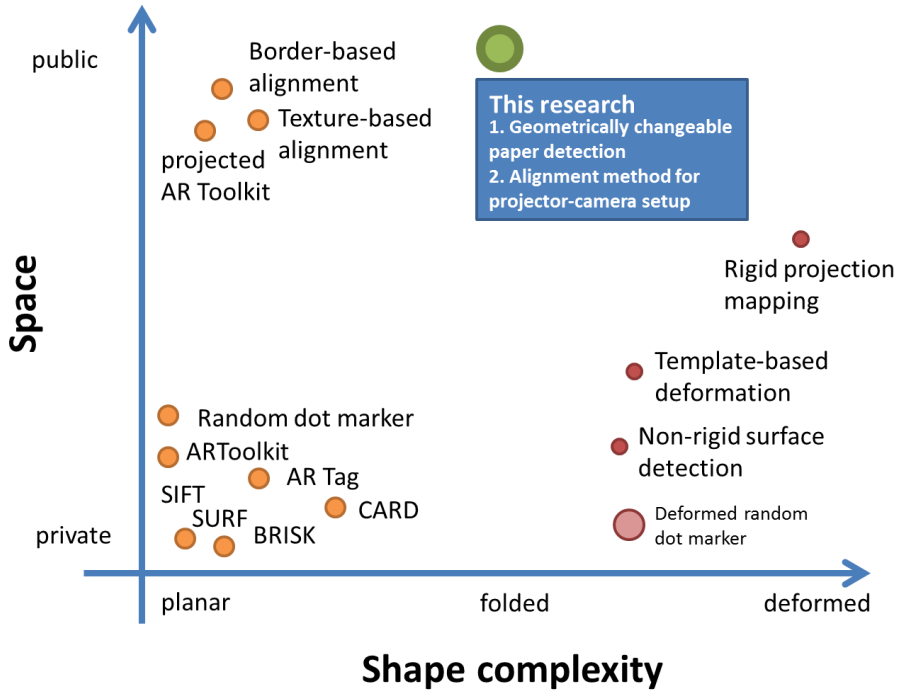


Figure 2.6: Thesis position. This thesis is mapped in the middle of the shape complexity axis. Based on the space axis, this thesis is located in the public space where multi users are allowed to view the projected contents together.

Chapter 3

Geometrically Changeable Paper Detection and Tracking

Conventional paper based augmented reality applications usually consider only planar papers as registration target. However, typical use of a piece of paper such as folding may change its geometrical properties which contains multiple planes. Paper can also be bent according to how human hold the paper which may alter the planarity of the paper. In addition, a paper can also be topologically changed by separation or cutting. Folding, bending, and cutting can be regarded as natural actions on paper. Therefore, it is important to facilitate those geometrical changes for augmented reality in order to enhance the functionality of paper.

The major issue for achieving folding, bending, and cutting in augmented reality is recognizing them. The typical augmented reality setup uses a video see-through HMD (monocular-based system) which make the problem difficult. In monocular-based system, the problem can be solved by recovering the surface shape of a reference plane from a single view image. This thesis applies distribution of keypoints for matching in order to detect the geometrically changeable paper. Keypoints are matched between an input image and a reference paper. Using these correspondences, a single plane can be divided into multiple planes on folded and separated paper or deformed into bended plane.

This chapter discusses three geometrical changes on paper: folding, bending and cutting. Firstly, each model is presented followed by the technical implementation. The scenario of each model in the application is then described. Finally, technical implementation is then evaluated especially for folding and cutting.

3.1 Folded Surface

In order to overlay a virtual object onto a surface, it is necessary to estimate the relative pose of the surface to a camera. When a surface is folded along a single line, it consists of two planes. In this case, it is necessary to estimate the relative pose of those two planes to the camera.

3.1.1 Folding Model

Folding can be modeled in two ways according to folding line position. The former model is the constrained model where the folding line is defined as a line that divides the paper vertically and horizontally. The latter model is the relaxed constrained model where the folding line is not limited to vertical and horizontal line but also diagonal line.

Constrained Model

A piece of paper can be folded in two directions based on the folding line: left-right folding and top-bottom folding as illustrated in Figure 3.1(a). Based on the folded shape, the folding ways can be classified into two types: mountain folding in Figure 3.1(c) and (e), and valley folding in Figure 3.1(b) and (d). The former is the case that the angle between two planes is more than 180 degrees. The latter is the case of less than 180 degrees.

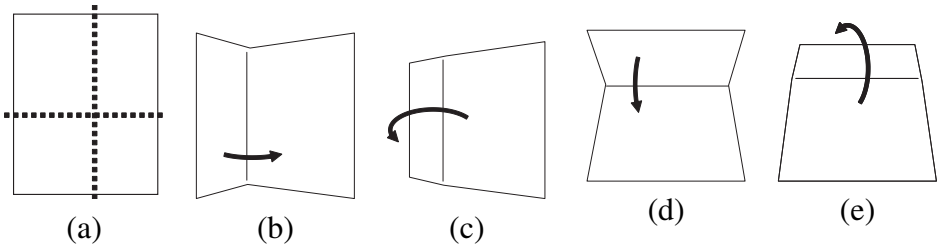


Figure 3.1: The constrained folding model. (a) A surface is divided into two planes separated by either horizontal or vertical folding line at arbitrary positions. (b) Left-right valley folding. (c) Left-right mountain folding. (d) Top-bottom valley folding. (e) Top-bottom mountain folding.

Relaxed Constrained Model

A piece of paper can be folded diagonally. To facilitate such kind of folding, the constraint model is relaxed so that the paper can be folded in arbitrary position as illustrated in Figure 3.2. A folded paper is then defined as two polygonal planes that are connected in the same border line. The border of each plane in the folded surface is composed by four or more points.

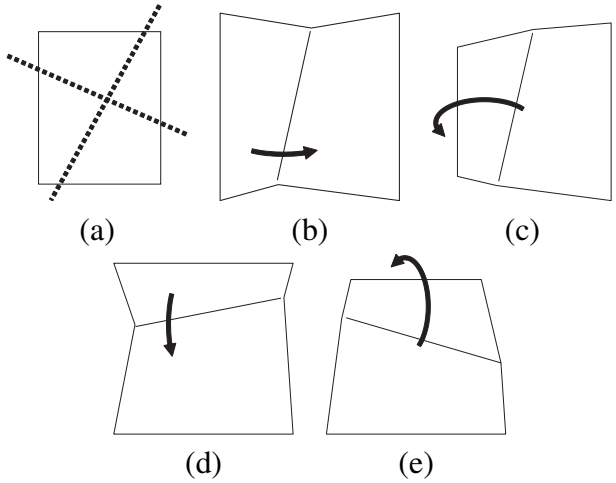


Figure 3.2: Relaxed constrained folding model. (a) The folding position constraint is relaxed. A surface is divided into two planes separated by an arbitrary folding line at arbitrary positions. (b) Left-right arbitrary valley folding. The angle between two planes is less than 180 degrees. (c) Left-right arbitrary mountain folding. The angle is more than 180 degrees. (d) Top-bottom arbitrary valley folding. (e) Top-bottom arbitrary mountain folding.

3.1.2 Procedure Overview

In order to detect a plane in the proposed model, distribution of keypoints are used as features. The plane detection is defined as matching between a keypoint in the captured image with the corresponding keypoint in reference plane in database. In a pre-processing phase, a keypoint and descriptor database of reference planes are prepared. The keypoint database consists of a set of keypoints in each reference plane, which represents 2D distribution of keypoints. Each keypoint is related with its descriptors. Because the position of folding lines is not pre-defined,

segmented planes are not stored beforehand. Instead, the segmented planes are automatically estimated in an online process.

Figure 3.3 represents the flowchart of the proposed folded surface detection. First, 2D correspondences between an input image and the reference plane (in the reference coordinate system) is established by using descriptor based keypoint matching. From the correspondences, multiple planes are detected by iterative geometric verifications. Next, a folding direction and folding line from the positional relationship of the planes are computed. The output of foldable surface detection is the two edge points of the folding line on the reference coordinate system.

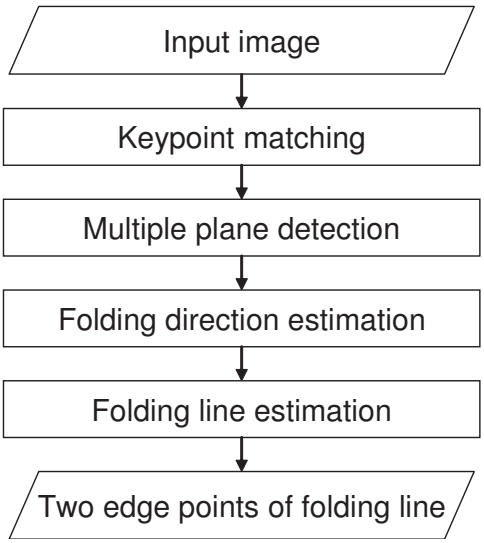


Figure 3.3: Procedure overview.

3.1.3 Multiple Plane Detection

To perform a multiple plane detection, a method similar to a sequential RANSAC-based approach is applied [92]. Firstly, two planes of a folded surface are detected. From an input image, keypoints are extracted. For each keypoint, the correspondence with reference planes is established using the descriptor based keypoint matching. From all the correspondences, the first homography is computed for detecting the first plane from the reference to the image with RANSAC.

By thresholding the distance between each projected keypoint and its nearest keypoint in the image (3 pixels is set), keypoints in the first plane from the

correspondences are excluded. For the rest of the correspondences, the second homography is computed with RANSAC to detect the second plane.

Folding Direction and Folding Line Estimation

In order to estimate left-right or top-bottom folding of the paper, the relative position between two planes is simply used. Suppose two planes are detected, the center of each plane on the reference plane is computed to make a vector connecting the centers. If the direction of the vector is horizontal, the folding direction is set to left-right. Otherwise, the folding direction is set to top-bottom. The estimated direction is utilized for folding line estimation in edge point estimation. The user can dynamically change the folding position and the directions during run time (online).

On a folded surface, two planes are segmented by a folding line. In order to achieve a natural augmentation on the surface, it is necessary to estimate the exact location of the folding line. In this case, the folding line is used as a separator between two planes. It is also used to divide the virtual contents according to the area of each plane.

As a result of multiple planes detection, two homographies between an input image and a reference plane are produced. This means that the reference plane is detected twice in the image as illustrated in Figure 3.4(a). In order to have the exact folding line, the intersection line of these two planes needs to be computed as illustrated in Figure 3.4(b).

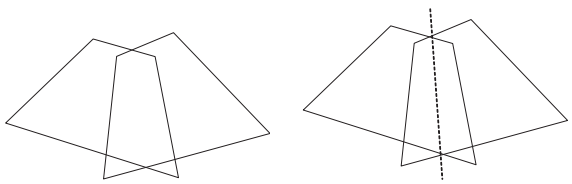


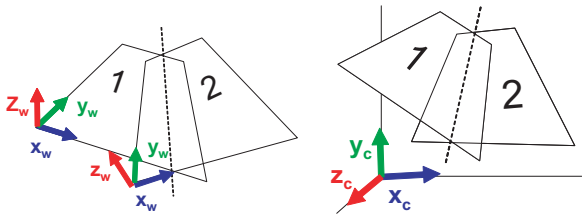
Figure 3.4: Folding line intersection. (a) A reference plane is detected twice in the image. (b) A folding line by computing the intersection line of two planes is estimated.

Coordinate Transformation

Two world coordinate systems independently exist in the image because two different correspondences between a reference plane and an image are established independently as illustrated in Figure 3.5(a). In order to estimate a folding line, these two planes should be described in the same coordinate system.

As a common coordinate system for two planes, the camera coordinate system ($\mathbf{X_c}$) is used, where the origin is the camera center, $X_c Y_c$ plane is parallel to the image plane and Z_c axis corresponds to the depth as illustrated in Figure 3.5(b). The intersection line is computed in the camera coordinate system.

For each plane, a 3×3 rotation matrix (**R**) is computed and 3×1 translation matrix (**T**) from the intrinsic camera parameters obtained by the camera calibration [94] and the homography (**H**) computed in multiple planes detection [38]. Next, the world coordinate system of each plane is projected onto the camera coordinate system by using **R** and **T** of each plane.



(a) (b)

Figure 3.5: Coordinate transformation. (a) Each detected plane is related with a different world coordinate system of the reference. (b) Two planes are transformed into a camera coordinate system to estimate a folding line.

Edge Point Estimation for Constrained Model

A folding line has two edge points because the size of a reference plane is not infinite. Instead of directly computing an intersection line by solving equations from the two planes, two edge points of the folding line are computed. An edge point is obtained by computing an intersection of two border (boundary) lines.

In the folding direction estimation, the direction is categorized into two cases: left-right folding case and top-bottom folding case as shown in Figure 3.6. Suppose there exist two planes composed of four corners **ABCD** and **EFGH** for left-right folding in Figure 3.6(a), one edge point is obtained by computing the inter-

section from two lines **AB** and **EF**. The other point is obtained from two lines **DC** and **HG**. The intersection from two lines **AB** and **EF** is described as

$$\mathbf{X_A} + a_{AB}\left(\mathbf{X_A} - \mathbf{X_B}\right) = \mathbf{X_E} + a_{EF}\left(\mathbf{X_E} - \mathbf{X_F}\right) \tag{3.1}$$

where each side represents a line equation and X_A is the vector representing point A. When an intersection of two 3D lines in the camera coordinate system is computed, a least square method is used because there are three equations with respect to two unknown parameters (each line of a). By computing two edge points from each set of corners, a folding line segment is obtained.

For top-bottom folding, the combination of corners is illustrated in Figure 3.6(b).

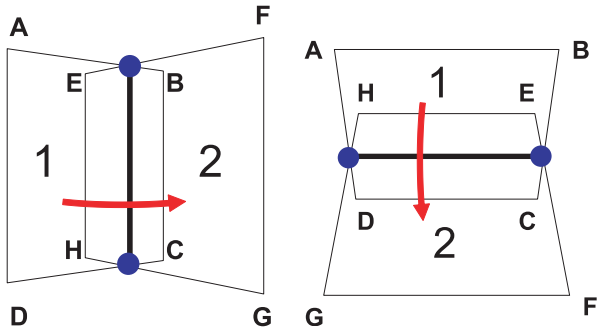


Figure 3.6: Edge point estimation. (a) Left-right folding case. Two edge points from two lines **AB** and **EF**, and two lines **DC** and **HG** are computed. By connecting those two edge points, a folding line is obtained. (b) Top-bottom folding case. Two intersections of two lines **AD** and **HG**, and two lines **BC** and **EF** are computed.

Edge Point Estimation for Relaxed Constrained Model

Folding line is estimated arbitrarily by computing the intersection line of two planes directly without the boundary consideration. The edge estimation can be computed directly from the folding line equation as the intersection of two planes (see Figure 3.7). In contrast with the fixed folding (vertical and horizontal folding), this folding considers each plane as the polygonal shape instead of rectangle.

Suppose there exists an arbitrary folding line which can represented as a parametric equation as follows,

$$L(t) = B + tN \tag{3.2}$$

where B is the position vector, t is the line parameter and N is the direction vector of the folding line.

The direction vector is estimated as follows. Suppose there are two detected planes that lay on the same coordinate system, N_1 and N_2 are the normal vector of each plane that are calculated by taking into account the keypoints on each plane. The direction vector of folding line(N) is the cross product of N_1 and N_2. The dot product of N_1 and N_2 yields the angle between two planes that that can be used also for judging the folded state or non-folded state.

The plane equation for plane 1 and 2 can be expressed respectively as

$$x_1 N_{1x} + y_1 N_{1y} + z_1 N_{1z} + d_1 = 0, \tag{3.3}$$

and

$$x_2 N_{2x} + y_2 N_{2y} + z_2 N_{2z} + d_2 = 0. \tag{3.4}$$

The d_1 and d_2 can be calculated with respect to all points on each plane.

A position vector $B(x,y,z)$ can be determined by any point that lay on the folding line. Because the folding line always intersects line $x = 0$, then y and z of the position vector (B) can be computed by setting $x = 0$ into each equation plane.

The next step is calculating two edge points that lay on the folding line. By setting the t with a constant value, two points that satisfy the folding line equation are acquired. Those two points are then projected back to the reference coordinate system. Those two points have $z = 0$ because the folding line lays on the reference plane which also lays on the $z = 0$. Then the intersection between the projected folding line with the edge of the plane is estimated. Two intersection points form the folding line segment.

3.2 Bended Surface

The bended surface is modeled by solving the progressive finite newton optimization [95]. In this model, a triangle mesh is used as deformation model. The finite newton optimization method is applied to deform the mesh into the paper shape. Suppose a vector S represents a mesh of a bended surface, the two energies are

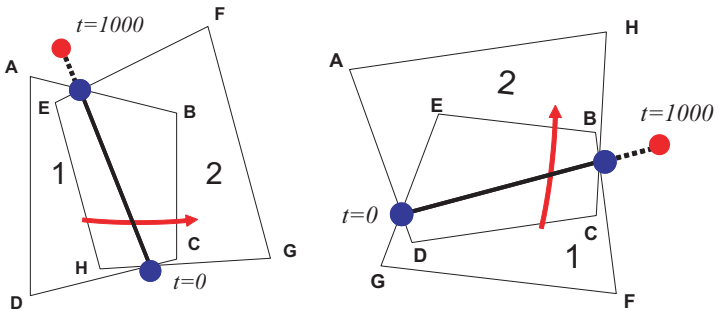


Figure 3.7: Edge point estimation on arbitrary folding. (a) Left-right folding case. (b) Top-bottom folding case. (a) and (b) use the intersection of two planes to estimate the edge points.

minimized: regularization term and correspondence term as the object function of S defined as

$$\varepsilon(S) = \lambda_D \varepsilon_D(S) + \varepsilon_C(S), \tag{3.5}$$

where $\varepsilon_D(S)$ represents the regularization term that constrains the length of a line along the surface should be constant in any deformation, $\varepsilon_C(S)$ is the correspondence term that takes into account of the difference between the keypoints in the input image and the keypoints in bended surface and λ_D is a constant.

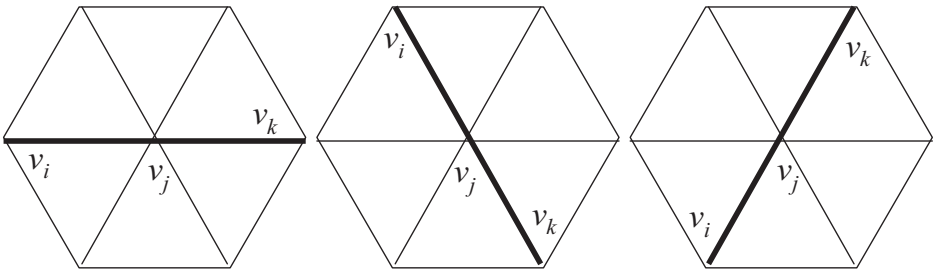


Figure 3.8: Index triplets. Set of index triplets are collected from the mesh. These triplets keep the shape of the surface.

The surface is modeled as a collection points (mesh) and the set of triplets E which consists of m index triplets (v_i, v_j, v_k) are collected from the mesh (Figure 3.8) and apply the following condition

$$\forall(i,j,k) \in E : v_i - v_j = v_j - v_k, \tag{3.6}$$

$\varepsilon_D(S)$ can be written in matrix form as

$$\varepsilon_D(S) = \frac{1}{2}(X^T K'^T K' X + Y^T K'^T K' Y), \tag{3.7}$$

where K' is a matrix $m \times n$. m is the number of triplets and n is the number of points in the mesh. K' is filled with 0 except the columns that are associated with the index triplets are filled with 1, -2 and 1 as shown in the following example

$$K' = \begin{bmatrix} 1 & 0 & -2 & 1 & 0 & 0... \\ 0 & 1 & 0 & -2 & 0 & 1... \\ ... & & & & & \end{bmatrix}. \tag{3.8}$$

For computing the correspondence term, correspondence points that are retrieved by the keypoints extraction and matching are used. A matrix for storing the barycentric coordinate (ξ_i, ξ_j, ξ_k) of each correspondence points is then build. The number of element in the matrix is the same as the number of points in the mesh. For each correspondence points, the barycentric coordinate of reference keypoint is calculated and inserted into the matrix t using the following values $t_i = \xi_i$, $t_j = \xi_j$, $t_k = \xi_k$. The other values is set into zero.

The newton optimization $\varepsilon(S)$ can be simplified into a linear equation as stated in Eq. 3.9 and 3.10. In each iteration, the mesh is updated using the result of the finite newton step written as the following equations

$$s_x = (\lambda_r K + A)^{-1} b_x, \tag{3.9}$$

and

$$s_y = (\lambda_r K + A)^{-1} b_y \tag{3.10}$$

which can be solved using LU decomposition. The $A \in R^{N \times N}$ and $b \in R^{2N}$ are matrices that are computed in each step using the following equations:

$$A = \sum_{m \in M_1} \frac{1}{\sigma^n} t t^\top \tag{3.11}$$

and

$$b = \begin{bmatrix} b_x \\ b_y \end{bmatrix} = \sum_{m \in M_1} \frac{1}{\sigma^n} \begin{bmatrix} ut \\ vt \end{bmatrix} \tag{3.12}$$

which σ value is divided by 2 in every step. The correspondence points are filtered for the next step by calculating the error mapping after the deformation. The points of which the error mapping is bigger than σ^2 are removed.

3.3 Cutting Paper

The third enhancement that can be applied on physical paper is cutting. Unlike the folding and bending that the geometric property is reversible, after cutting the geometric property is irreversible. When a piece of paper is cut, features are also separated. In order to keep detecting each piece, it is necessary to match the features continuously.

3.3.1 Flow

Maximally Stable External Region (MSER) [58] method is applied in order to extract regions in input image. A registration method using hash table is proposed. The cutting detection is simplified by adding separation outline so that the region can be distinguished from each other. As a result, the registration of the surface becomes the registration of the border of the region as illustrated in Figure 3.9.

During runtime, the user firstly draws closed lines on a planar surface as the guide for cutting. The shape of the region is then registered and inserted into a hash table. User then can cut the surface based on the drawn guide lines. Each piece will then be matched with the reference region in the hash table. Each piece is then tracked independently.

3.3.2 Extraction

The feature extraction is done by applying MSER to the input image as illustrated in Figure 3.10. The border of the MSER is simplified using relevance measure that is computed using three consequent points that form two connected lines in the border that is defined as

$$r = \frac{\theta l_1 l_2}{l_1 + l_2} \tag{3.13}$$

where l_1 and l_2 is the length of two connected segments (lines) and θ is the angle between two segments. The point that connects two segment is removed if the relevance measure is smaller than threshold. The result of the simplification is the distribution of keypoints on the border that form high value of relevance measure.

The relevance measure values are then combined to describes three points in the border of a region. The combination of a set of relevance measure is used as the index of a center point of two segments inside the hash table (see Figure 3.11).

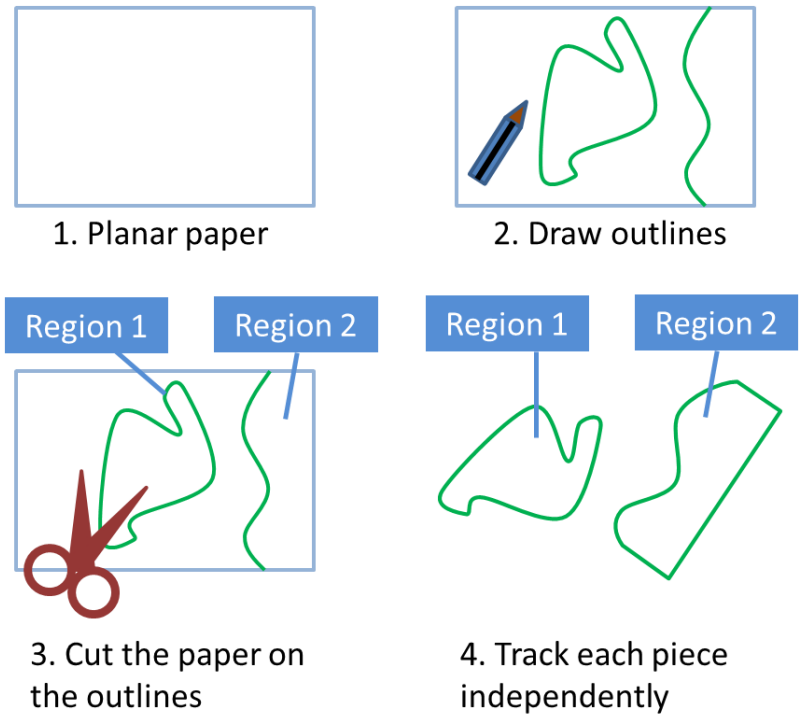


Figure 3.9: Cutting paper scenario. The paper is cut on the outline drawn by the user. Each region is then detected and tracked. The annotation is then augmented onto each region.

During matching, the border of unidentified region is extracted. The set of neighbouring relevance measure is then computed and matched with the index in hash table. If the index is found in hash table, the associated point id is returned.

3.4 Folding Implementation

The folding model is applied into the augmented maps application. Augmented map is an augmented reality application that uses a piece of paper map and overlays geographical information on top of the paper map using video see-through display. In reality, when user holds a map, it is likely to be folded or bent depending how the user holds it. Applying the folding and bending into the augmented maps application will make the usage more natural.

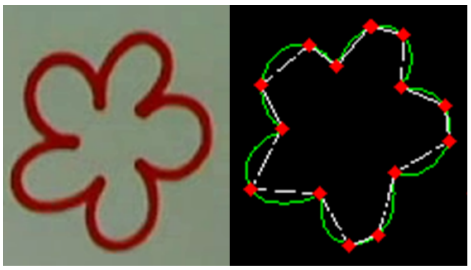


Figure 3.10: Region extraction. The region is extracted using MSER and simplified using the relevance measure.

For realizing an augmented maps application, the random dot marker technique is applied. The random dot marker technique makes use the geographic information on the map as the feature for the map registration. To register the map, the database for storing the reference map is prepared.

3.4.1 Database Preparation

The number of symbols on a typical paper map is considerably sufficient for the tracking purpose (approximately more than 100 dots in a single map). The symbols are geographic-related points of the map. They represent map features such as the position of buildings, houses, stations and important places. Their existence is not obstructive because they are meaningful data for the map. Therefore, the visualization of the keypoints on the map becomes important. At this stage, the symbols are visualized as black dots. It is possible to visualize the symbols as user-friendly map icons to show their importance for the map. In this case, the icon detection should also be considered. Moreover for outdoor use, the color of the symbols is crucial. Strong sunlight may cause high saturation on the captured image so that the extraction may become difficult.

As the off-line procedure in the random dot marker technique, the descriptors of each keypoint (symbol) are computed in the reference maps. It employs the local relationship between a keypoint and its neighbouring keypoints. Even when some keypoints are occluded, the global planarity of the map can still be obtained using the descriptors from the visible keypoints.

A keypoint has multiple indices (1D descriptors) computed from the geometrical relationship of neighbor keypoints. Because each keypoint has unique symbol

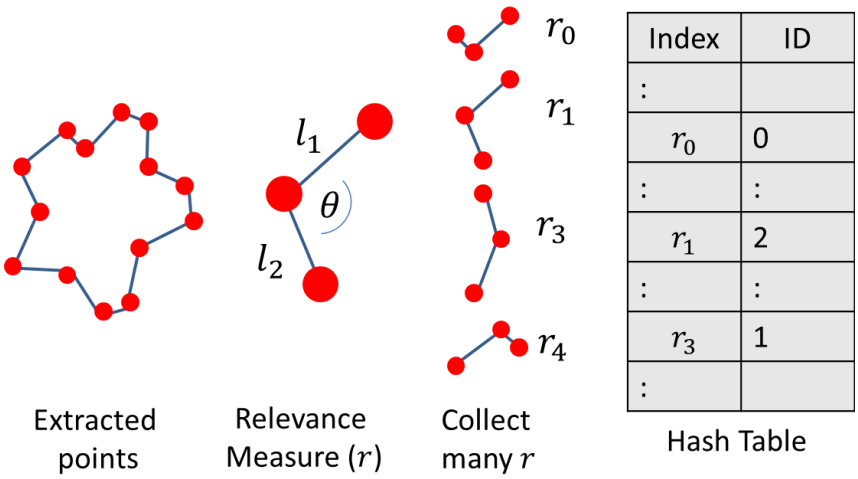


Figure 3.11: Descriptor making. Set of neighbouring relevance measures is stored in the database as the descriptor of a point in the center of two connected segments.

ID (= map ID + keypoint ID), each symbol ID is stored at the indices of the hash table as a descriptor database (an inverted file). The ID is also related with the world coordinate $\mathbf{X_w}$ ($Z_w = 0$) of the symbol.

3.4.2 Folding Initialization

In folded surface detection, the pixels of the symbol color are extracted from an input image. The center position of each symbol region is computed as a keypoint. Next, keypoint correspondences between the input and the reference maps are established. From the correspondences, the map ID is identified and two planes composing the folded surface are detected by iterative geometric verifications.

Because the planes are described in the same camera coordinate system, the geometrical relationship between the two planes is computed. The angle between the two planes is computed using the dot product of two border lines such as **AB** and **EF** in Figure 3.6(a). The angle is used for determining the folding states (folded and unfolded) and the folding conditions (mountain and valley). The details about folding angle is explained later in Subsection 3.4.4.

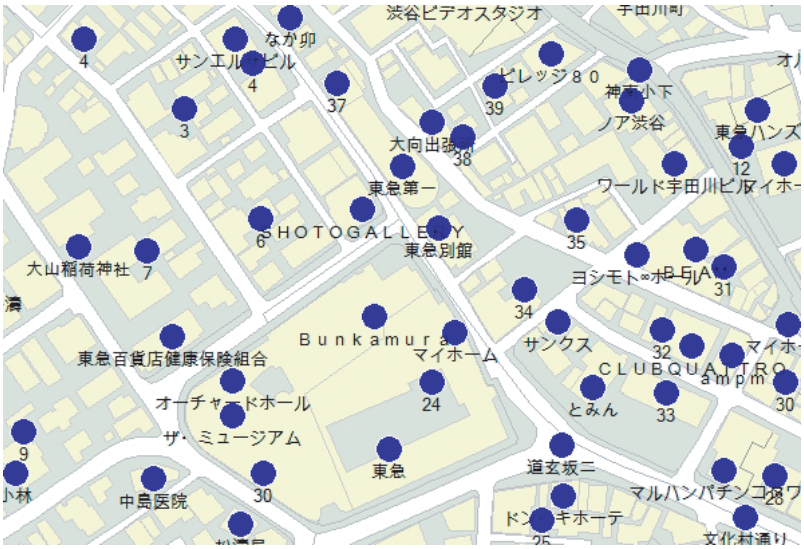


Figure 3.12: A part of the GIS-generated map provided by CAD CENTER COORPORATION in Japan. The center of the geographic symbols (circles) is extracted by using color segmentation as the keypoint extraction step.

3.4.3 Multiple Planes Tracking

A folded surface is modeled as two connected planes. The random dot marker technique facilitates the multiple plane detection and tracking. In the random dot marker database, the plane ID is inserted into the database to distinguish each plane from another.

For keypoints (symbols) in each segmented map, a new symbol ID (= new map ID + keypoint ID) is inserted. The symbol ID is also related with the world coordinate $\mathbf{X_w}$ ($Z_w = 0$). In every frame, the descriptors of keypoints are collected and inserted at the indices of the keypoint in the database.

In addition, for folding purpose, two descriptor databases for tracking are prepared: reference and folding database. Two databases are used in order to avoid false correspondences on keypoint matching. This is because the descriptors of a plane in a folded surface are the subset of the descriptors the reference plane. Therefore, when the surface is not folded, the tracking accesses and updates the default database. Accordingly, when the surface is folded, the tracking accesses and updates the folding database.

After the folding initialization has finished, the reference plane (map) is segmented into two planes. Keypoints and the descriptors that belong to each plane

are copied from the reference database into the folding database. Then the planes are tracked individually using the folding database. The descriptors of each plane are updated in the folding database.

In the folded state, a keypoint is matched by searching its descriptors inside the folding database. From an input image, keypoints are first extracted the same way as the folding initialization. Using LLAH, each keypoint in the image has a symbol ID retrieved from the tracking database. Next, all keypoints in the image are clustered by the map ID extracted from the symbol ID. For each keypoint cluster, RANSAC based homography computation is performed as geometric verification. Finally, two homographies corresponding to the two planes are yielded.

When the planes are tracked, the descriptors of the keypoints in each plane are updated into the folding database as in [89]. For each plane, all keypoints are projected in the reference map onto the image using the homographies. By this projection, the correspondences between keypoints in the reference and those in the image are established by thresholding their distances. If a keypoint in the image has a correspondence, the symbol ID of the projected keypoint is inserted at the indices of the keypoint in the folding database.

3.4.4 State Transition

As a sample case, the valley folding case is illustrated in Figure 3.13. In the initial state, two planes are detected. A threshold angle is used to determine the folding state. While the angle between two planes is smaller than the threshold angle, the state changes to folded. When the angle is larger than the threshold, the state does not change.

In the folded state, the similar process is performed. Two planes are detected and the angle between them is computed. When the angle is bigger than the threshold, the state is set back to the unfolded state. Otherwise, the state does not change.

In contrast with the valley folding, the state transition for the mountain folding uses the opposite comparison. In the unfolded state, when the angle between two planes is larger than the threshold, the state changes to folded. In the folded state, when the angle between two planes is smaller than the threshold, the state changes back to unfolded.

In each state transition, the folding angle computation fails when only one plane is detected. As a result, it is difficult to determine whether the state will change from the folded state into the unfolded state or vice versa. Thus, the

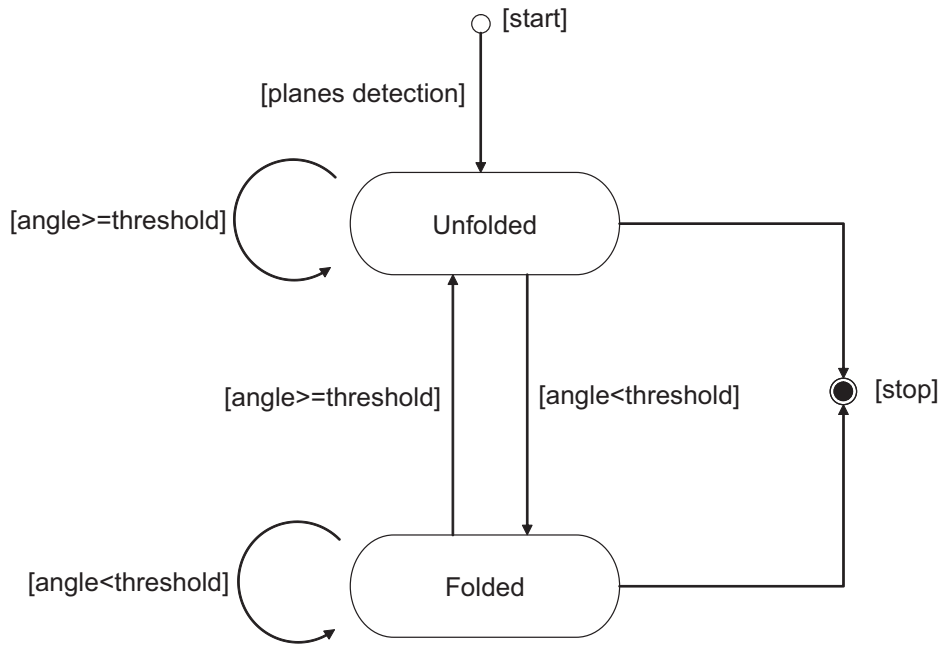


Figure 3.13: The state transition for valley folding. The folded and unfolded state is determined by comparing the angle between two detected planes with a threshold value. In the unfolded state, one plane is tracked and in the folded state, multiple planes are tracked.

state does not change. However, the successfully detected plane is continuously tracked.

3.4.5 Augmentation

A set of 3D models of buildings as the virtual contents provided by CAD CENTER CORPORATION, Japan is used. Each plane in the folded surface is augmented independently. Therefore, the virtual contents are divided into different parts according to the size of each plane.

When the map is not folded in folding initialization, the virtual content is rendered entirely. When the map is folded, the virtual content is divided into two parts at the estimated folding line. The virtual content on each plane is then overlaid using each homography as illustrated in Figure 3.14.

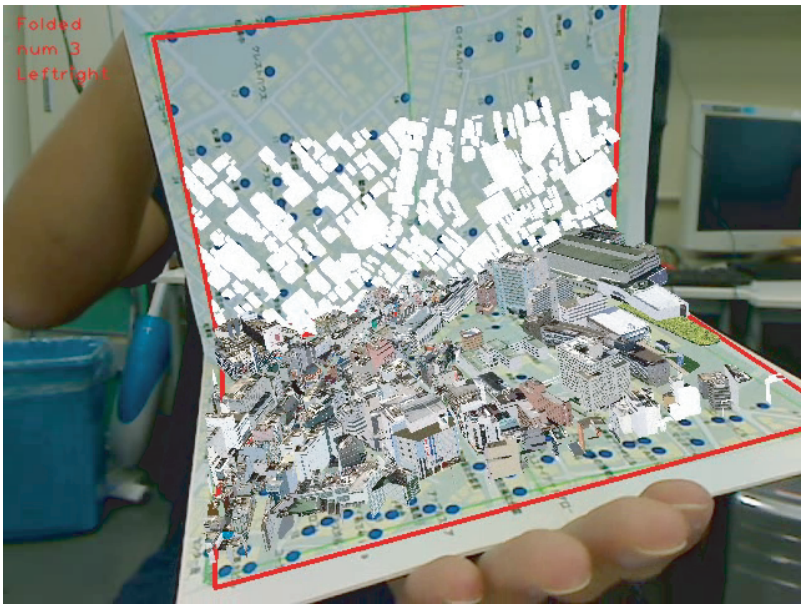


Figure 3.14: Augmentation on a folded surface. The virtual contents are divided into two parts according to the size of each plane, and overlay virtual contents on each plane independently.

3.5 Scenario of Use

3.5.1 Folding

By applying the proposed method, it is possible to add innovative functionalities to paper-based media. The paper becomes interactive and offers a new experience to the readers. Hence, the proposed method is useful for both AR investigations and real world application such as newspaper or books publishers.

2D-3D Foldable Augmented Maps

Novel interactions with augmented paper maps are investigated because maps are widely used and their functionality can be extended. Normally, when the users hold a map and search for destinations around a city, they often refer to the map in order to find their destination. By using augmented maps, the user can see and compare the 3D models and the physical buildings. The virtual contents on augmented maps can help the user recognize the surroundings.

Virtual information could be shown depending on the inclination and distance of the paper to the camera. For instance, it is possible to display subway maps when the map is far from the camera. In contrast, it shows 3D buildings when the map is close to the camera. The user can watch them as they pop up from the map. Depending on the viewer or camera position, the contents can be changed. This mechanism can be achieved because the tracking method can produce the relative pose between surface and camera.

Interactive augmented application using folding can be realized. The content of map changes due to folding states. Moreover, two kinds of information on a folded map can be visualized as shown in Figure 3.15. When the user folds the map, the bigger area of the map visualizes the 3D building of the map. The smaller area of the map visualizes the 2D information. In Figure 3.15, the 2D information of the map is a weather condition of the area displayed in the main folding area.

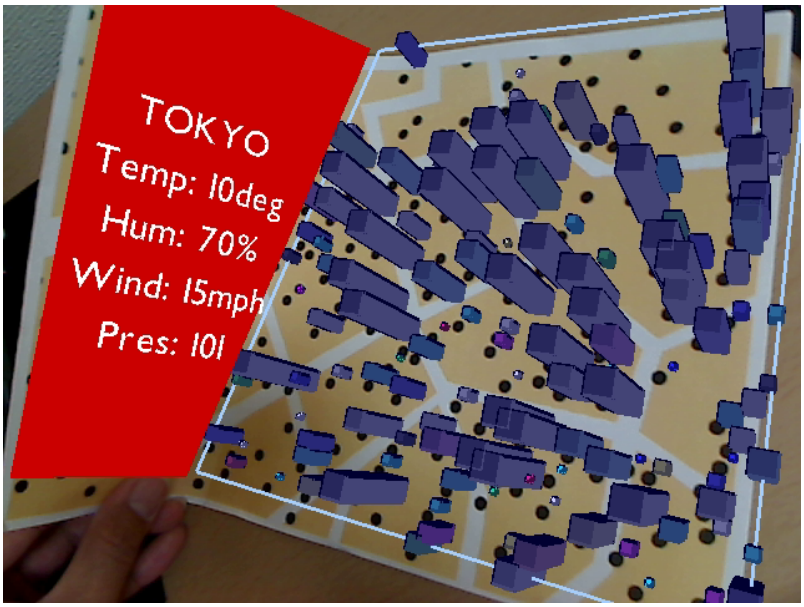


Figure 3.15: Augmented maps with relaxed-constraint folding. The content of the augmented map varies to the folding angle and folding direction on each plane. The larger area contains 3D virtual contents such as 3D city model. The smaller area contains 2D information such as weather information.

Augmented Books

An augmented book is a book that consists of dynamic contents visualized on a display. Early augmented books has been developed by some researchers[14, 26, 82]. Similarly, the augmented books can also be realized as shown in Figure 3.16. The book itself is represented as a scattered-dot-printed paper. The user can fold the side of the paper to change the content of the book.

(a) (b)

Figure 3.16: Arbitrary folding for augmented books. (a) Only one page is displayed when it is unfolded. (b) The folded state of augmented books. The small folding area displays the preview of the next page. When the user folds the paper more, it displays the next pages.

3.5.2 Bending

Interactive application can be realized by using bending upon a piece of paper. Similar to opening a book, bending a piece of paper can be regarded as flipping a page inside a book as illustrated in Figure 3.17. Bending the paper to left will make the system display the next content. Bending the paper to right will make the system display the previous content.

3.5.3 Cutting

In cutting recognition, outline of the regions is drawn in order to simplify the problem. Therefore, before cutting, the user draws the region outline for cutting. This outline is used as the feature for region recognition. This scenario can be applied in the augmented maps application.

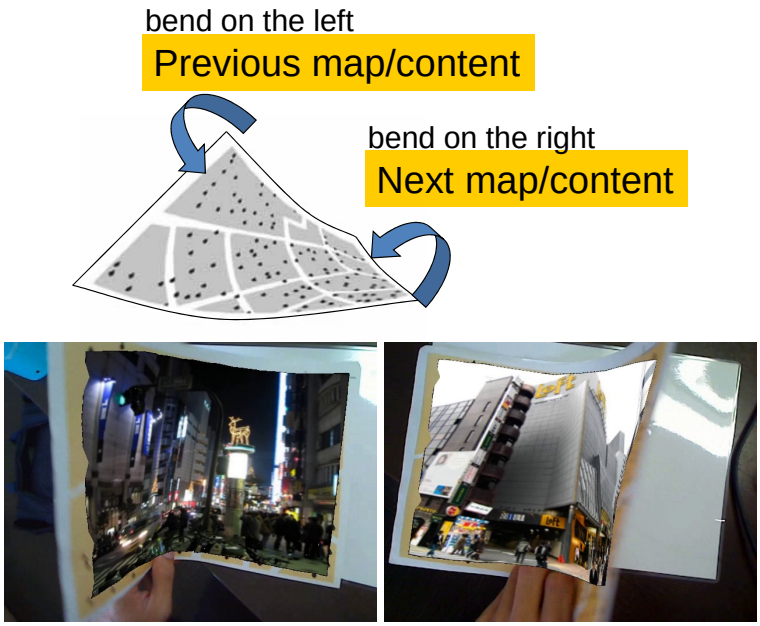


Figure 3.17: Changing contents according to bending. Bend in left or right edge can change the content of the application.

Rearrangeable Maps

A map consists of sets of region as the representation of islands or area. Cutting an area in a paper map and detect the separated areas is proposed as the example of application of cutting interaction.

First, the user draws the outline of a region in a paper map. The system then registers the region. When the registration succeeded, the homography is computed in order to estimate the camera pose. After the camera pose is estimated, the label of the region is superimposed on top of the paper map as illustrated in Figure 3.18.

The next step is the user cuts the detected region. From this point the user can arrange the separated region in order to make a new arrangement of map as illustrated in Figure 3.19. Cutting interaction is also useful for making augmented puzzle application or in-situ authoring system for maps application.

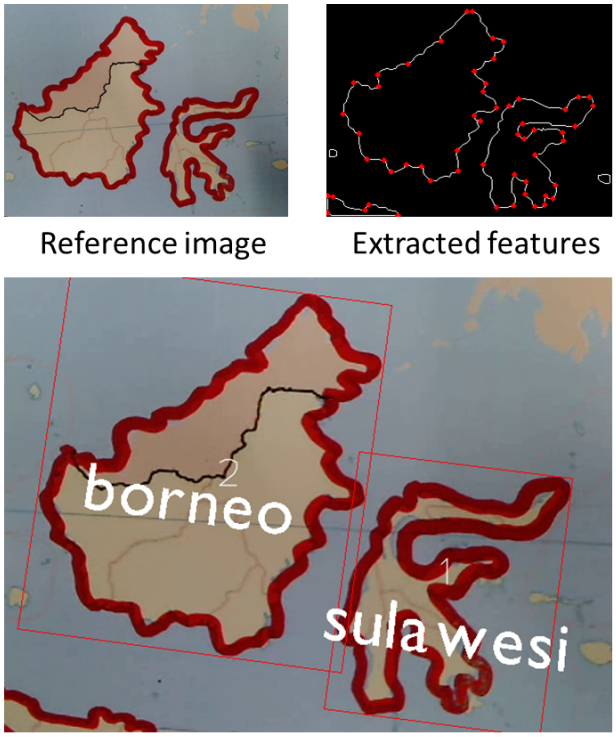


Figure 3.18: Overlaying map information. A map contains regions of island that can be detected using the proposed outline detection method. The name of the island (region) is then virtually overlaid. http://en.wikipedia.org/wiki/Indonesia (map image of Indonesia is from Wikimedia Commons)

Craft Making

On making products made from paper or fabric, designers usually design by drawing or printing patterns on the surface of paper or fabric. Making a note or handwriting on the fabric may change the appearance of the final product. Therefore, using an augmented reality application that applies region detection method, it is possible to superimpose information virtually on the fabric. Even after the user cut the fabric along the pattern lines, the information can be overlaid in order to show how to make or arrange the parts for completing the whole making process. The overlaid information can be reused in for making the same product. The scenario of making crafts is illustrated in Figure 3.20.

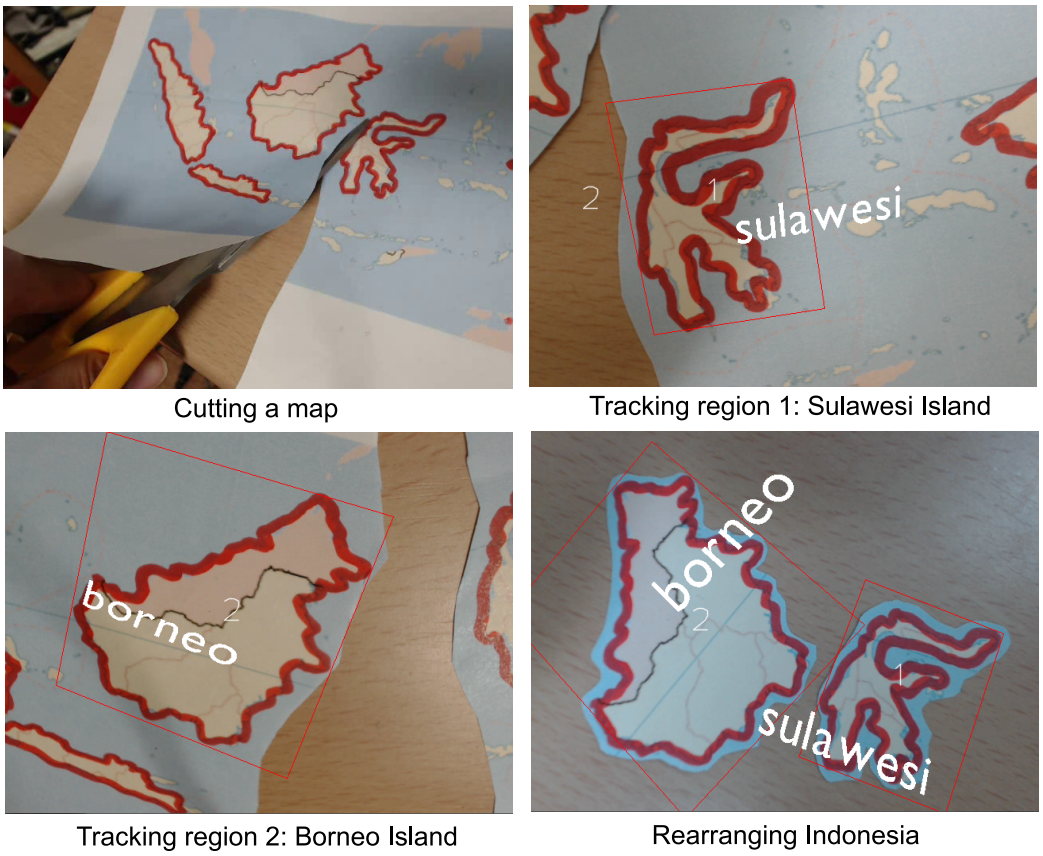


Figure 3.19: Rearranging an archipelago. Firstly, the user cuts a map based on the outline of island. Each island can still be detected and tracked. Each island can be rearrange to make a new composition of archipelago.

A prototype to show the scenario is implemented. In this prototype, a piece of paper is used as the material. A pattern can be hand drawn or printed on the paper. The final product (a paper bag) is illustrated in the Figure 3.21 (b).

The user then cuts the paper and folds the paper in order to make the final product. During the interaction, the information is superimposed virtually over the paper to guide the user as illustrated in Figure 3.22.

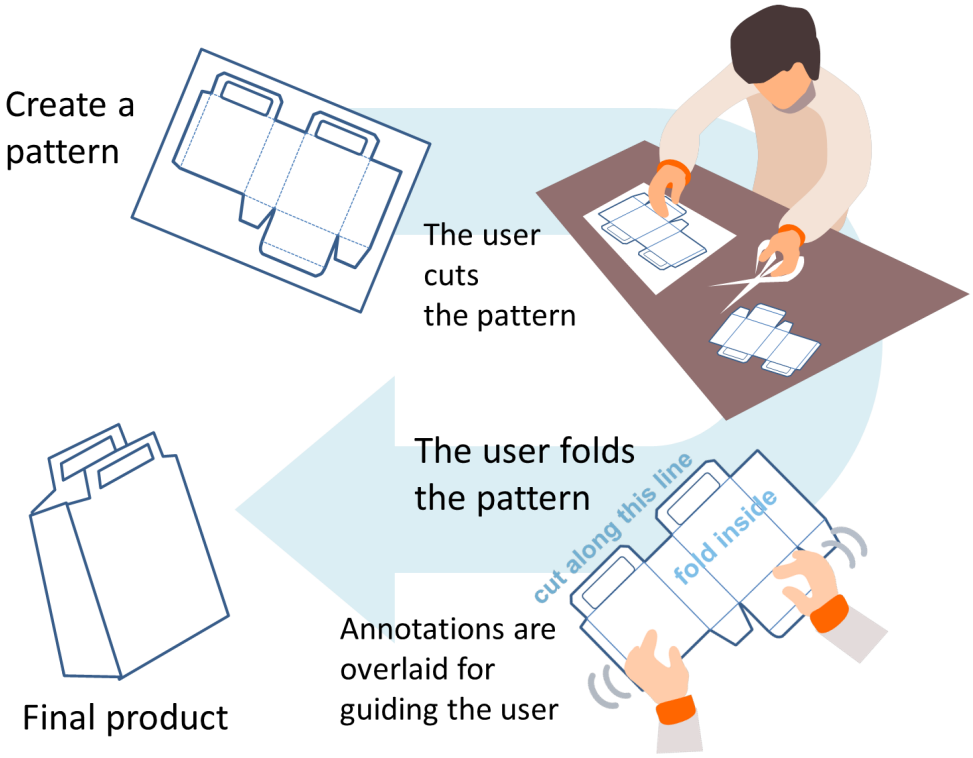
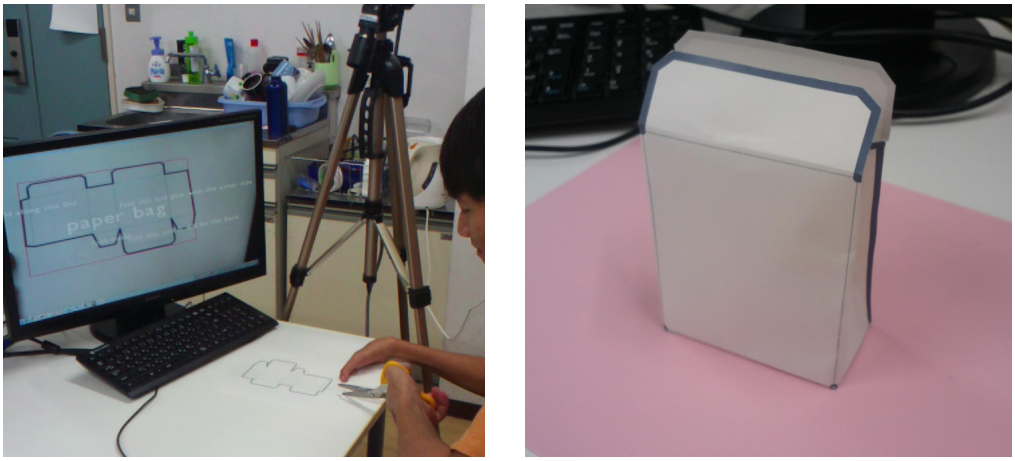


Figure 3.20: Cutting scenario for making crafts.

3.6 Evaluation

For experiments, a desktop PC is used with specifications: Intel (R) Core (TM) 2 Quad CPU Q9550 2.8GHz, 4GB RAM and 640×480 pixel camera. The camera calibration is based on Camera Calibration Tools [4]. The algorithm is deployed in C++ with OpenCV [66] and the 3D models is visualized using OpenVRML [5]. In order to get quantitative evaluations, instead of running the program with a plugged camera, a scene when the user folds a piece of paper is captured and used for further experiments. For whole experiments, the captured video sequence in order to get a valid comparison and ideal environment regardless of the camera frame rate. Two folding models are evaluated in the same way.



(a) prototype setup (b) prototype output

Figure 3.21: Setup and output.

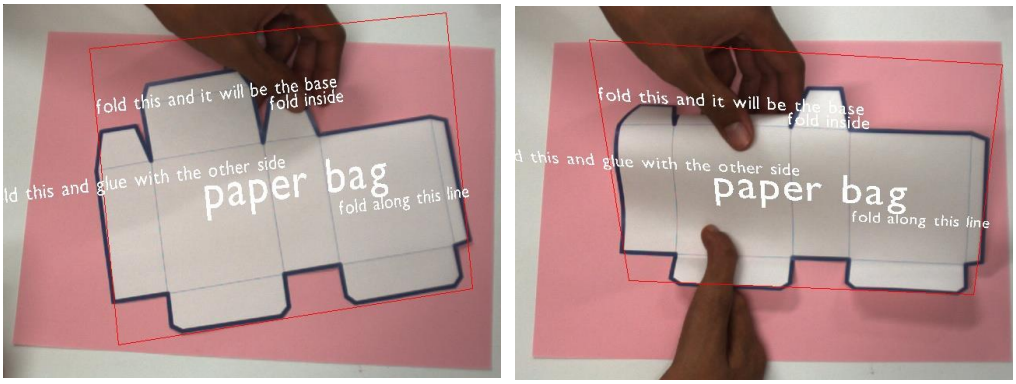
3.6.1 Constrained Folding

Accuracy of Estimated Folding Line

This experiment evaluates the accuracy of the estimated folding line in folded surface detection. The relation between the folding accuracy and the number of tracking points included in two planes is studied. It is assumed that the number of the tracking points appear on each plane also indicates the robustness of the proposed method against occlusions.

First, paper sheets are prepared and folded on the middle so that it forms two planes. For each plane, blue dots are randomly put as described in Table 3.1. In one plane (first), 70 blue dots are randomly put. In the other plane (second), several number of random dots from 20 to 70 are randomly put. For each sheet, one database is prepared. Therefore, the number of points inside the database is the same as the points printed in the sheet.

For each sheet, two edge points are estimated on the folding line and projected them onto the image coordinate system by using the computed camera pose. The two actual edge points are manually clicked on the folded paper captured in the image as the ground truth. For each edge point, its Euclidean distance is computed as illustrated in Figure 3.23, and averaged the results as the error of folding. It is assumed that a good folding accuracy is acquired when the error is close to zero. Experiments are performed on both with and without tracking cases in order to



(a) after cutting (b) folding action

Figure 3.22: Making a paperbag. Virtual text is superimposed during the making in order to guide the user.

study the impact of tracking to the folding accuracy.

As described in Table 3.1, when the tracking method is used, the accuracy increases with the number of tracked points. This means that the accuracy is mainly affected by the estimated homographies because the accuracy of the homography is improved by using more tracking points to disperse the error.

The accuracy of folding without tracking is also calculated (the last column in Table 3.1). It is assumed that tracking will stabilize the detection so that the folding line position doesn't change in each frame.

By comparing the error from the results, it is known that the tracking yielded a better accuracy for the folded surface detection. It stabilizes the keypoints detection because the information from previous frames is kept. Tracking the planes also keeps the folding line remain in the same position. On the other hand, the error of folding line doesn't decrease significantly even the number of tracking points increases when tracking is not used. Thus, this experiment proves that tracking can increase the folding accuracy.

There is also error discrepancy with and without tracking. The experiment with tracking stores the descriptors of previous frame into database. The database contains descriptors of the tracked frames from first to current frame. It contains sufficient and accurate information about the keypoints distribution in previous frames. This technique minimizes the computation error in current frame. Thus, the homography computation error is small. On the other hand, removing the tracking makes the database only contains descriptors for top view image. For tilted conditions, the detection fails or the homography computation gives inaccu-

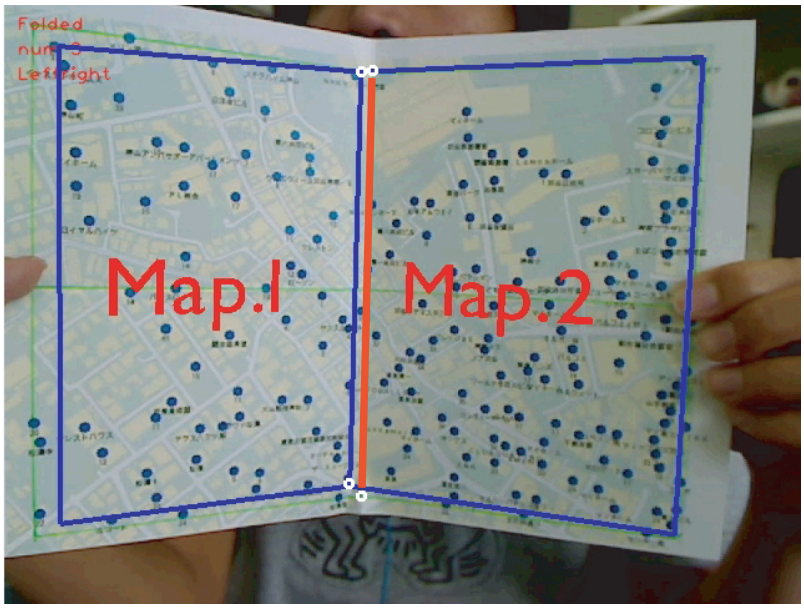


Figure 3.23: Accuracy of the estimated folding line. The distances between the two edge points of the projected line and the ground truth folding line are calculated.

rate result because there is not enough information about tilted conditions in the database. It yields higher error. This make the error of computation is different with and without tracking.

Accuracy of Estimated Folding Angle

This experiment evaluates the accuracy of estimated folding angle. A piece of paper map which contains 140 keypoints is used. It is placed in a fixed position in front of the camera. A folding line is then prepared on the paper such that it divides the paper into two planes and each plane has 70 keypoints. First, the map is detected and tracked. The map is then folded on fixed angles (the fixed angle is manually measured using a protractor) as the ground truth data.

The folding angle values produced by the folded surface detection are then compared with the ground truth data. On each angle, the folding angle is estimated from several consecutive frames and averaged them. The result is shown in Figure 3.24.

The proposed detection method can calculate the angle between plane from

Table 3.1: The average error of estimated folding line with and without tracking. The error is almost proportional to the number of points in each plane with tracking. The error doesn't decrease significantly along with the number of points when the tracking is omitted.

Number of tracking points (first plane)	Number of tracking points (second plane)	Error with tracking (pixels)	Error without tracking (pixels)
70	20	23.04	35.98
70	30	8.76	29.51
70	40	6.63	31.89
70	50	6.69	15.93
70	60	5.95	23.42
70	70	2.81	19.49

40 degrees to 290 degrees. Outside that range, due to extreme tilt, the proposed method successfully detected and tracked only one plane. In this case the folding angle between two planes is not calculated.

The detected angle is almost correctly estimated around the planar condition (the average error from angle 150 degrees to 210 degrees is 4.19 degrees). On valley folding (angle range is between 40 degrees and 180 degrees), the angles tend to be higher than the ground truth angle. On the other hand, the mountain folding (angle range is between 180 degrees and 290 degrees), the angles tend to be lower than the ground truth. The average error of the estimated folding angle from all of experiment set is 9.07 degree. This average error information can be taken into account in order to improve folding accuracy by optimizing the folding angle.

Optimal Threshold Angle for Folding

An angle as threshold value is determined for changing from planar state into folded state and vice versa. To be precise, this threshold value should be a value near 180 degrees. However, due to the tolerance of planarity in RANSAC, two planes cannot be detected distinctively in near planar condition. In addition, in near planar condition the orientation of two planes change from frame to frame that makes the folding line estimation inaccurate. Therefore, it is necessary to determine the threshold value to start the folding line estimation.

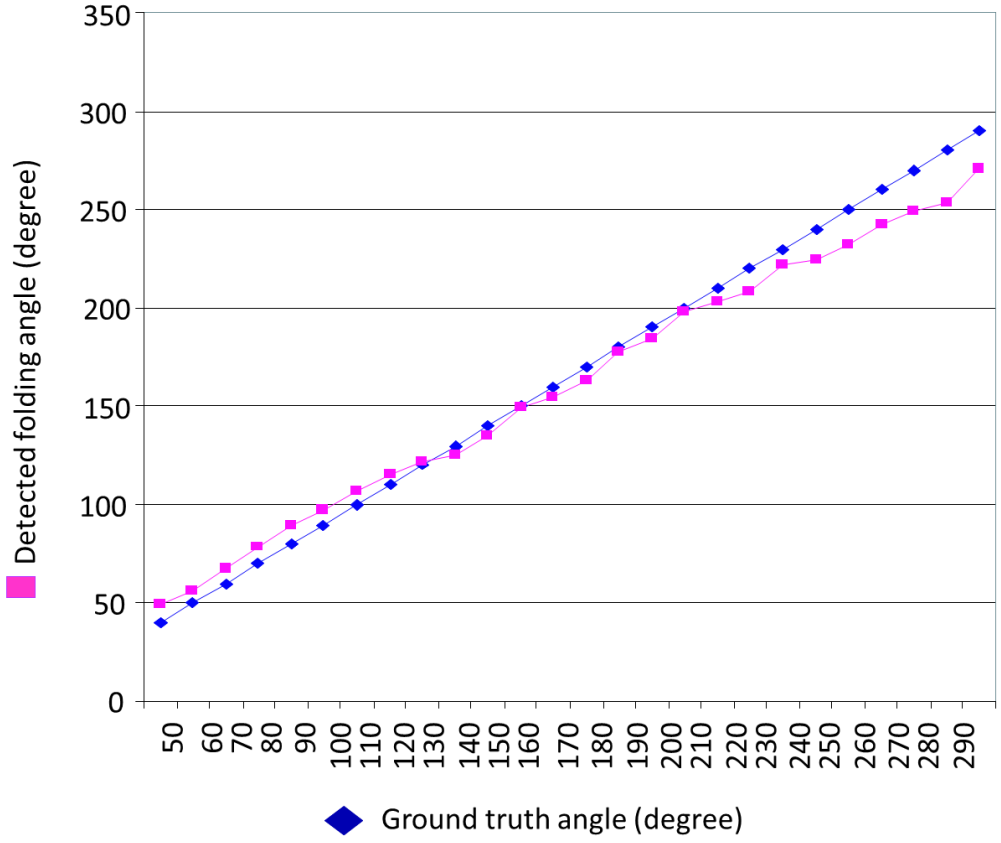


Figure 3.24: Estimated folding angle vs ground truth angle. Several ground truth angles are provided. The folded surface detection achieved folding angle closely to the ground truth data.

The threshold value can be determined arbitrarily. Suppose there is an optimal value that can be achieved from stable detection of two planes, this value can be determined based on the number of detected keypoints. It is assumed the maximum detected keypoints will determine the best condition for folding because the number of detected keypoints is equivalent with the accuracy of plane tracking. Therefore, an optimal threshold value depends on the number of detected keypoints.

In this experiment, the relationship between the calculated folding angle and the number of detected keypoints is studied. When largest number of detected keypoints in a certain angle is obtained, it is assumed that angle is the optimal

threshold value. A piece of paper map which contains 140 keypoints is used. It is placed in a fixed position in front of the camera. A folding line is then prepared on the paper such that it divides the paper into two planes and each plane has 70 keypoints. The paper is folded based on this folding line on several angles. Then the number of detected keypoints is observed as shown in Figure 3.25 for valley folding and Figure 3.26 for mountain folding.

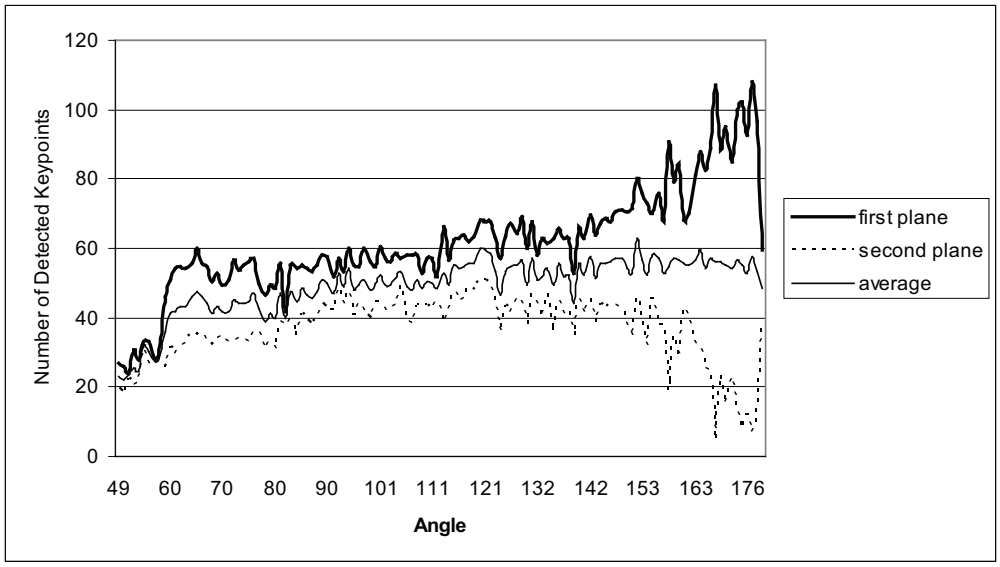


Figure 3.25: The folding angle and the number of detected keypoints in valley folding. When the folding angle approaches 180 degrees, keypoints are mainly detected in the first plane.

The result shows that the number of keypoints in the first plane tends to get larger close to the planar condition (angle 150 degrees to 210 degrees). This is because in those angles, the paper can regarded as planar even though the paper is slightly folded. Accordingly, keypoints are mainly acquired in the first plane. On the other hand, the second plane tends to decrease. The keypoints in second plane are regarded the outliers of the homography computation in the first plane. As a result, it is difficult to distinguish the first plane and the second plane in nearly planar condition.

There is a certain angle where the average of detected keypoints in the first plane and second plane is maximal. The detected keypoints are distributed equally on both planes. This condition is used as the best time to start the folded state and

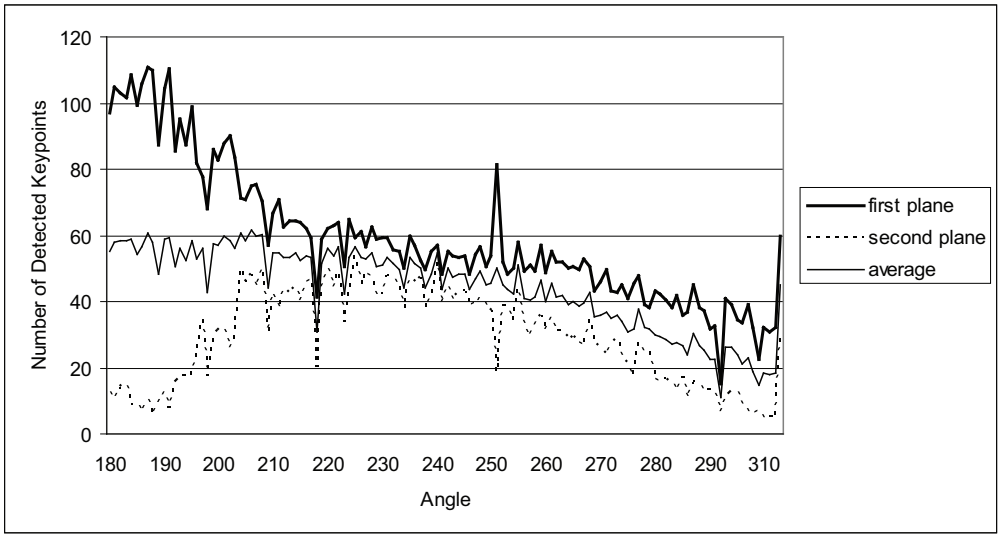


Figure 3.26: The folding angle and the number of detected keypoints in mountain folding. Some keypoints are detected in each plane. When the folding angle approaches 180 degrees, keypoints are mainly detected in the first plane.

the angle is the optimal threshold for folding. From the experiment the optimal threshold of valley folding is 152 degrees with the average keypoints on each plane is 63 and the optimal threshold for mountain folding is 206 degrees with the average detected keypoints is 62.

Folding at Arbitrary Positions

The examples of folding at arbitrary positions are illustrated in Figure 3.27. The figure shows that the user can vertically and horizontally fold the map.

The folding in arbitrary position in the proposed scenario to relax the constraint (half side folding) is studied. In fact, in practical use, the paper may be folded at arbitrary positions accidentally If this condition is ignored and divide the map equally when the user folds at arbitrary positions, the number of keypoints in the database and the map will differ. Some keypoints that belong to a plane will be misplaced into another plane. As a result, this difference will lead to inaccurate homography computation. On top of that, accommodating multiple folding lines requires folding in arbitrary position. Toward a general folding interaction, it is necessary to consider the folding line at arbitrary positions.

The minimum area of each plane for successful detection depends on the num-

ber of keypoints included in each plane as discussed in Section 3.6.1 Accuracy of Estimated Folding Line. If the number of keypoints in one of the detected planes is equal or higher than 20, the folding detection can be performed, as proved by the experiments. However, the estimated folded line includes error in that case.

All experiments in the evaluation always use the ideal condition by folding the paper fairly into two sides where the folding line is located in the middle of the paper. It is assumed that the result of this ideal condition will also apply to folding lines in arbitrary positions. In fact, the folding at arbitrary positions is equivalent to the half side folding containing different number of keypoints on each plane. Therefore, the accuracy of folding at arbitrary positions is proven in Section 3.6.1 Accuracy of Estimated Folding Line. The accuracy of folding line and the angle will decrease because dividing the plane unequally is similar to reducing the number of keypoints in one side of plane.

Performance

Processing time of the different tasks in the implementation is given in Table 3.2. Each computational cost is recorded and averaged in consecutive 451 frames. In the folded surface detection, the RANSAC-based homography computation is the most costly task. However, this computation can be replaced with multi-RANSAC [99] for faster computation in the future. Because tracking multiple planes takes at about 3 ms, the augmentation during the tracking was at over 30 frames per second.

The computation cost for augmentation depends on the complexity if the 3D models. Simplifying the 3D models can speed up the augmentation. In the implementation, a region clipping method is simply used for separating 3D models in each plane. Overlapping 3D model around the folding line requires an automatic visualization handling such as re-meshing based on the clipping. This process demands more computational time which will be a challenge for the future research.

Table 3.2: The computational time.

Tasks	Computational time (msec)
Folded surface detection	71.7
Multiple planes tracking	2.9
Augmentation	8.03

3.6.2 Relaxed Constrained Folding

Accuracy of Estimated Folding Line

This experiment is done similarly to Section 3.6.1 Accuracy of Estimated Folding Line. However, the paper is folded in slanted direction instead of vertical or horizontal. As described in Table 3.3, the accuracy increases when the number of keypoints in a plane increases in case tracking method is used. This means that the accuracy is mainly affected by the accuracy of the estimated homographies because the accuracy of the homography is improved by using more keypoints to disperse the error.

Table 3.3: The average error of estimated folding line. Different number of keypoints are prepared for one sheet of paper. The error is almost proportional to the number of keypoints.

Number of tracking points (first plane)	Number of tracking points (second plane)	Error (pixels)
70	20	32.32
70	30	37.12
70	40	23.52
70	50	23.43
70	60	22.65
70	70	19.35

Accuracy of Estimated Folding Angle

This evaluation is done similarly to Section 3.6.1 Accuracy of Estimated Folding Angle. The result is shown in Figure 3.29.

The detection method can calculate the angle between plane from 50 degrees to 290 degrees. Outside that range, because one plane is tilted extremely, only one plane is detected. As a result, the folding angle between two planes is not measured.

The result shows that the detected angle is almost correctly estimated. The average error of the folding angle in respect to the ground truth data is 4.45 degrees.

Various Folding Results

The examples of folding on arbitrary positions are illustrated in Figure 3.30. This figures shows that the user can fold the paper in diagonal position.

Because RANSAC based homography computation is used as geometric verification, two regions of a folded surface should be planar. To keep two regions as planar surfaces, thick papers are used for the whole experiments.

3.6.3 Region Detection Performance for Cutting

In order to evaluate the cutting interaction, it is necessary to test the proposed region detection. This test shows how accurate the region detection on five hand drawn regions as illustrated in Figure 3.31. The purpose of this experiment is proving that the proposed region detection method performs well and is repeatable. 100 sequence images are chosen from the captured video as the hand drawn region template. The number of detected region for each frame is counted.

The result of the detection is illustrated in Figure 3.33. Table 3.4 shows that regions with fewer corners such as region 0 is detected less accurately than the other regions. Figure 3.32 shows that region 0 is simplified into a set of points that form fewer unique features than the other regions.

Table 3.4: Successful detection rate. Most regions have successful detection except for region 4 that has fewer features than other regions.

Region	Number of detected frames
0	1
1	96
2	97
3	93
4	99

3.7 Summary

The folding, bending, cutting-based model and recognition for paper based augmented reality are explored. The application of the modelling and the recognition is implemented. In the application, the augmentation when tracking is performed at over 30 fps, which is sufficient for real time application.

The recognition of the geometrical property change on a piece of paper such as folding, bending and cutting are challenging topics to explore. The folding model can be extended into non-constrained interaction which is possible for developing origami system in the future. While the bending interaction can be extended into multiple bended surfaces or bended book application especially in 3D environment. Finally, the cutting interaction can also be applied into another application that requires multiple region detection and tracking.

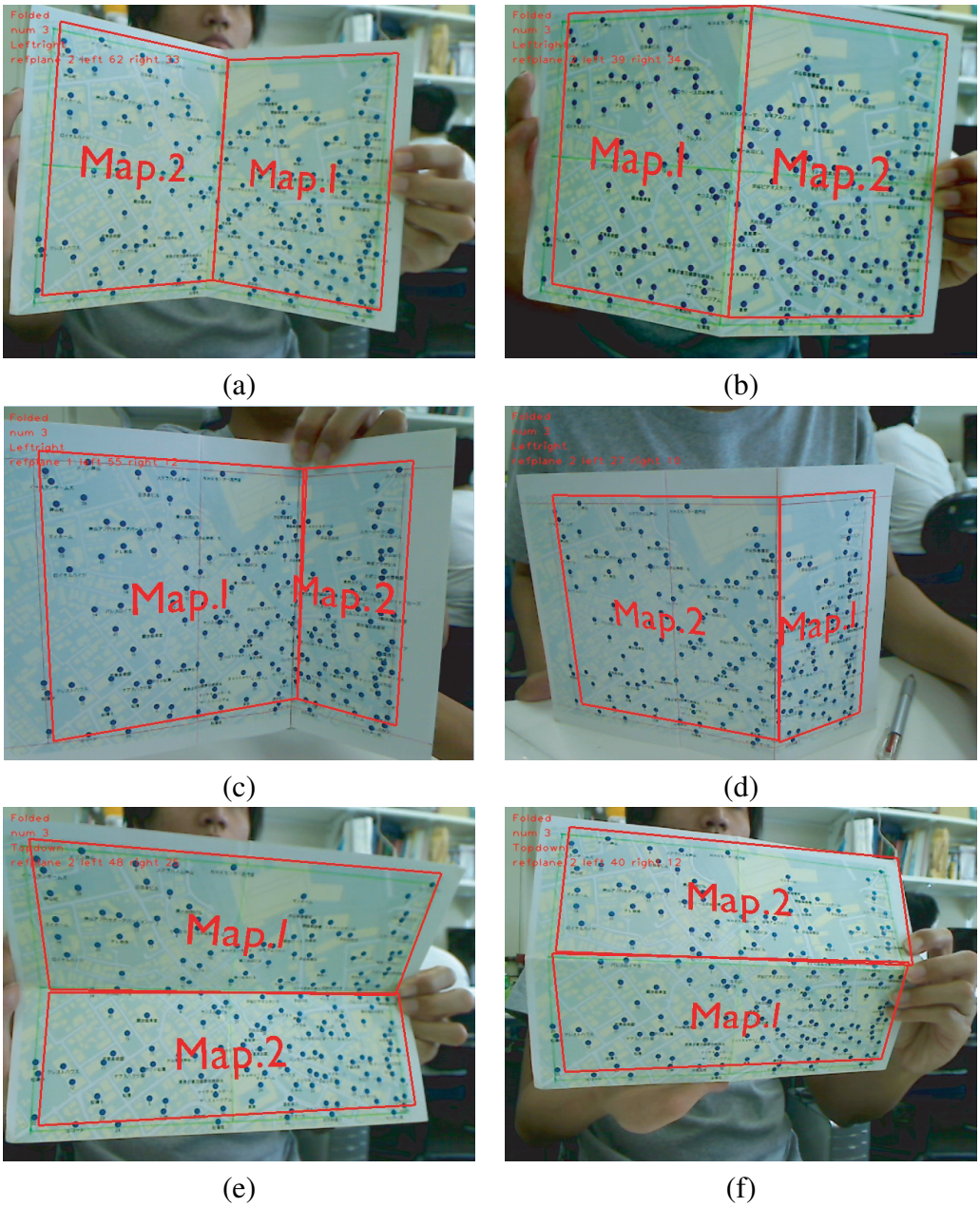


Figure 3.27: The folding at arbitrary positions. (a) Left-right valley folding. (b) Left-right mountain folding. (c) Left-right valley folding in a position different from (a). (d) Left-right mountain folding in a position different from (b). (e) Top-bottom valley folding. (f) Top-bottom mountain folding.

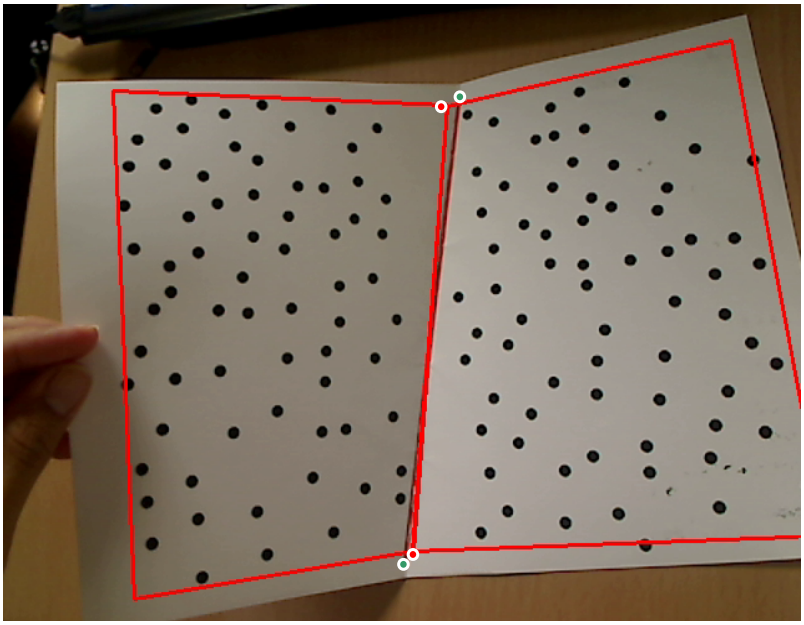


Figure 3.28: Accuracy estimation. The estimated folding line is projected onto an image. A red line printed in the paper is the ground truth folding line. Each distance between two edge points of the projected line (red dots) and the ground truth folding line (green dots) is calculated.

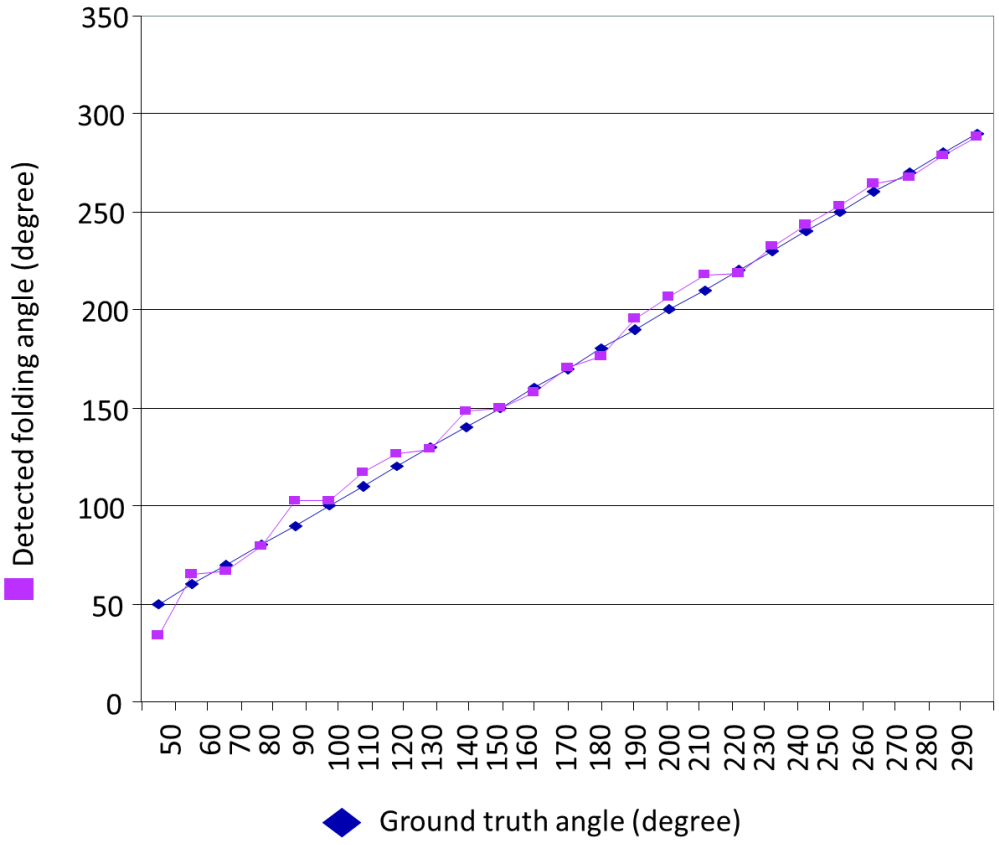
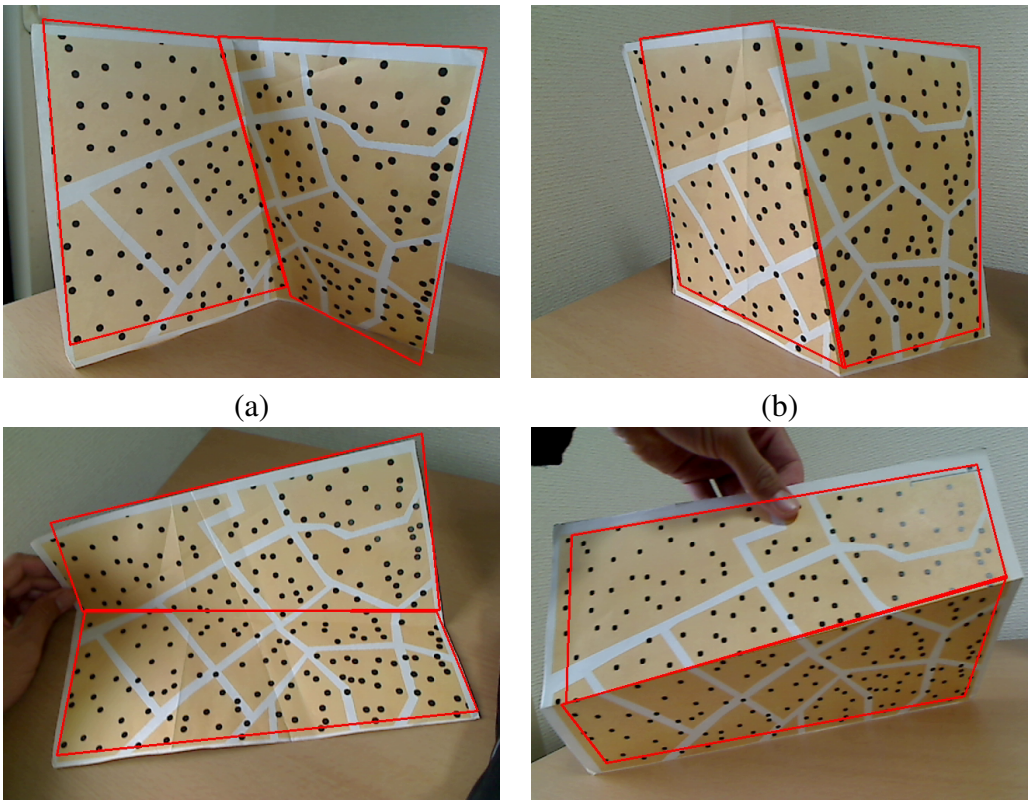


Figure 3.29: Detected folding angle vs ground truth angle. The several ground truth angles are provided. Our folded surface detection achieved folding angle close to the ground truth data.



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

Figure 3.30: Mountain and valley folding results. (a) Left-right valley folding. (b) Left-right mountain folding. (c) Top-bottom valley folding. (d) Top-bottom mountain folding.

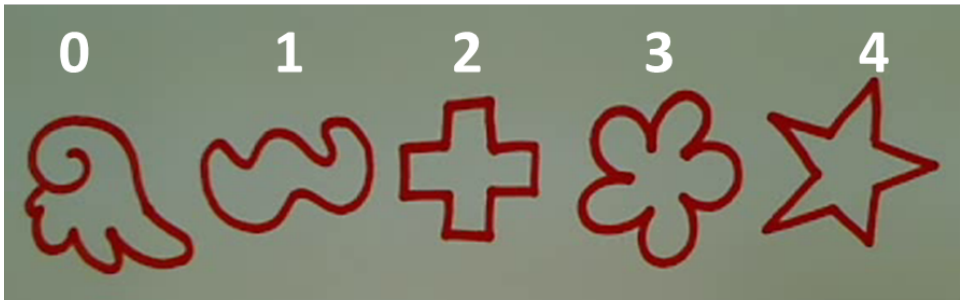


Figure 3.31: Region template. Five different templates are captured and inserted in a hash table.

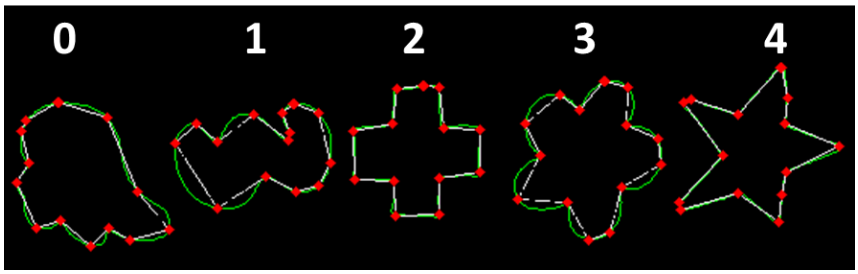


Figure 3.32: Extracted feature. Shape 0 has fewer sharp corners than the other shape which contributes low accuracy on detection.

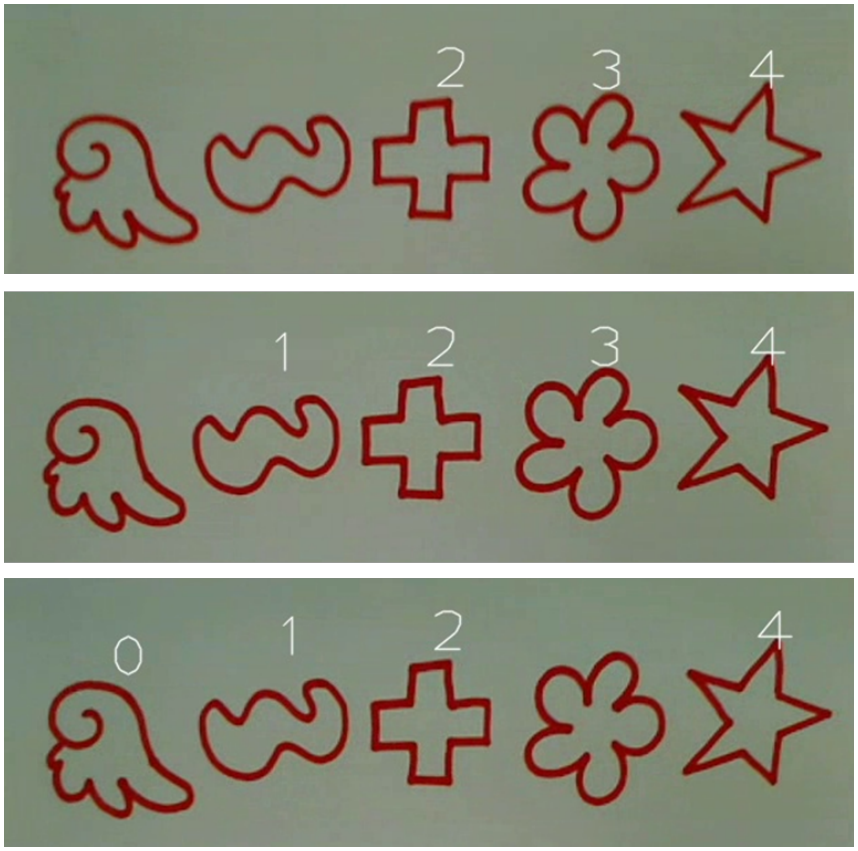


Figure 3.33: Region detection result in 3 sample frames. Regions can be detected in most frames. Regions with fewer corners (mostly smooth curves) have less accuracy.

Chapter 4

Alignment Method for Projector-camera Setup

In conventional augmented reality systems, a monitor or a head mounted display is used. When monitor is employed, the user can only move in a small private space around the monitor display. It limits the user freedom and is impractical for outdoor system. When a head mounted display is employed, each user must wear one to view the same virtual content together. In order to remove those limitations, augmenting virtual contents directly onto physical paper is explored. By augmenting virtual contents directly onto physical paper, the functionality of physical paper can be enhanced and augmented reality on any surface can be achieved.

Recently, projectors are used to augment virtual contents onto a real surface using spatial augmented reality [15, 73]. Spatial augmented reality can be applied in large area that can creates public space. Furthermore, recent investigations allowed the user movement using a portable projector attached on a mobile phone [47, 54, 76, 78]. Spatial augmented reality systems usually place a target surface and a projector stationarily. In order to realize a portable system where user can hold the surface and move it freely, projecting contents on a movable surface is necessary. A planar tracker is used for registering a movable surface and projecting virtual contents in aligned position on the surface [48, 57].

This chapter describes a feature-based method for aligning virtual contents on a movable planar paper using a projector-camera pair without any special devices such as light sensors, motion sensors, or infrared cameras (see Figure 4.1). In the wearable and surveillance setup, camera, projector, and target paper can move independently. The proposed method in those setups is capable of doing the automatic calibration (camera and projector pose estimation) during runtime by

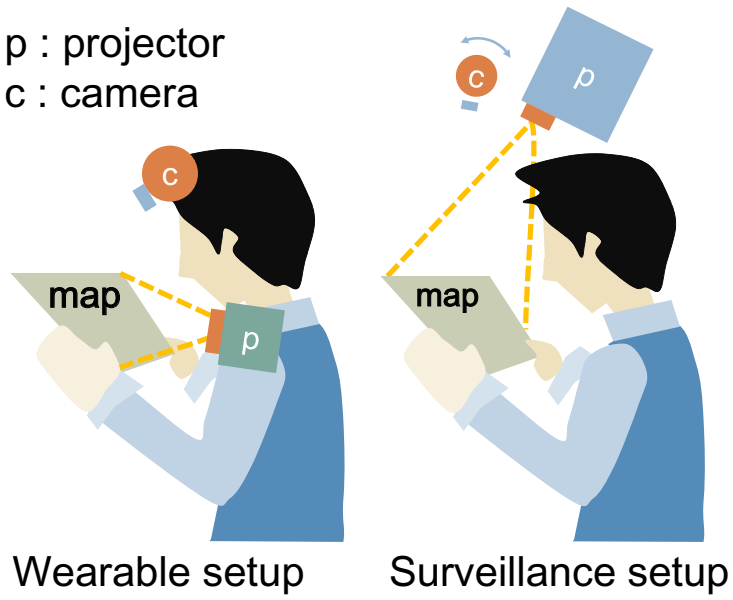


Figure 4.1: A wearable and surveillance setup. Both setups allow the arbitrary paper movement.

detecting the target paper and the projected content simultaneously.

A method that is applicable for pre-calibrated (integrated) and non-calibrated setup without losing a generality is developed. Any setup such as integrated setup or free moving projector-camera pair could be realized. Two kinds of applicable setup using free-moving projector and camera (non-calibrated) are presented. Note that the non-calibrated setup is desirable and ideal setup that has been the goal of recent researches in spatial augmented reality field. Regardless fixing the setup beforehand (pre-calibrated) would be technically feasible and easier to implement.

Compared to the integrated setup where the camera and projector are fixed and precalibrated, the proposed method allows the user to move arbitrarily while holding the target surface. In this case, the camera detects the target surface and projector automatically projects the aligned content. The user movement and the unintentional geometrical changes between projector and camera in each frame will thus require re-calibration. The proposed method can automatically compensate these changes and estimate the camera and projector pose in real time.

In order to register features in the paper, the random dot marker technique is

applied. First, the keypoint features are printed on the paper. During runtime, the same features in different color from the printed ones are projected using projector. Both printed and projected features are detected. Each homography of the printed and projected features relative to the features in database is calculated. A transformation matrix between these two homographies is estimated. The transformation matrix is used to warp virtual contents so that it will be projected aligned to the paper.

4.1 Proposed Method

An alignment of virtual contents on a piece of paper is achieved by image warping before the projection. First, the random points contained in the paper (reference dots) are projected in blue color. Note that the same points are printed in the paper as in red color beforehand.

In each frame the red dots are detected and a homography H_c that relates the reference dots in reference coordinates system into the image coordinate system is calculated. The projected points is detected and a homography H_p that relates the reference dots in reference coordinate system into the image coordinate system is calculated (see Figure 4.2).

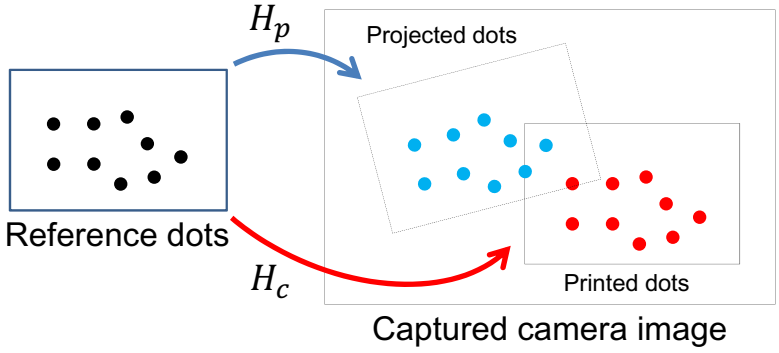


Figure 4.2: Estimating homographies. In a captured image, both the printed red icons and projected blue dots are detected. Then the planarity or homography of the projected blue dots (H_p) and printed red icons (H_c) is estimated.

Using H_c and H_p, a homography that transforms the virtual content (in the reference coordinate system) into the aligned position on the paper in the pro-

jector coordinate system is calculated as illustrated in Figure 4.3. The complete algorithm for estimating the transformation including the initialization and the transformation update is listed Figure 4.4.

4.1.1 Initialization

First, partial content of the paper (points) are projected as random dots in blue color. A transformation (homography) between the projected random dots to the printed red icons is estimated as follows:

$$X_a = HX_p, \tag{4.1}$$

$$H = H_p^{-1} H_c, \tag{4.2}$$

where the H_c is the homography that maps the reference dots into the printed dots and H_p is the homography that maps the reference dots into the projected dots captured by the camera. X_p is the original contents to project, X_a is the final output which is the aligned virtual contents, and H is the output homography for the alignment.

4.1.2 Transformation Update

After the projected blue dots are detected, the blue dots are warped to the location of the corresponding red icons. This condition is called as aligned state. In each succeeding frame, it is necessary to continue warping virtual contents into the aligned coordinate system. Therefore, the blue dots are re-detected and the homographies (H_c and H_p) are re-estimated. Before computing the transformation (warping) function (H), the error between two homographies is computed using the following error function:

$$E(X'_p, X'_c) = \frac{1}{4} \sum_{i=1}^{4} d(X'_{pi}, X'_{ci}) \tag{4.3}$$

where $d(X'_{pi}, X'_{ci})$ is the euclidean distance between each corresponding four corners of the boundary of the map and the projected dots in the image coordinate system. When $E(X'_p, X'_c)$ is inside a $range_0$ to $range_1$ the H_p will be updated using the old H_p and if it exceeds a threshold, the method reinitializes.

The transformation H is updated as follows:

$$X_{a^{t-1}} = H_t X'_p, \tag{4.4}$$

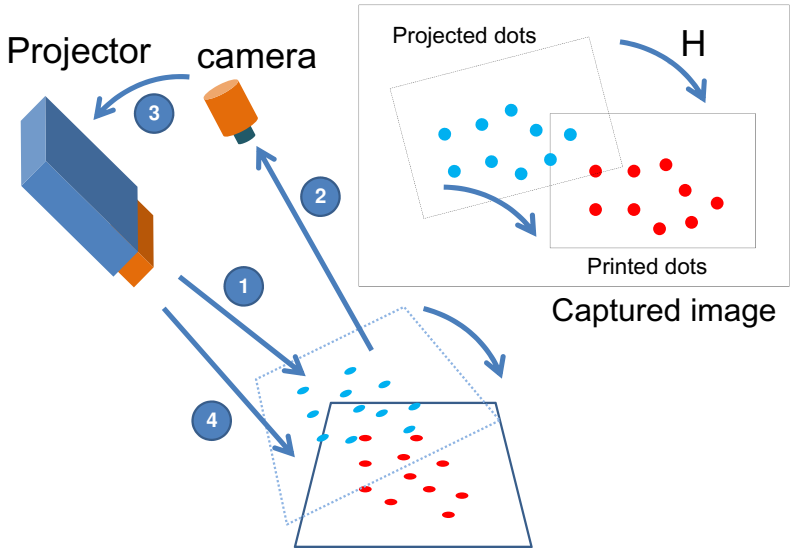


Figure 4.3: Transformation estimation. After H_c and H_p are estimated, a homography (H) that relates the projected blue dots planarity to the printed red icons planarity is calculated. The virtual contents are warped using H.

$$H = H_t H_c, \tag{4.5}$$

where the H_t is the homography that maps the captured blue dots in the image (X'_p) into the blue dots in the aligned coordinate in previous frame ($X_{a^{t-1}}$). The homography for transforming the content for the current frame (H) is estimated by multiplying H_t with the homography that relates the red icons captured in the image with the reference dots in database H_c. Thus, the correct H calculated from the previous frame is used.

4.2 Implementation

The proposed alignment method is applied to desktop version of augmented maps in previous chapter. The alignment method extends the previous augmented maps on the visualization by adding the random dots projection, warping and virtual contents projection.

Require: H_p, H_c
  if *init* **then**
    $computeH$ ▷ (*Equation* 4.2)
  else
    $X_p \leftarrow H_p X$
    $X_c \leftarrow H_c X$
    $e \leftarrow E(X_p, X_c)$ ▷ (*Equation* 4.3)
    if $e > range_0$ **then**
      $init \leftarrow true$
    else if $e < range_0$ & $e > range_1$ **then**
      $H_p \leftarrow H_{p0}$
      $computeH$ ▷ (*Equation* 4.5)
    else
      $computeH$ ▷ (*Equation* 4.5)
      $H_{p0} \leftarrow H_p$
    end if
  end if
  return H

Figure 4.4: The algorithm for computing H in every frame.

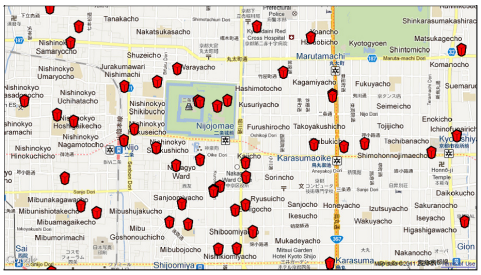
4.2.1 Map and Geographical Contents

A piece of paper map contains landmark points and a background map. Landmark points are printed as red icons on top of the background map as illustrated in Figure 4.5 (a). The geographical contents such as labels, navigation information and photo are used as virtual contents as illustrated in Figure 4.5 (b),(c), and (d).

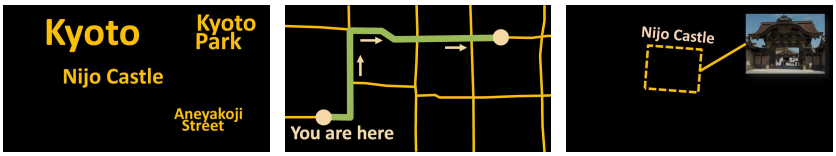
4.2.2 Features Extraction

The map features are extracted using color-based extraction. First, the input image is separated into three color channels R, G, and B. The value of R channel is subtracted with (G+B) channel. The result is then binarized to get blobs of red icons. The center of each blobs is used as the features.

In order to extract the blue projected dots, the value of B channel is extracted with (R+G) channel. The result of the feature extraction is illustrated in Figure 4.6 (b) and (c).



(a) A map image ©2011 Google,ZENRIN



(b) Map labels (c) Navigation (d) Photo

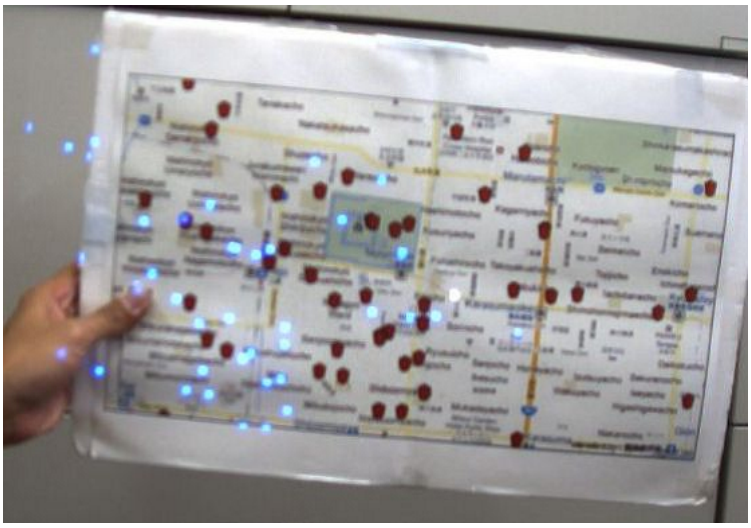
Figure 4.5: A map image and geographical contents.

In order to distinguish the projected blue dots with the other blue color printed in the map, the brightness value of the blue color should be high. The brightness value is evaluated by extracting the HSV channel of the image. The third channel (value) is compared with a certain threshold in order to get the projected blue dots.

4.2.3 Paper Map Registration

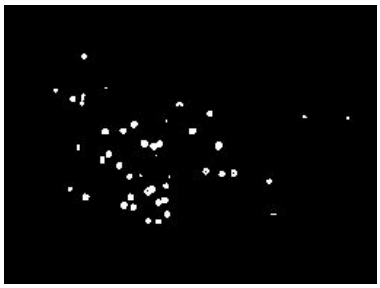
The random dot marker method [90] is used for surface detection and tracking because of its robustness to partial occlusions. Partial occlusions occur because projected contents may alter the captured features in the paper map. The random dot marker method uses the affine invariant descriptor computed from the combination of neighbors dots.

In an offline phase, a descriptor database from a text file containing the location of landmarks (reference dots) in the geographical coordinate system is created. The paper map is registered using the following steps. The red icons printed on the surface are extracted based on color (Section 4.2.2). Then the extracted keypoints are matched with the reference dots stored in the database. Finally, the keypoints correspondences and the homography (H_c) are calculated. The projected blue dots are detected similarly and matched with the same reference dots. The homography H_p is then calculated.



(a) A captured image containing red icons and projected blue dots

(b) Extracted red icons



(c) Extracted projected blue dots

Figure 4.6: A captured image and features extraction.

4.3 Evaluation

4.3.1 Setup

The setup for experiments, are Intel (R) Core (TM) i7 CPU M 640 2.8GHz 4GB RAM laptop and 640×480 pixel Point Grey camera. The projector specification is PLUS U4-232h projector in 1024×768 image resolution. The code is implemented in C/C++ with OpenCV [66] and OpenGL for rendering. The camera is placed a projector and a piece of paper map is in front of the projector as illustrated in Figure 4.7.

4.3.2 Accuracy

The projection error is observed by calculating the distance between two boundaries projected by H_p and H_c as illustrated in Figure 4.8. The first 170 frames are examined and the error function is calculated using the Equation 4.3.

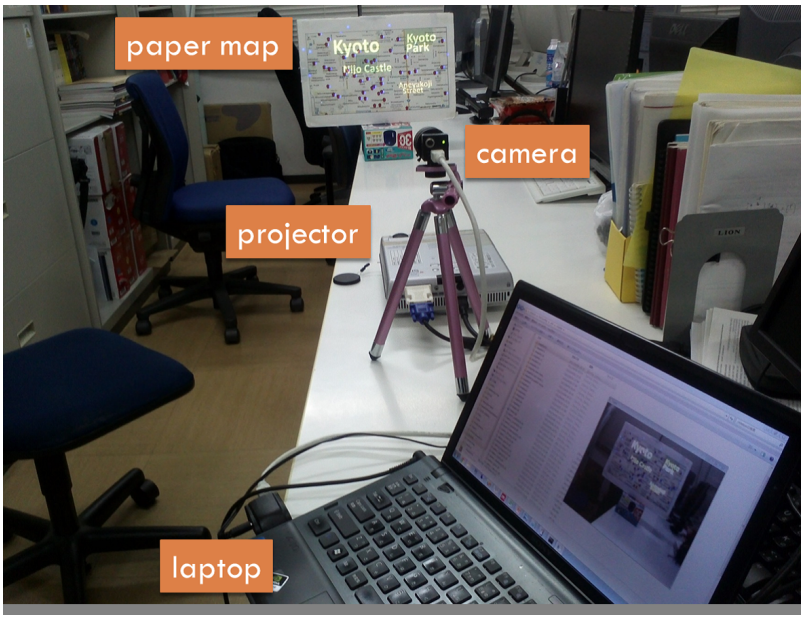


Figure 4.7: The experiment setup.

Three conditions were observed based on the paper map motion, the camera motion and the projector motion as listed in Table 4.1. The projection error is plotted in Figure 4.10, 4.11, and 4.12 respectively.

Table 4.1: The projection error during movement.

Condition	Average distance (pixels)	Standard deviation
Paper map moves	11.9	5.3
Camera moves	9.3	8.4
Projector moves	7.4	5.6
Average	9.5	

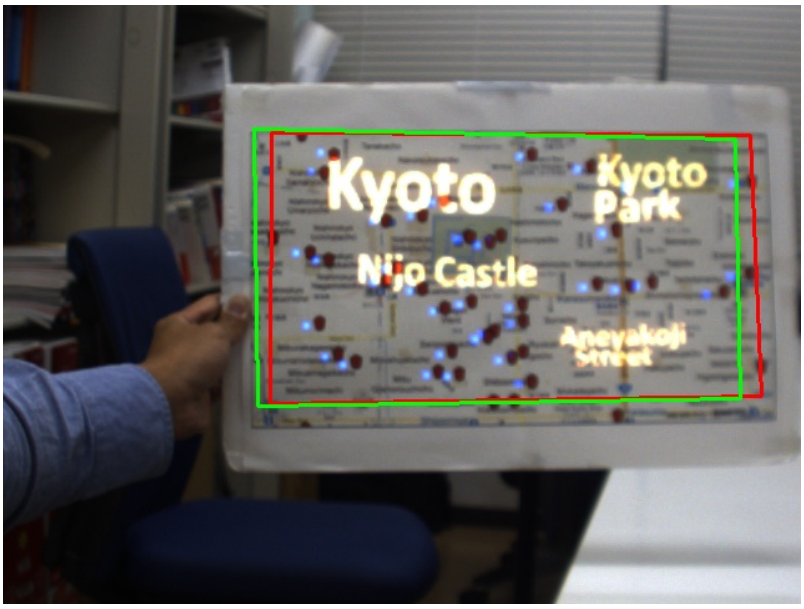


Figure 4.8: Projection error. The projection error is estimated by averaging the distance between four corners of the red border and green border. Red border is the projection result to image coordinate by the H_c. The green border is the projection result to image coordinate by the H_p.

The average error due to the paper motion such as translation and rotation (see Figure 4.9) in the first 170 frames is 11.93 pixels. In Figure 4.10, the error suddenly increases in some frames. In frame 16, the blue dots are not easily detected because it is necessary to store information from the previous frames until the tracking becomes stable (less jitter). When the paper map remains stationary in frame 46, the error reaches minimum. The paper map is rotated along z-axis from frame 46 to frame 80. Because of the delay of the projection, the projected blue dots can not be projected at the same time as the paper map moves and the error reaches maximum in frame 76. The paper map is rotated along x-axis and y-axis from frame 81 to frame 170 and the error reaches maximum when the detection fails in frame 154. After some detection failures (large projection error), the proposed method re-initializes the detection for the next alignment as shown in the graph.

The mapping error in all cases are caused by the accuracy of H, H_c and H_p. The accuracy of those matrices is influenced by the number of matched features.

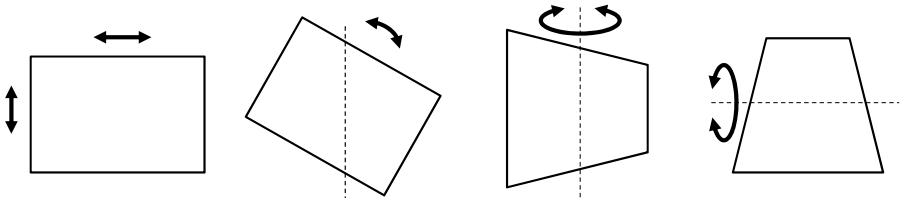


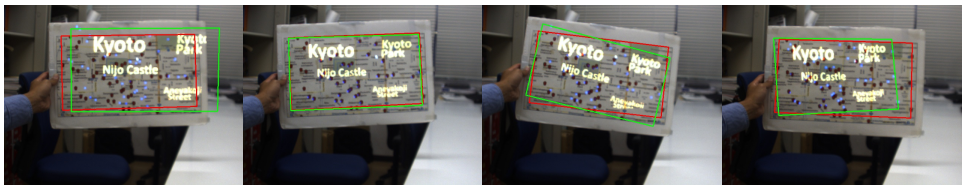
Figure 4.9: The paper map motion (translation and rotation).

Note that the number of matched features is influenced by the feature matching (registration) result. In Figure 4.10,4.11, and 4.12, the big errors are caused by the abrupt movement of the paper map, camera or projector which make some of the features are not extracted or occluded and thus unmatched. Therefore, the observable error is measured by evaluating the influence of the keypoint correspondence to the mapping errors. Figure 4.10,4.11, and 4.12 show that abrupt movements of the paper, camera or projectors will reduce the number of matched features and thus increase the mapping error. Figure 4.10,4.11, and 4.12 also show that even after the big mapping error occurs, the proposed method can reinitialize and reduce the mapping error automatically.

The average error due to the camera motion (see Figure 4.11) in the first 170 frames is 9.28 pixels. The error is lower compared to the paper map motion. However the extreme camera motion produces big errors as shown in frame 75, 96 and 150. The average error due to projector motion (see Figure 4.12) in the first 170 frames is 7.42 pixels. Similar to the camera motion, the extreme projector motion produces big errors as shown in frame 62 and 126.

4.3.3 Speed and Alignment Results

The processing time is calculated using the same videos from the accuracy experiments. The average computation time is 75.6 msec in each frame. In other words, the proposed method can run in approximately 14fps. This computation cost is small and thus fast enough for augmented reality application. The results of the alignment on a movable paper map are shown in Figure 4.13. Figure 4.14 shows that the proposed method is also capable to compensate the 90 degree to 180 rotation. Furthermore, the speed of the projection can be examined in the videos provided. As described in previous section, the projection mapping error



(a) Frame 16 (b) Frame 46 (c) Frame 75 (d) Frame 154

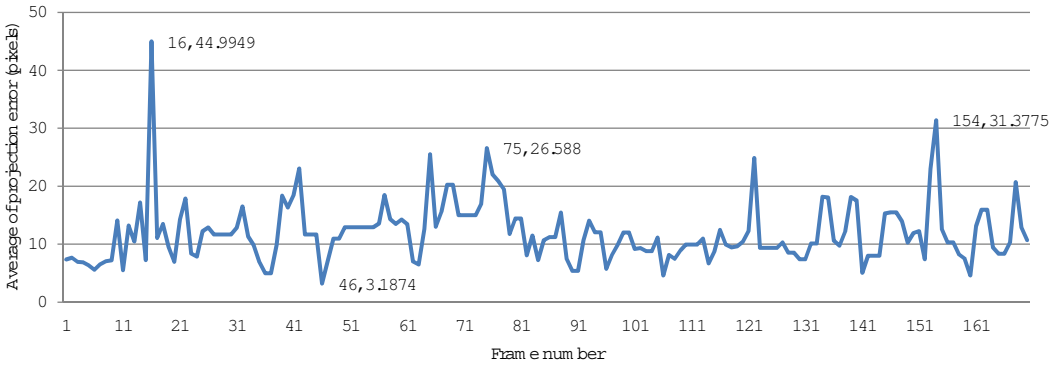


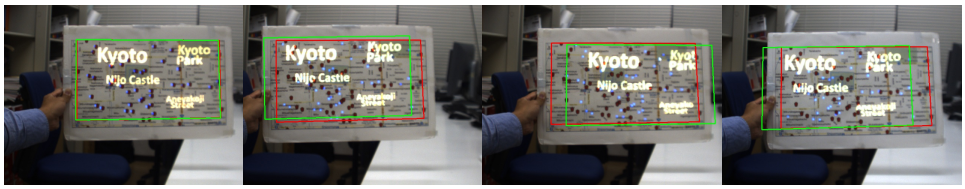
Figure 4.10: Projection error in 170 frames due to map motion. Frame 1-16 is the initialization process. The paper map is translated in frame 17-45. The paper map is rotated along z-axis in frame 46-80. The rest is the paper map is rotated along x-axis and y-axis.

in each projection is computed. When the error exceeds a threshold due to abrupt motions, the proposed method re-initializes and maintains the correct projection as demonstrated in the result videos as well.

Although the computation time required for alignment is small, transferring the contents into projector and capturing the new projected content takes longer than 75.6 msec. In order to get the correct content from the previous projection, 100 msec delay in each iteration is added intentionally before starting to capture the next frame. This delay value is determined by some trials that show values lower than 100 msec produced false detection.

4.4 Occlusion Handling

There are some conditions occur when projecting the virtual contents on the paper map. On initial state the blue dots are projected without any alignment as



(a) Frame 26 (b) Frame 75 (c) Frame 96 (d) Frame 150

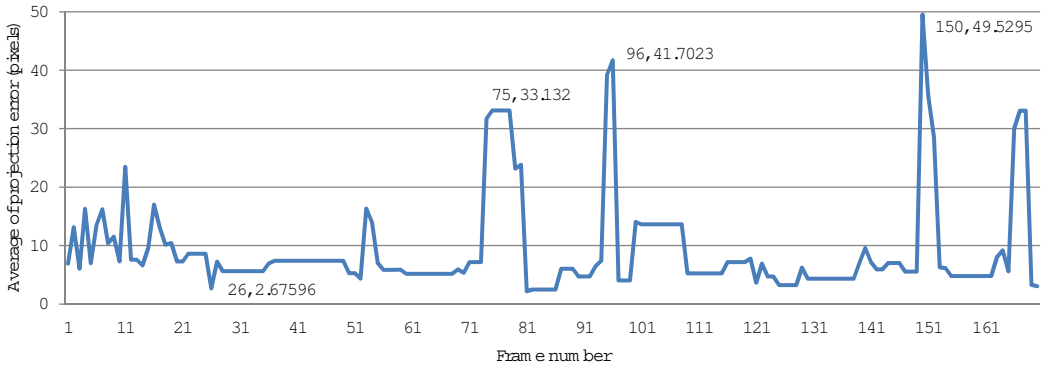
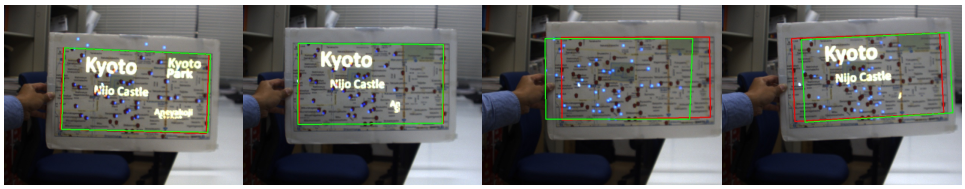


Figure 4.11: Projection error in the first 170 frames due to the camera motion (rotation along y-axis). The error is relatively lower than due to the paper map motion.

illustrated in Figure 4.15.

When the initial alignment is successfully performed, the blue dots will occlude the red dots on the map. The red dots and blue dots can not be detected since red and blue color mix and they are captured as different color as shown in Figure 4.16. This problem is called projection hinder tracking. In this case, the previous successful alignment homography is used and content projection can be continuously performed.

When the paper or camera or projector moves the projected blue dots and the red dots reappear and the detection can be performed as shown in Figure 4.17. Therefore, the homography is updated and the content is projected using the updated alignment homography. Using this mechanism the proposed method can track the paper continuously even after the occlusion occurs.



(a) Frame 2 (b) Frame 54 (c) Frame 62 (d) Frame 126

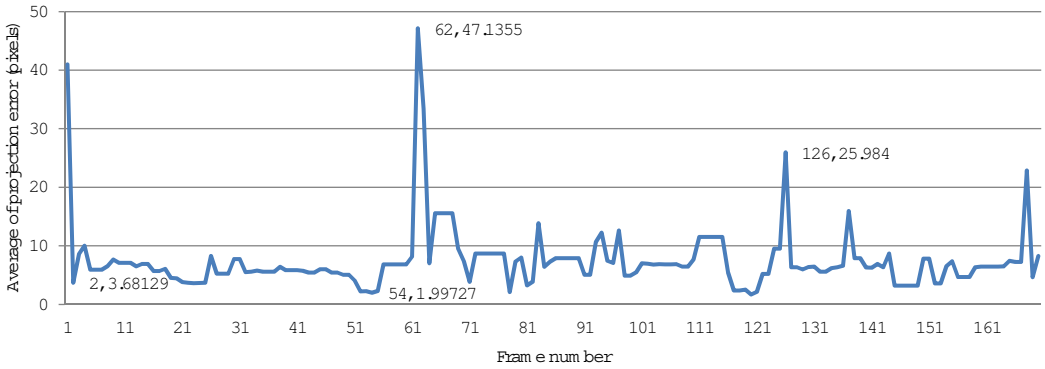


Figure 4.12: Projection error in the first 170 frames due to the projection motion (rotation along y-axis).

4.5 Limitations

The proposed aligned method has limitations. Since the random dot marker method is applied for detection, the limitation of the random dot marker method affects the accuracy.

4.5.1 Projection Hinder Tracking

Projection hinder tracking issue also occurs when the virtual contents are projected dominantly on the paper map that may occlude the random dots printed on the map. Virtual contents such as label and other content that do not acquire space on the paper map can be successfully projected.

4.5.2 Rotation

The proposed aligned method takes into account the information from previous frames. When the current tracking fails then the proposed method uses the previ-

ous homography. In addition, when the projection error reaches a threshold value, the method reinitializes. The initialization is performed by projecting blue dots at initial position. As a result, when the paper map is rotated and the tracking fails, the method reinitializes and the position of the paper map should be rotated back to its initial position. The cases of failures due to ration are depicted in Figure 4.18.

4.6 Summary

An alignment method on a piece of paper map using a projector-camera setup have been presented. In order to allow the paper movement, the random dot marker method for registering and tracking the paper and warp the virtual contents onto the aligned position is applied. Thanks to the update of the transformation in each frame, the geometrical changes between the projector and the camera can be compensated. This approach allows the aligned projection without depending the calibrated projector-camera setup nor specific devices such as motion sensors, light sensors or infra red cameras. This method can track the movable paper map and project the aligned contents with projection error of 9.5 pixels.

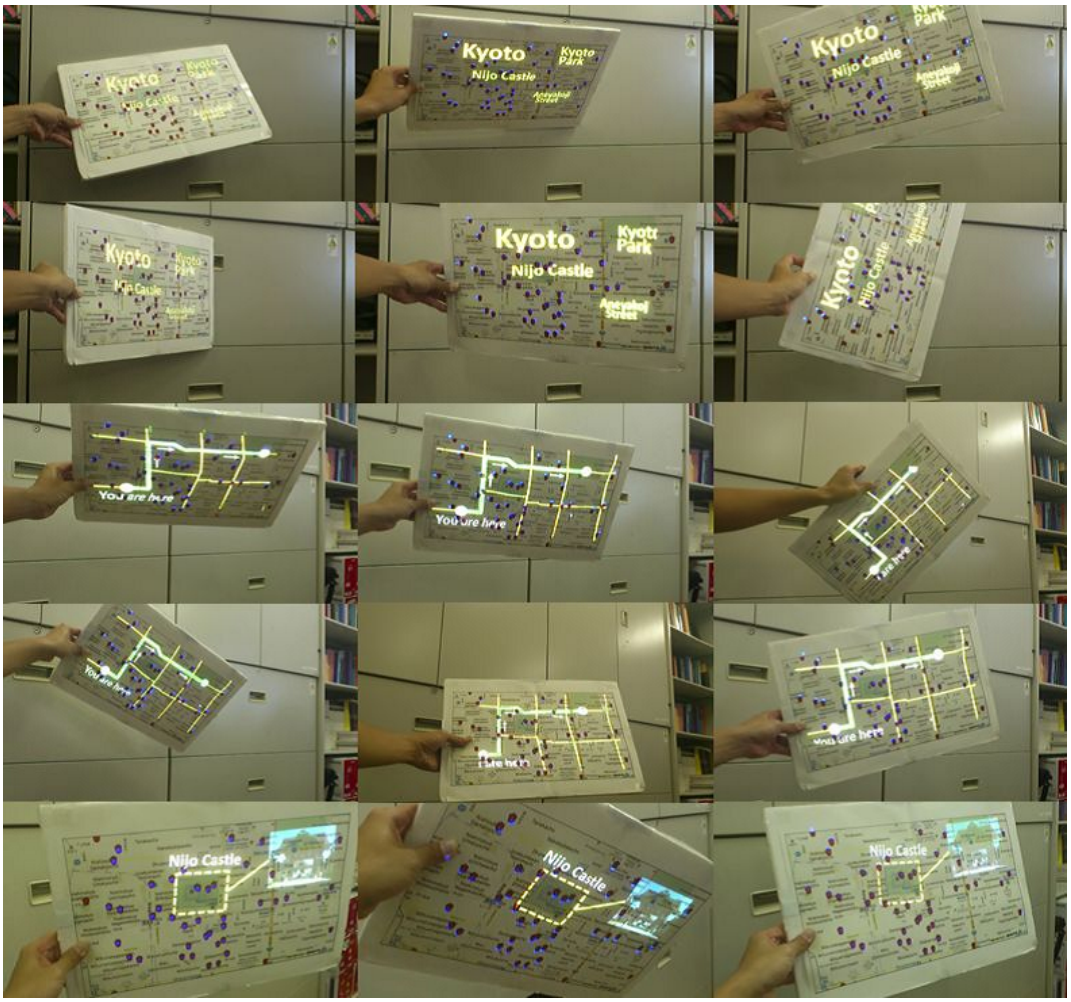


Figure 4.13: Projection mapping results. The proposed method can compensate arbitrary paper map motions.

Row 1 and 2 show the projected text `http://youtu.be/L43wVrn-M_0`

Row 3 and 4 show the projected navigation information

`http://youtu.be/tmG7U4I5aOA`

Row 3 and 4 show the projected photo or image

`http://youtu.be/31l99RF4lik`

90 degrees rotation 180 degrees rotation

Figure 4.14: Extreme rotation. The proposed alignment method can also compensate 90 to 180 degrees rotation.

Figure 4.15: Initial projection. The initial projection estimates the first alignment. Both the blue dots and red dots can be detected.

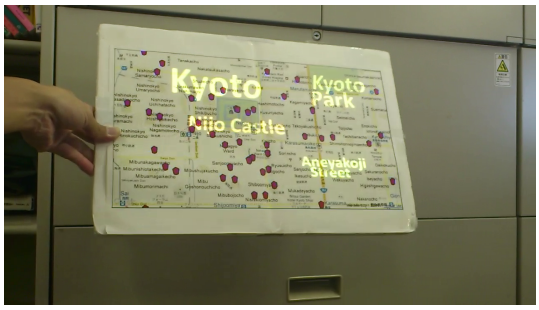


Figure 4.16: Overlapping state. When the alignment is successful, the projected blue dots overlap the red dots which cause projection hinder tracking problem. In this case, the previous successful alignment homography is used.

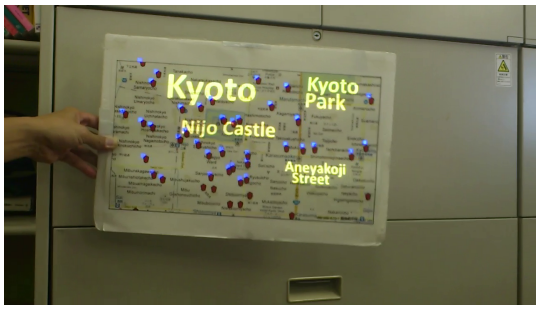


Figure 4.17: Dots reappear. When the paper or the camera or the projector moves, both red and blue dots reappear and can be detected. In this case, the new alignment homography is recomputed.

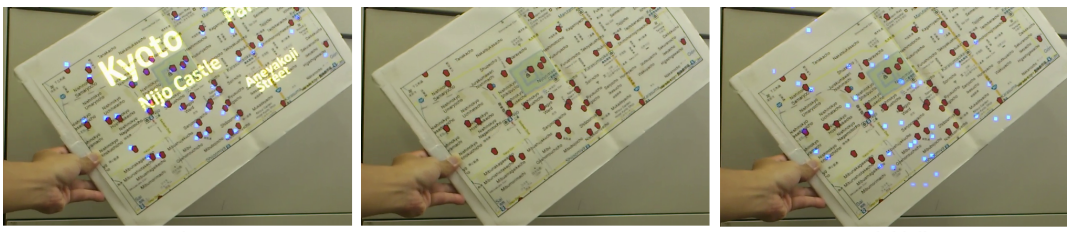


Figure 4.18: Failure due rotation. Although the alignment method works on rotation up to 90 degrees, there are some cases where the blue dots can not be detected.

Chapter 5

System Architecture and Applications

Augmented maps application can combine digital layers and paper map as introduced in previous chapters. Augmented maps overlays virtual data such as city names, region descriptions or 3D models of landmarks and buildings on top of paper map. In conventional augmented maps application, 3D models are prepared beforehand and dedicated for specific application. Augmented maps application is usually limited for particular area. When the user requires an augmented map for another area, the developer must prepare the contents beforehand.

To solve this issue, geographical data in the Internet such as Google Maps and 3D Warehouse are used. Google maps and 3D warehouse use the same geographic coordinate which allows augmentation without changing the coordinate system. In order to realize this idea, it is necessary to explore robust detection and tracking method for Google maps. Image analysis for image tracking such as map indexing is feasible by separating the map into layers such as roads, intersections or regions. However, usually those layers are not accessible and the user can only get the map as raster images. Therefore, further image analysis and data preparations are required in order to create trackable maps.

Instead of applying complicated image processing on a map, trackable maps are made by adding a tracking layer above the background map. Any features for tracking can be added regardless the type of maps. This chapter discusses a system architecture that includes a tool for retrieving trackable maps and 3D city models from Google Maps and 3D Warehouse. Using the proposed system architecture, the user can download desired maps and virtual contents on demand by simply inputting city name as illustrated in Figure 5.1. By printing the map and preparing

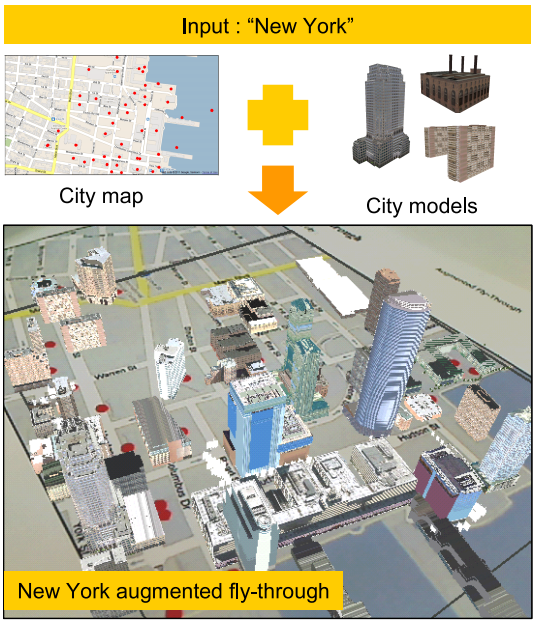


Figure 5.1: System architecture overview. First, the user inputs a city name into the data extraction tool, for instance ”New York”. The user then downloads the 3D models from the 3D warehouse. The user prints the map and views the overlaid 3D models on the map.

the augmented reality setup, the user can view 3D city models are superimposed in real time via monitor display or projection on the map.

5.1 Proposed System Architecture

The proposed system architecture focuses on map and 3D models access for making on demand augmented maps application. Google Maps and 3D Warehouse are used as data and contents resources. The flow of the proposed system is illustrated in Figure 5.2.

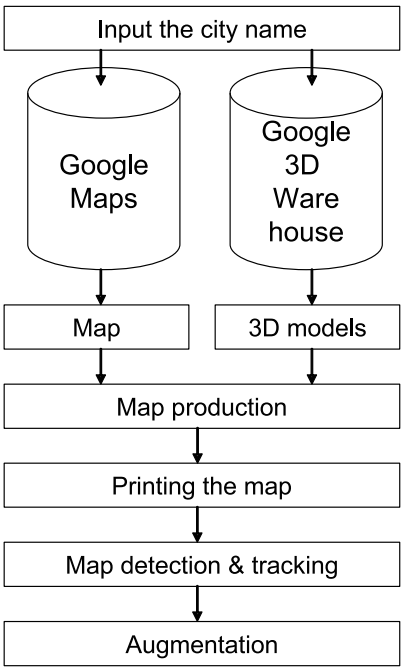


Figure 5.2: Flow of the proposed system architecture. First, the user inputs a city name. The system requests to servers for a map and 3D models. The trackable map is then made in the map production. The user then prints the map. At online phase, the paper map is detected and tracked. The 3D models are then superimposed.

5.1.1 Map Data

Maps are retrieved from Google Maps. Three types of map are available: default, terrain, and satellite maps as illustrated in Figure 5.3. The map for the system is created by combining 3D model locations as a tracking layer on top of a background map as illustrated in Figure 5.4. The tracking layer is defined as colored dots. This layer is then extracted back in the initialization step using color detection.

A tool for retrieving maps (tracking layer + background map) from Google Maps and 3D models from 3D Warehouse are built as illustrated in Figure 5.5. This tool receives a city name as the input and displays locations of 3D models on top of the background map the city as colored dots. The right panel beside the map enlists the name of the model appears on the map. The user can download the local database of the city map with the associated 3D city model. The output

of this tool are a trackable map, a set of 3D building models inside the map area and a text file contains the position of 3D building in geographic coordinate. The text file is used for detecting the paper map in the initialization step.

5.1.2 3D Data

Google 3D Warehouse allows users to create and share their 3D models. Because all 3D models in Google 3D Warehouse are made in geographic coordinate, they can be superimposed onto map that also uses geographic coordinate without performing coordinate transformation. 3D models of buildings and landmarks in a city are used as the virtual contents. Currently, the user downloads the maps and 3D models before runtime. However, it is also possible to extend this approach by downloading 3D models during runtime.

5.2 Results and Evaluation

The outputs of the system architecture are the augmentation of any locations in the world as illustrated in the Figure 5.6. The virtual contents fully depend on the models availability in Google 3D warehouse.

For tracking, four detection and tracking methods are applied: random dots markers, SIFT, SURF, and random ferns. Successful tracking based on the map and the computational cost are estimated. For experiments, a web camera with resolution 640×480 and the map in A4 size paper are used.

The system is implemented using OpenCV library [66]. The city models are loaded using Open Asset Import Library (Assimp)[2]. The camera is calibrated using the Calibration tool [4] that is based on the implementation of Zhang calibration method [94]. For experiments, a laptop computer with specifications: Intel I7 Quad Core 2.80GHz and 4GB memory is used.

5.2.1 Tracking Robustness

This section shows the percentage of successful tracking on image sequences contains the image map. For this experiments, a Kyoto map that consists 84 mesh models is used. Accordingly, 84 dots exist on the map. Three image maps are prepared: default, terrain and satellite maps. Those three maps are captured using web camera and recorded as image sequences. Each tracking method is then applied on the image sequences. Successful tracking occurs if the system can detect

the map on the image sequences. The border of the map is then re-projected using the homography to the map image. The frame of which the projected border that is near to the actual border of the map in the paper is counted divided by the number of frames as illustrated in the Figure 5.7.

According to the results, texture-based tracking using SIFT, SURF and random ferns are robust on default and terrain maps. Note that default and terrain map have strong edges and distinctive colors that help the successful tracking. On the other hand, the robustness of tracking drops on the satellite maps. In the satellite maps, the texture is relatively uniform that makes the matching becomes difficult.

As expected, the random dots marker technique could track the paper maps regardless the type thanks to the tracking layer. Surprisingly, the tracking robustness even increased on the satellite maps . The dots become distinctive enough on the satellite map to make them easy to extract.

5.2.2 Computational Cost

The computational cost of the proposed application is examined during the matching and rendering process. The average time for matching the map with the reference map and rendering the models as listed in Table 5.1.

Table 5.1: Computational cost based on the type of the map and tracking method.

Method	default map (ms)	terrain map (ms)	satellite map (ms)
Random dots marker	45.41	52.9	51.49
SIFT	853.39	844.91	926.02
SURF	462.61	440.06	827.46
Random ferns	174.26	168.44	202.14

The results show that random dots method works faster than the other method because it depends only on the color detection. In addition, it uses a hashing technique for fast descriptor lookup. On the other hand, the matching method that utilizes the texture information such as SIFT, SURF and random ferns requires longer time. SURF has better performance than SIFT thanks to the integral image approach. Random ferns method works best among the three methods. However, random ferns method requires around 10 minutes learning or building database beforehand. This 10-minutes-long learning is not preferable because preparing

learning data every time the user download the map will make the application becomes impractical.

5.2.3 Comparison of Random Dots and Texture-based Method

In experiments, tracking using SIFT, SURF and random ferns can work robustly for default and terrain maps. Those methods are suitable for default map and terrain maps. On the other hand, random dots marker technique can be the alternative for tracking satellite maps.

Moreover, random dots marker technique contributes less time than the other method thanks to the simple extraction method and matching using hash table. The computational cost is significant for deciding the suitable tracking method for augmented maps. Comparison of another feature descriptor for tracking paper map such as BRIEF [21], or GLOH [61] and its variant are the next step of this research. Furthermore, it is interesting to explore on combining the random dots marker technique with the other texture based method for realizing the best tracking method for augmented maps.

Technically, a tracking layer is added on a background map for initialization. Currently, colored dots are overlaid as tracking layer over the background map. In order to create more realistic map, instead of colored dots, the colored icons can be used.

Instead of adding tracking layer on the background map, different approach can be explored in the future by extracting specific features on the original map. Because there are terrain and satellite map, feature extraction will be different for each type and the generality for each type of map will lose. However, defining features for particular type of map is interesting issue to explore.

The application shows that the geographic coordinate can be used as shared space for augmented reality. However, current implementation only covers viewing workspace of the collaboration. To add the individuality features for collaborative AR environment [41], it is necessary to let user to create and modify the shared 3D models. Thus, each user can view coherent 3D models on their site.

5.3 Implementation on Mobile Phones

This section describes the implementation of the augmented maps application on mobile phones. The user can use the augmented maps application even in outdoor places. Furthermore, nowadays many people use mobile phones for supporting

their daily activities. As a result the augmented maps application can be widely used by many people. The scenario of augmented maps on mobile phones is illustrated in Figure 5.8.

This section also compares the implementation on a desktop computer and a mobile phone.

5.3.1 Map Feature Extraction

The map used in the mobile version of augmented maps application is made similarly to the desktop version. Generally, the background of digital maps are in pale or light colors. In order to perform the map registration, the feature extraction using color separation is applied. Then binary image is created and the blob detection is applied. The center of each blob is the keypoint for estimating the correspondence between the captured map with the reference map.

5.3.2 Overlaying Virtual Contents

The virtual contents are overlaid on the screen using texture mapping method. The geographic information including geographic layers or another digital maps. Two layers such as river or aerial map layer are used as illustrated in Figure 5.9.

5.3.3 Results

The screen shot of augmented maps on a mobile phone is shown in Figure 5.10. The user can hold the printed map using one hand and the other hand holds the mobile phone. The overlaid information can be viewed on the mobile phone display.

The setting for the implementation in listed in Table 5.2. The captured image size on mobile phone is 864×480. The image is then scaled into 0.3 smaller size than the captured image. The scaled image is then used for the map detection and tracking. For image overlay, the texture images are resized into 0.5 smaller than the original in order to reduce the computation time for texture mapping. For detecting bended surface, the mesh size is reduced into 8×6 (8 rows $\times$ 6 columns of patches). To optimize the speed the double precision floating for computation is avoided.

Table 5.2: The setting comparison on a mobile phone and a desktop computer.

Parameters	Mobile phone	Desktop computer
Image size	864×480	640×480
Scaling ratio	0.3	0.4
Mesh size for bended surface	8×6	10×8
Texture image size scale	0.5	1
Precision	single	double
Active descriptors	300	1000

5.3.4 Evaluation

The performance is evaluated using a mobile phone: Sony Ericson EXPeria Arc, resolution 864x480 pixel, and memory RAM: 512MB. The codes are implemented in C++ using OpenCV [66] and compiled in Java Native Interface (JNI) code for Android platform. The Android platform uses Java code and it calls the function defined in JNI to process the image, extract and match keypoints with the database. The output of the JNI function is the estimated camera pose. The camera pose is then used for augmenting the geographical information. For comparison, the application is build on desktop computer with specifications: Intel (R) Core (TM) i7 CPU M 640 2.80GHz, 4GB RAM and 640×480 pixel camera with setting listed in Table 5.2.

Tracking results

For evaluating the detection and tracking method, a white paper that consists of black colored dots is used. For proving that the application works on mobile phones, some tests on several relative position of the mobile phone to the paper are performed. The tests also include multi maps detection. In addition, the application is tested using real maps. The screen-shots of the visualization in a mobile phone are displayed on the Figure 5.11.

The results show that the application technically works on mobile phones. Even though the camera is positioned in tilted conditions, keypoints are extracted successfully in sufficient numbers. Therefore, the map can be successfully tracked. The robustness to occlusion and multi detection are also proven.

Computation time

The computation time on a mobile phone and desktop computer are compared in Table 5.3. The table shows that the computation on the desktop computer is obviously faster since the specification is higher than the mobile phone (RAM). However, by modifying the input image size and computation parameters, computation time could be reduced in order to achieve a real time application.

Table 5.3: Computation time on desktop and mobile environment. Computation time is mainly required on the visualization.

Process	Mobile phone (msec)	Desktop computer (msec)
Detection and tracking	52	5
Folding	89.8	12.04
Bending	73.5	5
Overlaying texture on folded paper	199.1	41.7
Overlaying texture on bended paper	343.6	26

Table 5.3 shows the cumulative computation time. There are three categories: detection and tracking, folding or bending, and overlaying texture. The folding or bending includes the detection and tracking computation time. The detection and tracking process is approximately ten times slower than desktop computer implementation due to limited memory and processor specifications. However, processing a frame in a mobile phone (camera capturing, detection and tracking, augmenting border) can be performed up to 14fps for real map. This result is sufficient for real time application because capturing and displaying on mobile phone (without any processing) is performed in 18fps.

The speed decreases during the overlaying process. In the current implementation, a raster scan-based warping method that checks each pixel value of the image and copy to the output image is applied. This warping involves a process of traversing all pixels on the image, thus the complexity depends on the image size. However, improvement can be performed by reducing the computation time using an efficient method such as texture mapping in OpenGL. In addition, hardware accelerated supported device can be an option to speed up the overlaying process.

5.3.5 Scenario

Geometrical change on physical paper including folding and bending are implemented on mobile phones as well.

Folding

An application for folded maps by displaying multiple contents of the map is implemented as shown in Figure 5.12. When the user views the map using the camera in a planar map, one content is overlaid. When the user folds the map, two types of content are superimposed on each folded area.

Bending

A texture is overlaid on a bended map. This scenario is suitable for a normal paper because a paper tends to bend. 2D image is overlaid as illustrated in Figure 5.13.

When the user bends the printed map in left or right direction, content of the map will change. The 2D layers are superimposed following the shape of the paper map. The geographic information such as river layer or hotel location are overlaid using texture mapping applied for each patch in the surface.

5.4 Interaction

Besides folding and bending as the trigger of actions, interaction using camera and finger gesture are also studied such as pointing and tapping. The simplest method for accessing data on a paper map can be realized using the center of the camera as the pointer. In addition, the application recognizes the user's finger tip and uses the location of the fingertip to access and visualize information.

5.4.1 Camera-based Pointing

An augmented fly-through application is developed to allows user to browse and view the 3D city models through camera or HMD using a piece of paper map. The user can select the information on the map, by moving the camera as a pointer (see Figure 5.14). The building name as the virtual information appears when the center of the camera approaches the models.

5.4.2 Finger-based Pointing

After a paper is detected or tracked, finger tip is detected from an input image. Because the border of the detected paper is computed in order to render AR contents, the finger tip can be detected inside the border. In order to extract a hand region, the simplest but accurate enough HSV color space classificatory is applied. By thresholding HSV, a mask image of the hand region is created.

The detection is performed when the user points somewhere on the paper as illustrated in Figure 5.15(a). The user hand has to pose a pointing gesture and dorsal part should appear in the image entirely. The upper end of the hand is detected as the finger tip. The finger tip is obtained by computing the center of gravity of the hand region and finding the farthest point from the center as illustrated in Figure 5.15(b).

5.4.3 Tapping

Besides pointing interaction, tapping interaction can also be applied using the trajectory of the finger movement. Similar to the pointing interaction, for finding the position of the user's finger, color skin detection is applied.

Tapping gesture is defined as the fingertip movement that forms check or **v** sign as shown in Figure 5.16. This gesture is defined as a metaphor of finger tapping on a surface resembling button pressing gesture. Tapping interaction takes into account the angle calculation between two directions of fingertip.

5.4.4 Accessing Related Data

The user can access the related data of each map symbol by pointing or tapping a dot in the map. Because we assume that the user actually touches the map while pointing or tapping, the nearest map symbol at the finger tip is selected as the pointed symbol in the image.

The pointing interaction is defined by observing the position of finger tip. If user's finger tip stays near to a map symbol constantly in several frames, then that map symbol is stated as pointed. Whereas the tapping interaction is defined by examining the trajectory of the finger tip.

The data related to the symbol is overlaid after the pointing or tapping occurs. As an example of implementation, a photo is overlaid when a map symbol is pointed as illustrated in Figure 5.17.

5.4.5 Interaction on Mobile Phones

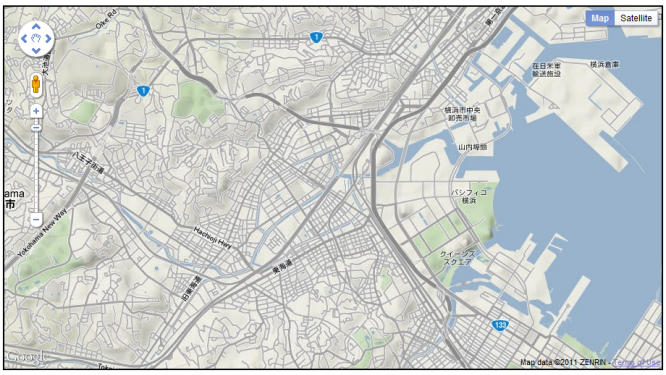
The interaction can also be applied on the display of mobile phone. Current-generation mobile phones have touch sensitive display that lets the user interact with the application using finger tapping. Using this interaction, the augmented maps application on mobile phones is capable of receiving input command via finger tapping. A scenario of the augmented maps application on mobile phones lets the user to select data on the superimposed data such as pictures or icons as illustrated in Figure 5.18.

5.5 Summary

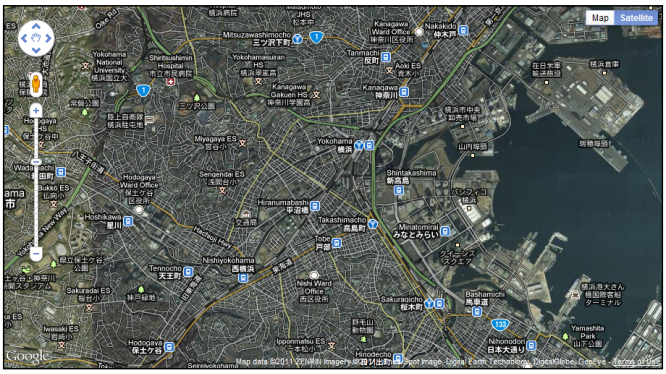
The system architecture for making the augmented maps application using online geographical data is introduced. The users can use maps and 3D model database that is made and shared by other users. Therefore, they can make augmented maps for any location using the system architecture. The characteristics of map and its usage for developing augmented maps are also studied. Finally, it is shown that random dots marker technique is suitable for building the proposed system in terms of the minimum computational cost.

Providing the efficient way for retrieving maps and virtual data for augmented maps remains the main challenges in this work. In the future, on-line connection to the database server and the cloud architecture are promising outlooks in order to access the maps and virtual data. Furthermore, occlusion handling and realistic rendering for augmented maps are also required.

(a) ©2011 Google - Map Data ©ZENRIN



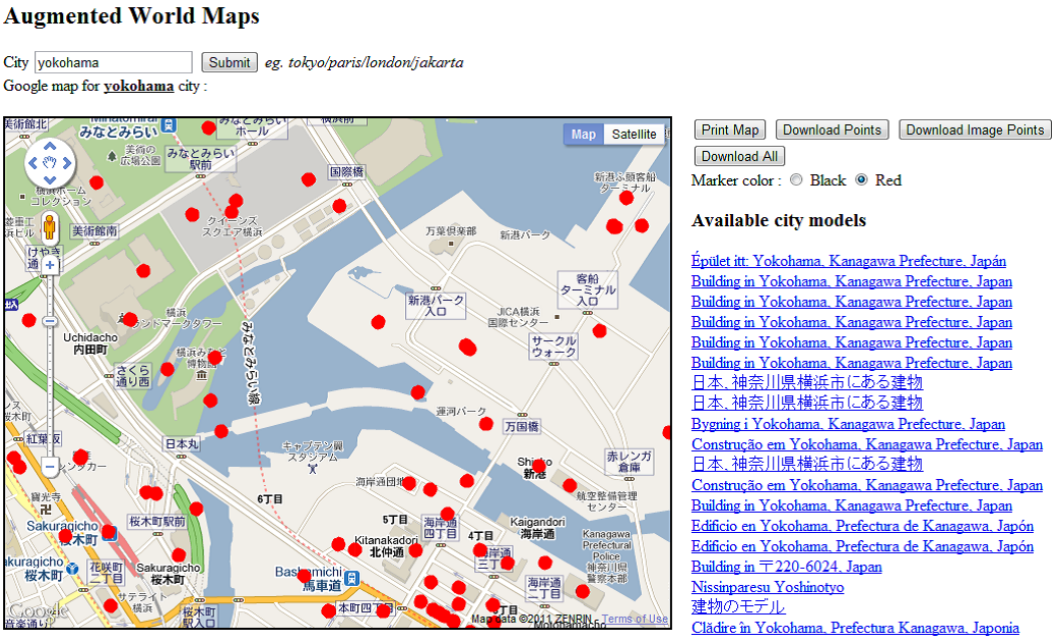
(b) ©2011 Google - Map Data ©ZENRIN



(c) ©2011 Google - Imagery ©2011 Cnes/Spot Image, Digital Earth Technology, DigitalGlobe, GeoEye, Map data ©2011 ZENRIN

Figure 5.3: Three types of map. (a) Default map. It consists of some labels. (b) Terrain map. It consists dense edges and lines. (c) Satellite map. The real captured image from the satellite.

Figure 5.4: Map production flow. First, a city name is queried. The tool extracts the map and 3D models of the city. The local database for initialization is then built. The 3D model positions are overlaid as the colored dots.



©2011 Google - Map Data ©2011 ZENRIN

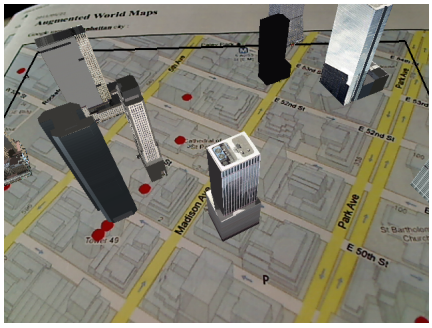
Figure 5.5: A tool for extracting city map from Google Maps and 3D warehouse. The red dots on the map represent the location of 3D building models. The list on the right panel shows the available 3D building models.

Map Data ©2011 Google, Sanborn

(a)

Map Data ©2011 ZENRIN

(b)



Map Data ©2011 Google

(c)

Map Data ©2011 Google, Sanborn

(d)

©2011 Google

Figure 5.6: Visualization results. 3D building models are superimposed on top of printed maps. (a) Honolulu (b) Kobe (c) Park Avenue (d) San Diego.

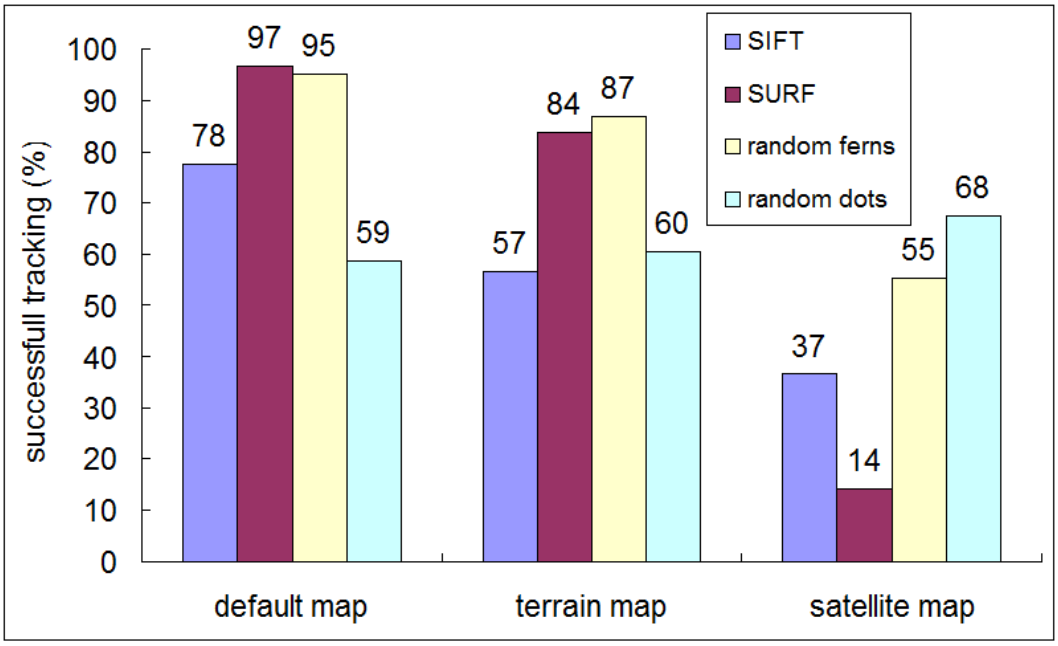


Figure 5.7: Successful tracking rate. Random dots marker technique works on any type of map.

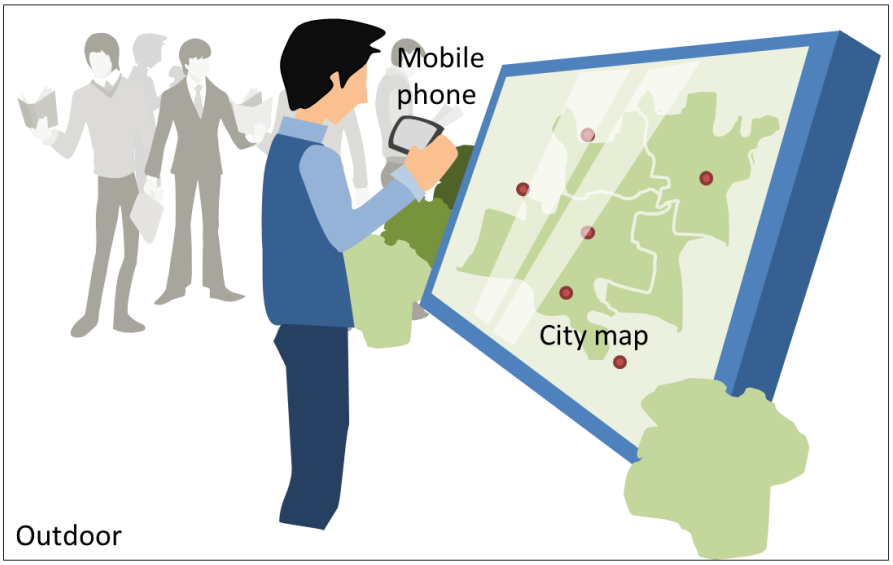


Figure 5.8: Scenario using mobile phones. The augmented maps application is used to view virtual information over maps that is placed in the public space like streets or train stations. The user can see dynamic information that is not printed on the maps on the mobile phone's display.

Figure 5.9: (a) Example of vector layer (river). (b) Example of raster layer (aerial map).

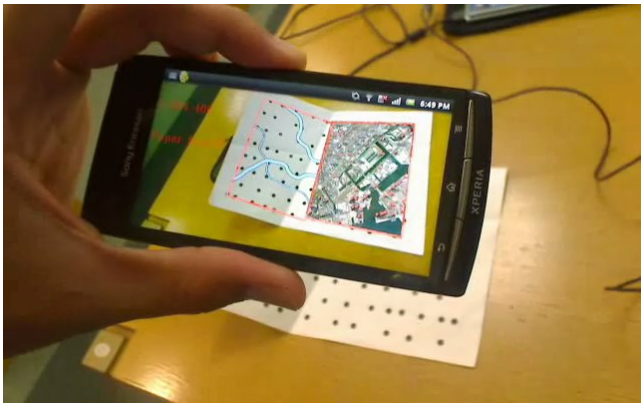


Figure 5.10: Implementation on mobile phones. A paper map is placed in front of the mobile phone's camera. The user can view geographic information on the mobile phone's display.

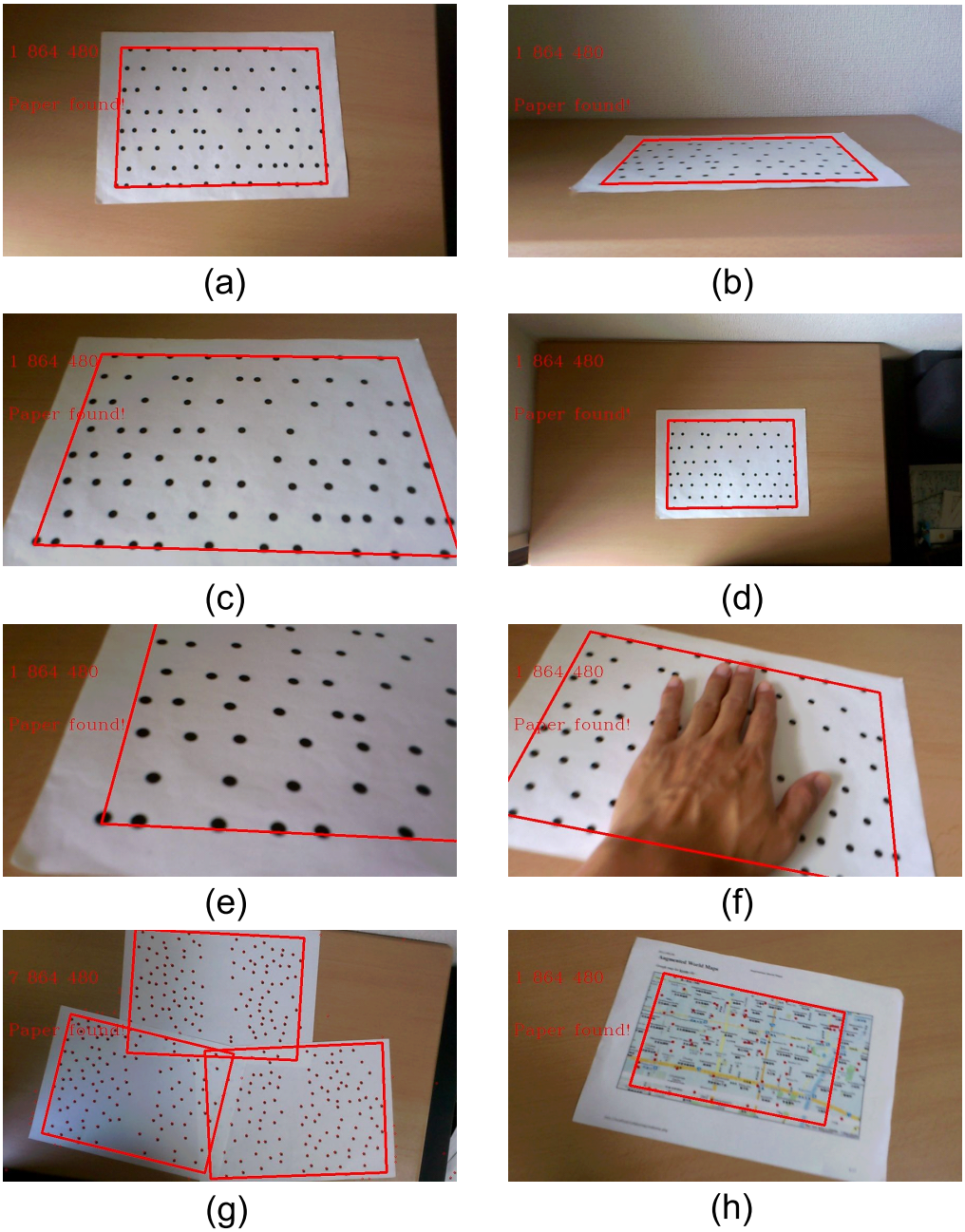


Figure 5.11: Tracking results on a mobile phone. (a) Top view image. (b) Tilted. (c) Mobile phone camera is located close to map. (d) Mobile phone camera is located far from the map. (e) and (f) Map is occluded. (g) Multiple maps detection and tracking. (h) A real map with colored symbol.

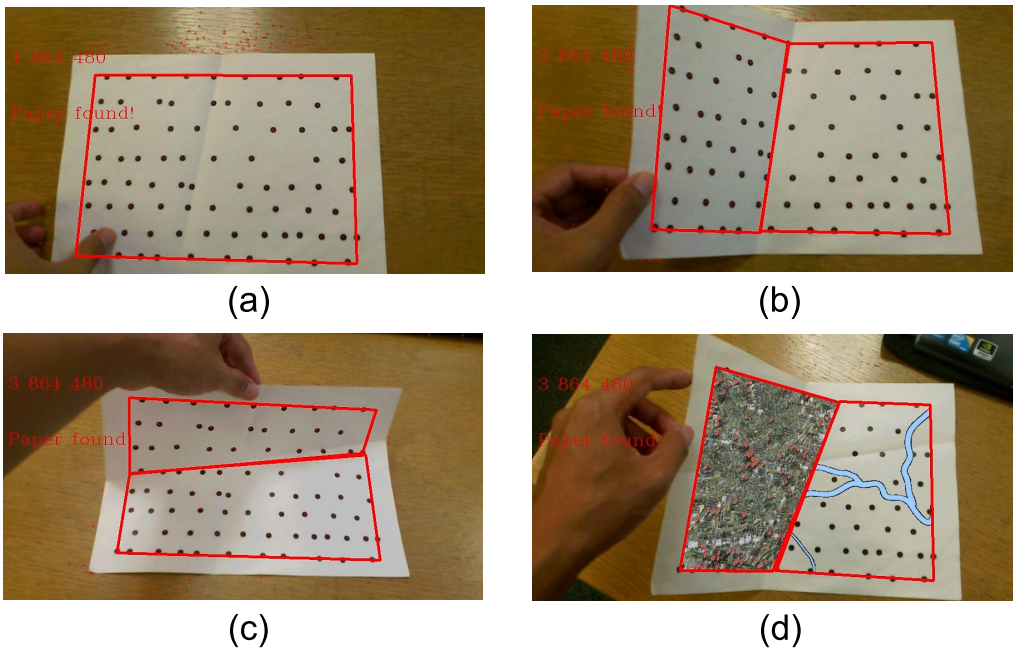


Figure 5.12: Scenario for folded maps. (a) User places the mobile phone camera on top of the map. The user starts to fold the map. (b) The map is folded. The area is divided into two planes. (c) Folding in other direction. (d) Different contents are superimposed in each plane.

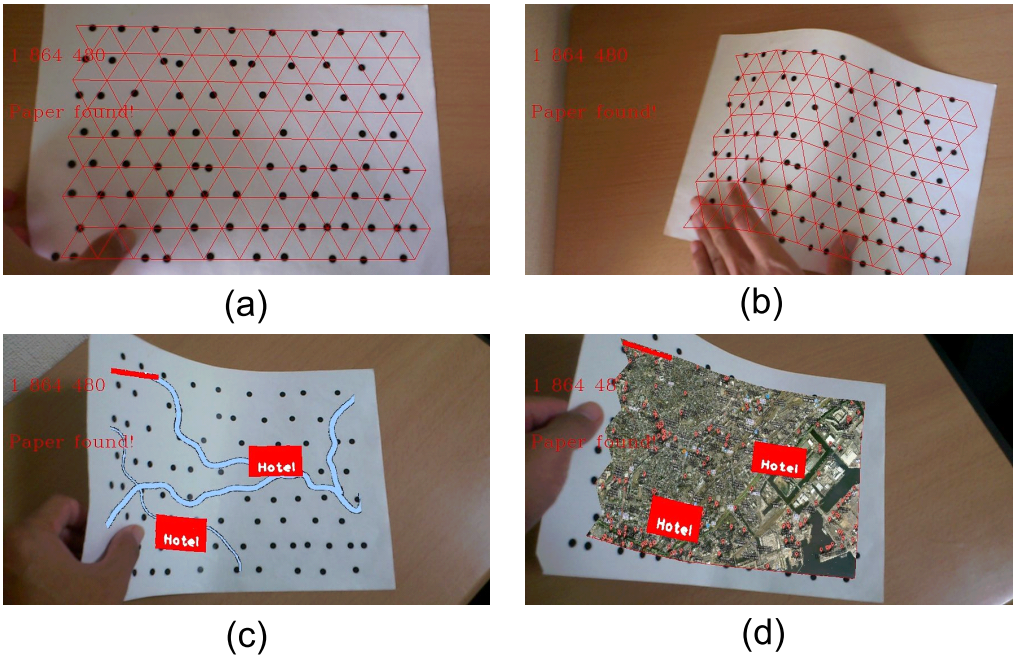


Figure 5.13: Scenario for bendable augmented maps. (a) The user puts the camera on top of the printed map. The mesh is overlaid. (b) The user bends the paper. The mesh is overlaid following the shape of the paper. (c) and (d) When the user bends the map in left or right direction, content of the map will change. The geographic information are the river layer, location of hotels or the aerial map.

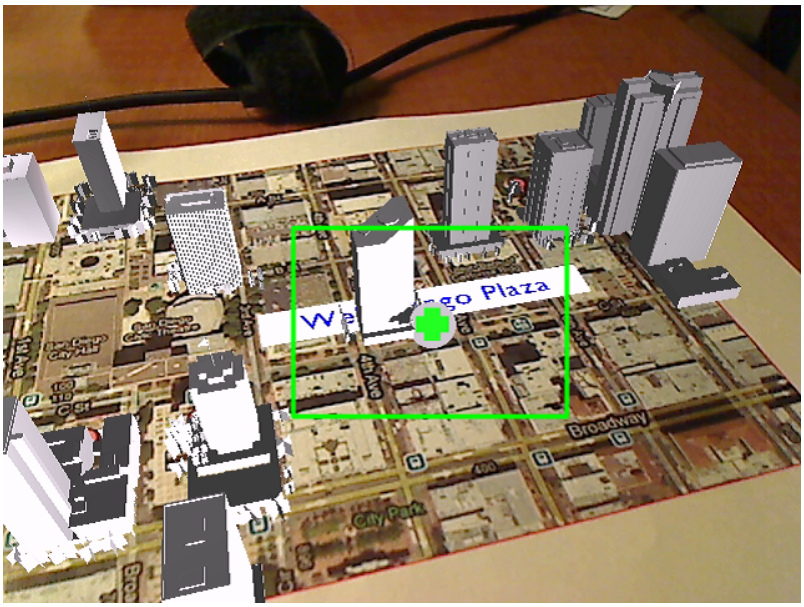
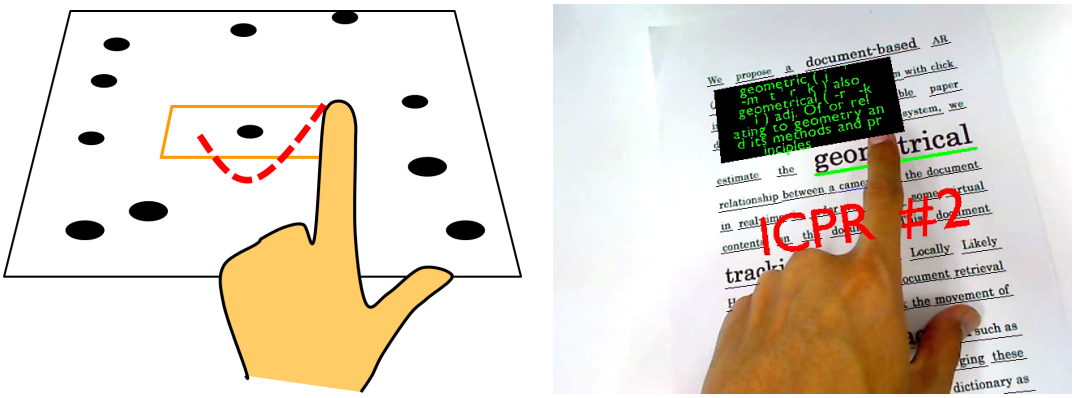


Figure 5.14: Data selection using center of the camera image. The user moves the camera to select and display the information of the San Diego map. The name appears when the center of the camera image approaches a building model.

(a) (b)

Figure 5.15: Finger tip detection. (a) User's pointing. A user points a map symbol with touching the paper map. (b) Definition of finger tip position. The farthest point from the center of the hand region is defined as a finger tip.



(a) **v** trajectory for tapping (b) Tapping result

Figure 5.16: Tapping interaction. The tapping result shows the interaction is applied on a augmented dictionary application using random dot markers. When the user taps a document on a particular word, the meaning of the word is superimposed.

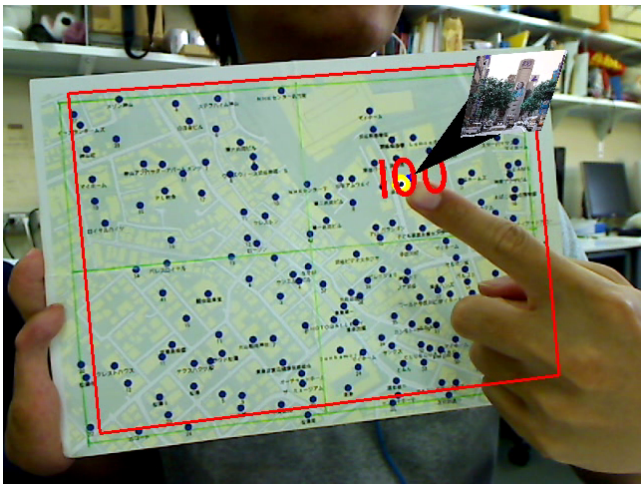
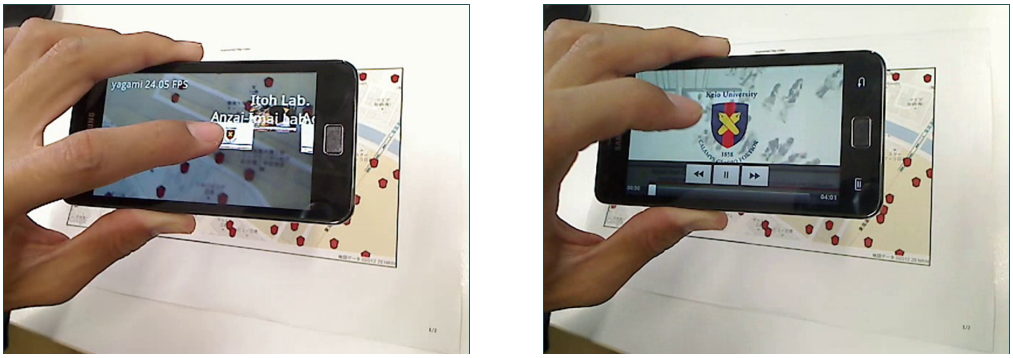


Figure 5.17: Accessing the data of each symbol. A photo is overlaid when a map symbol is pointed. The ID and a picture of a popular spot near the map symbol are example contents.



(a) Tapping on display (b) Displaying a movie of selected location

Figure 5.18: Tapping interaction on mobile phones. Current-generation mobile phone is equipped by touch display that accommodates the user to select a particular location in the map using tapping. (b) shows a movie related to a place in the map is being displayed.

Chapter 6

Conclusion

This thesis explored the enhancement of physical paper using the augmented reality approach. Conventional augmented reality applications only handle physical paper in planar. However, in daily life, when the user holds the paper, the paper may be folded or bended in order to follow the user's hand.

There are two proposals that had been explored. Firstly, this work enhances the physical paper by changing its geometrical properties. This thesis solved technical problems of recognizing the changes in the geometrical properties of paper including folding, bending and cutting. By folding, bending, and cutting a piece of paper, virtual contents can be visualized according to the shape or structure of the paper. Secondly, this work enhances the physical paper by projecting virtual contents directly on the paper using a projector-camera setup. By projecting directly onto physical paper, the user can view the virtual content on the paper and interact with the real paper. Additionally, a system architecture for making on demand augmented maps is explored.

6.1 Contributions

This thesis presented three major technical contributions:

- Folding, bending, and cutting-based interaction on physical paper.
- Feature-based alignment method for projector-camera setup.
- A system architecture for retrieving maps and virtual contents from the Internet for building on demand augmented maps application.

In this thesis, the random dot marker technique is used because it is robust. The random dot marker method can perform fast as the result of the search mechanism on hash table that can be done in real time. The random dot markers method employs the distribution of keypoints and it can contribute to high matching accuracy. Hence, the partial occlusions on the paper will not affect the detection and tracking. Furthermore, random dot marker method is able to detect and track multiple planes that is suitable for achieving the goal of this thesis.

In folded surface detection, a planar paper is divided into multiple planes. By iterating the plane detection using the random dot marker method, each plane can be detected and tracked. As a result, a folded piece of paper can be detected and tracked. In bended surface detection, a reference planar is deformed in order to approximate the shape in the input image. For cutting detection, the problem is simplified into region detection. By using keypoints distribution in the outline of region, each region is tracked.

On the second contribution, in order to align the virtual contents to a moving piece of paper, the relationship of the projector, camera and paper should be estimated. In order to estimate the relationship, random dots are initially projected onto the paper and then the transformation from the projected dots to printed dots is estimated.

The last contribution is the development of system architecture for making on demand augmented maps application. In the system, the user can retrieve maps and virtual contents of any location from the Internet. For map detection and tracking, several type of maps and tracking method were explored to demonstrate the tracking performance of the method. The system architecture is implemented on desktop computer and mobile device. For both implementation the projector-camera setup can be used by applying the the alignment method proposed in the second contribution.

Furthermore, in order to add the interactivity of the system, pointing and tapping interaction were explored. The pointing interaction is performed using camera center and fingertip. The user can point a particular location of the paper map in order to access the data related to that location. Tapping gesture that is commonly performed in a mobile phone is also implemented in order to access the data of the physical paper directly on the screen of the device. The system architecture and the interactions can be applied in making augmented maps application for showing localized data such as places information, videos and photos of areas and also navigation purposes.

6.2 Future Works

This thesis have initially started to overcome the issue of geometrical change on physical paper. The current implementation of the geometrical change on physical paper are folding, bending and cutting. Each implementation of the contribution can be extended for future works. Theoretically, the number of folding in the proposed folding model is not limited to two or three planes. However, since the tracking method requires sufficient number of keypoints, current implementation allows only folding into three planes. As the improvement, the current implementation can be extended into multiple folding such as origami. In origami interaction, user's hands occlude the paper frequently. Therefore, a research direction on occlusion handling especially when dealing with physical paper or arbitrary surfaces can be added.

There are many possibilities of interaction on a piece of paper besides folding, bending and cutting. Some explorations has given clues of future paper interaction [40, 49]. Creasing, crumpling, collocating, or stapling are some examples of intuitive interactions on physical paper which can be explored in the future. The current implementation explore the solution for folding, bending and cutting in the monocular solution where the setup only consists of a single camera. With the help of recent devices such as depth cameras and motion or light sensors, some of the limitations of the current implementation can be reduced.

Arbitrary surfaces detection and tracking become more difficult when there are multiple objects or surfaces appear in the same scene. This case takes place when multi users perform a action simultaneously. The collaboration and social aspect can be the keyword on exploring new issues for future work.

In the proposed alignment method for projector-camera setup, the speed can be increased by removing the delay that was added intentionally. However, synchronization between projector and tracking becomes a major issue because the projection can hinders the tracking. Especially, when the virtual contents are rich textured images. Another registration method such as texture-based method is interesting to explore for advancing this work. However, the projection-hinder-tracking problem will eventually occur and thus an effective solution for this problem remains to be the major issue.

Paper as the ubiquitous media has brought interesting issues to explore. The idea of recognizing any surface in the real world and use it as a display has been partially achieved in this thesis. However, recognizing and registering the book shape and its content for instance, has not been realized yet. In order to register any surfaces in the real world, a robust registration method is required. Moreover,

projecting the virtual contents on arbitrary surfaces and objects can also be an interesting issue to solve.

In this research, a system architecture for building on demand augmented maps has been presented. In the future, the content of the system architecture can be extended to more general contexts and not limited only to maps and geographical contents. Thus, the proposed tool can be reused for creating and building paper-based augmented reality for a wide range of surfaces and purposes.

Bibliography

[1] AR Sights. http://www.arsights.com/.

[2] Assimp. http://assimp.sourceforge.net/.

[3] Bing Maps. http://www.bing.com/maps/.

[4] Camera Calibration Tools. http://www.doc.ic.ac.uk/ dvs/calib/main.html.

[5] OpenVRML. http://openvrml.org/.

[6] *Multimedia Technologies*. McGraw-Hill Education (India) Pvt Limited, 2010.

[7] K. Akasaka, R. Sagawa, and Y. Yagi. A Sensor for Simultaneously Capturing Texture and Shape by Projecting Structured Infrared Light. In *International Conference on 3-D Digital Imaging and Modeling*, pages 375–381, August 2007.

[8] L. Alem and W. Huang. *Recent Trends of Mobile Collaborative Augmented Reality Systems*. SpringerLink : Bücher. Springer, 2011.

[9] M. Ambai and Y. Yoshida. Card: Compact and real-time descriptors. In *IEEE International Conference on Computer Vision*, pages 97–104. IEEE, 2011.

[10] S. Audet, M. Okutomi, and M. Tanaka. Direct image alignment of projector-camera systems with planar surfaces. *Image (Rochester, N.Y.)*, pages 303–310, 2010.

[11] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (SURF). *Computer vision and image understanding*, 110(3):346–359, 2008.

[12] H. Benko, R. Jota, and A. Wilson. Miragetable: freehand interaction on a projected augmented reality tabletop. In *ACM Conference on Human Factors in Computing Systems*, pages 199–208. ACM, 2012.

[13] O. Bergig, N. Hagbi, J. El-Sana, and M. Billinghurst. In-place 3D sketching for authoring and augmenting mechanical systems. In *IEEE International Symposium on Mixed and Augmented Reality*, pages 87 –94, October 2009.

[14] M. Billinghurst, H. Kato, and I. Poupyrev. The MagicBook: a transitional AR interface. *Computers & Graphics*, 25(5):745–753, 2001.

[15] O Bimber and R Raskar. *Spatial Augmented Reality: Merging Real and Virtual Worlds*, volume 6. 2005.

[16] P. Bo and W. Wang. Geodesic-controlled developable surfaces for modeling paper bending. In *Annual Conference of the European Association for Computer Graphics*, 2007.

[17] J. Bobrich and S. Otto. Augmented maps. In *International Society for Photogrammetry and Remote Sensing*, 2002.

[18] H. Brown and P. Robinson. Integrating paper and digital documents. *J. Vice, & R. Earshaw, Digital Media: The Future*, pages 128–143, 2000.

[19] Michael S. Brown and W. Brent Seales. Image restoration of arbitrarily warped documents. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10):1295–1306, 2004.

[20] F. Brunet, R. Hartley, A. Bartoli, N. Navab, and R. Malgouyres. Monocular template-based reconstruction of smooth and inextensible surfaces. *Asian Conference on Computer Vision*, pages 52–66, 2011.

[21] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. BRIEF: Binary Robust Independent Elementary Features. In *European Conference on Computer Vision*, September 2010.

[22] A. Dame and E. Marchand. Accurate real-time tracking using mutual information. In *IEEE International Symposium on Mixed and Augmented Reality*, pages 47–56, 2010.

[23] A. Del Bue and A. Bartoli. Multiview 3D warps. In *IEEE International Conference on Computer Vision*, pages 675–682. IEEE, 2011.

[24] M. Donoser, P. Kontschieder, and H. Bischof. Robust planar target tracking and pose estimation from a single concavity. In *IEEE International Symposium on Mixed and Augmented Reality*, pages 9 –15, October 2011.

[25] J. Ehnes and M. Hirose. Projected Reality - content delivery right onto objects of daily life. *International Journal of Virtual Reality*, 5(3):17–23, 2006.

[26] M. Emori and H. Saito. Texture overlay onto deformable surface using HMD. In *IEEE Virtual Reality Conference*, pages 221 – 222, 2004.

[27] M. Fiala. ARTag, a fiducial marker system using digital techniques. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 590 – 596 vol. 2, June 2005.

[28] B. Furht. *Handbook of Augmented Reality*. Springer Science+Business Media. Springer, 2011.

[29] V. Gay-Bellile, A. Bartoli, and P. Sayd. Direct estimation of nonrigid registrations with image-based self-occlusion reasoning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:87–104, 2010.

[30] L. Gruber, S. Gauglitz, J. Ventura, S. Zollmann, M. Huber, M. Schlegel, G. Klinker, D. Schmalstieg, and T. Höllerer. The City of Sights: Design, construction, and measurement of an Augmented Reality stage set. In *IEEE International Symposium on Mixed and Augmented Reality*, pages 157–163, 2010.

[31] F. Guimbretière. Paper augmented digital documents. In *ACM Symposium on User Interface Software and Technology*, pages 51–60. ACM, 2003.

[32] N. Gumerov, A. Zandifar, R. Duraiswami, and L. Davis. Structure of applicable surfaces from single views. In *European Conference on Computer Vision*, pages 482–496, 2004.

[33] N. Hagbi, O. Bergig, J. El-Sana, and M. Billinghurst. Shape recognition and pose estimation for mobile augmented reality. *Visualization and Computer Graphics, IEEE Transactions on*, 17(10):1369 –1379, October 2011.

[34] R.R. Hainich and O. Bimber. *Displays: Fundamentals, Applications, and Outlook*. Taylor & Francis Group, 2011.

[35] M. Haller, M. Billinghurst, and B.H. Thomas. *Emerging Technologies of Augmented Reality: Interfaces and Design*. ITPro collection. Idea Group Pub., 2007.

[36] W.J. Hansen and C. Haas. Reading and writing with computers: a framework for explaining differences in performance. *Communications of the ACM*, 31(9):1080–1089, 1988.

[37] C. Harrison, H. Benko, and A.D. Wilson. OmniTouch: wearable multitouch interaction everywhere. In *ACM Symposium on User Interface Software and Technology*, pages 441–450. ACM, 2011.

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

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

[40] D. Holman, R. Vertegaal, M. Altosaar, N. Troje, and D. Johns. Paper windows: interaction techniques for digital paper. In *ACM Conference on Human Factors in Computing Systems*, pages 591–599. ACM, 2005.

[41] A. Ismail and M. Sunar. Survey on collaborative AR for multi-user in urban studies and planning. In *Learning by Playing. Game-based Education System Design and Development*, volume 5670 of *Lecture Notes in Computer Science*, pages 444–455. 2009.

[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] Y.L. Kergosien, H. Gotoda, and T.L. Kunii. Bending and creasing virtual paper. *IEEE Computer Graphics and Applications*, 14(1):40–48, 1994.

[44] K. Kim, S. Oh, J. Lee, and I. Essa. Augmenting aerial earth maps with dynamic information. In *IEEE International Symposium on Mixed and Augmented Reality*, 2009.

[45] G. Kipper and J. Rampolla. *Augmented Reality: An Emerging Technologies Guide to AR*. Elsevier Science, 2012.

[46] M. Kobayashi and H. Koike. EnhancedDesk: integrating paper documents and digital documents. In *ACM Conference on Human Factors in Computing Systems*, pages 57 –62, July 1998.

[47] A. Kushal, J. Van Baar, R. Raskar, and P. Beardsley. A handheld projector supported by computer vision. In *Asian Conference on Computer Vision*, pages 183–192, 2006.

[48] J.C. Lee, S.E. Hudson, and E. Tse. Foldable interactive displays. In *ACM Symposium on User Interface Software and Technology*, pages 287–290, 2008.

[49] S.S. Lee, S. Kim, B. Jin, E. Choi, B. Kim, X. Jia, D. Kim, and K. Lee. How users manipulate deformable displays as input devices. In *ACM Conference on Human Factors in Computing Systems*, pages 1647–1656. ACM, 2010.

[50] V. Lepetit, P. Lagger, and P. Fua. Randomized trees for real-time keypoint recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 775–781. IEEE, 2005.

[51] A. Letouzey and E. Boyer. Progressive shape models. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 190–197. IEEE, 2012.

[52] M.C. Leung, K.K. Lee, K.H. Wong, and M.M.Y. Chang. A projector-based movable hand-held display system. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1109 –1114, June 2009.

[53] S. Leutenegger, M. Chli, and R.Y. Siegwart. BRISK: Binary robust invariant scalable keypoints. In *IEEE International Conference on Computer Vision*, pages 2548–2555. IEEE, 2011.

[54] M. Löchtefeld, J. Schöning, M. Rohs, and A. Krüger. LittleProjectedPlanet: an augmented reality game for camera projector phones. In *Artificial Intelligence*.

[55] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60:91–110, 2004.

[56] R. Mannings. *Ubiquitous Positioning*. Artech House Mobile Communications. Artech House, 2008.

[57] M.R. Marner. Digital Airbrushing with Spatial Augmented Reality. In *International Conference on Artificial Reality and Telexistence*, 2010.

[58] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767, 2004.

[59] D. McGee, X. Huang, P. Barthelmess, and P. Cohen. Poster: The NetEyes collaborative, augmented reality, digital paper system. In *IEEE Symposium on 3D User Interfaces*, pages 145 –146, March 2008.

[60] P. McIlroy, S. Izadi, and A. Fitzgibbon. Kinectrack: Agile 6-dof tracking using a projected dot pattern. In *IEEE International Symposium on Mixed and Augmented Reality*, 2012.

[61] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:1615–1630, 2005.

[62] M. Moll and L. Van Gool. Separating rigid motion from linear local deformation models. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop*, pages 37–44. IEEE, 2011.

[63] A. Morrison, A. Oulasvirta, P. Peltonen, S. Lemmela, G. Jacucci, G. Reitmayr, J. Näsänen, and A. Juustila. Like bees around the hive: a comparative study of a mobile augmented reality map. In *ACM Conference on Human Factors in Computing Systems*, pages 1889–1898, 2009.

[64] 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.

[65] S. Nishizaka, T. Narumi, T. Tanikawa, and M. Hirose. Detection of divided planar object for augmented reality applications. In *IEEE Virtual Reality Conference*, pages 231 –232, March 2011.

[66] OpenCV. OpenCV. http://sourceforge.net/projects/opencvlibrary/.

[67] J.O.M. Östlund, A. Varol, and P. Fua. Laplacian meshes for monocular 3D shape recovery. In *European Conference on Computer Vision*, 2012.

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

[69] V. Paelke and M. Sester. Augmented paper maps: Exploring the design space of a mixed reality system. *International Society for Photogrammetry and Remote Sensing*, 65:256–265, 2010.

[70] M. Perriollat and A. Bartoli. A quasi-minimal model for paper-like surfaces. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 0, pages 1–7, 2007.

[71] M. Perriollat, R. Hartley, and A. Bartoli. Monocular template-based reconstruction of inextensible surfaces. In *British Machine Vision Conference*, 2008.

[72] J. Pilet, V. Lepetit, and P. Fua. Fast non-rigid surface detection, registration and realistic augmentation. *International Journal of Computer Vision*, 76:109–122, 2008.

[73] R. Raskar, G. Welch, and H. Fuchs. Spatially augmented reality. In *IEEE and ACM International Workshop on Augmented Reality*, pages 11–20, 1998.

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

[75] J. Rekimoto and K. Nagao. The world through the computer: Computer augmented interaction with real world environments. In *ACM Symposium on User interface and Software Technology*, pages 29–36. ACM, 1995.

[76] S. Robinson, M. Jones, E. Vartiainen, and G. Marsden. PicoTales. In *ACM Conference on Computer Supported Cooperative Work and Social Computing*, page 671, 2012.

[77] M. Rohs, J. Schöning, A. Krüger, and B. Hecht. Towards real-time markerless tracking of magic lenses on paper maps. In *adjunct Proc. Pervasive*, pages 69–72, 2007.

[78] E. Rukzio and P. Holleis. Projector phone interactions: Design space and survey. In *Workshop on Coupled Display Visual Interfaces*, 2010.

[79] M. Salzmann and P. Fua. Reconstructing sharply folding surfaces: A convex formulation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1054 –1061, June 2009.

[80] M. Salzmann and P. Fua. Linear local models for monocular reconstruction of deformable surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):931–944, 2011.

[81] M. Salzmann, F. Moreno-Noguer, V. Lepetit, and P. Fua. Closed-form solution to non-rigid 3D surface registration. In *European Conference on Computer Vision*, pages 581–594, 2008.

[82] C. Scherrer, J. Pilet, P. Fua, and V. Lepetit. The haunted book. In *IEEE International Symposium on Mixed and Augmented Reality*, pages 163–164, 2008.

[83] T. Siriborvornratanakul and M. Sugimoto. A Portable Projector Extended for Object-Centered Real-Time Interactions. In *Conference for Visual Media Production*, pages 118–126, November 2009.

[84] J. Steimle, M. Mühlhäuser, and J.D. Hollan. *Pen-and-Paper User Interfaces: Integrating Printed and Digital Documents*. Human–Computer Interaction Series. Springer, 2012.

[85] N. Takao, J. Shi, S. Baker, I. Matthews, and B. Nabbe. Tele-Graffiti: A pen and paper-based remote sketching system. In *IEEE International Conference on Computer Vision*, volume 36, page 750, 2001.

[86] J. Taylor, A.D. Jepson, and K.N. Kutulakos. Non-rigid structure from locally-rigid motion. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2761 –2768, June 2010.

[87] Q.H. Tran, T.J. Chin, G. Carneiro, M. Brown, and D. Suter. In defence of RANSAC for outlier rejection in deformable registration. *Computer Vision–ECCV 2012*, pages 274–287, 2012.

[88] H. Uchiyama and E. Marchand. Deformable random dot markers. In *IEEE International Symposium on Mixed and Augmented Reality*, pages 237 –238, October 2011.

[89] H. Uchiyama and H. Saito. Augmenting text document by on-line learning of local arrangement of keypoints. In *IEEE International Symposium on Mixed and Augmented Reality*, pages 95–98, 2009.

[90] H. Uchiyama and H. Saito. Random dot markers. In *IEEE Virtual Reality Conference*, pages 35 –38, March 2011.

[91] J. Vince and R. Earnshaw. *Digital Media: The Future*. Springer, 2000.

[92] E. Vincent and R. Laganiere. Detecting planar homographies in an image pair. In *International Symposium on Image and Signal Processing and Analysis*, pages 182–187, 2001.

[93] M. Weiser. The computer for the 21st century. *Scientific American*, 265(3):94–104, 1991.

[94] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *IEEE International Conference on Computer Vision*, pages 666–673, 1999.

[95] J. Zhu, M.R. Lyu, and T.S. Huang. A fast 2D shape recovery approach by fusing features and appearance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(7):1210–1224, July 2009.

[96] K. Zhu, O. Fernando, A. Cheok, M. Fiala, and T.W. Yang. Origami recognition system using natural feature tracking. In *IEEE International Symposium on Mixed and Augmented Reality*, pages 289–290, October 2010.

[97] A. Ziegler and S. Belongie. Non-rigid surface detection for gestural interaction with applicable surfaces. In *Proc. WACV*, pages 73 –80, January 2012.

[98] S. Zollmann, T. Langlotz, O. Bimber, and J. Herder. Passive-Active Geometric Calibration for View-Dependent Projections onto Arbitrary Surfaces. *Virtual Reality*, 4(6), 2007.

[99] M. Zuliani, CS Kenney, and BS Manjunath. The multiransac algorithm and its application to detect planar homographies. In *International Conference on Image Processing*, pages 153–156, 2005.

Publications

Journal Articles (in English)

1. Sandy Martedi, Maki Sugimoto, Hideo Saito, and Bruce Thomas, "Feature-based alignment method for projecting virtual content on a movable paper map," *IEEJ Transaction on Electronics, Information and Systems*, 133(3), March 2013.

2. Sandy Martedi, Hideaki Uchiyama, Guillermo Enriquez, Hideo Saito, Tsutomu Miyashita and Takenori Hara, "Foldable augmented maps," *IEICE Transaction on Information and Systems*,95(1):255-256, January 2012.

International Conferences

1. Takayuki Nakamura, Francois de Sorbier, Sandy Martedi, and Hideo Saito, "Calibration-Free Projector-Camera System for Spatial Augmented Reality on Planar Surfaces," In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR)*, page 85-88, November 2012.

2. Sandy Martedi, Pega Sanoamuang, Milica Muminovic, Sebastien Callier, Hideo Saito, "Visualization of urban prediction using augmented maps," In *Proceedings of the 11th Biennial Conference on Engineering Systems Design and Analysis (ESDA)*, July 2-4, 2012.

3. Sandy Martedi, Maki Sugimoto, Hideo Saito and Bruce Thomas, "Visualizing map layers using spatial augmented reality," In *Proceedings of the 18th Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV)*, page 115-119, February 2-4, 2012.

4. Sandy Martedi and Hideo Saito, "Augmented Fly-through using Shared Geographical Data," In *Proceedings of the 21st International Conference on Artificial Reality and Teleexistence (ICAT2011)*, page 47-52, November 28-30, 2011.

5. Sandy Martedi and Hideo Saito, "Towards Bendable Augmented Maps," In *Proceedings of the 12th IAPR Conference on Machine Vision Applications (MVA2011)*, page 566-569, June 13-15, 2011.

6. Sandy Martedi and Hideo Saito, ”Foldable Augmented Papers with a Relaxed Constraint,” In *Proceedings of the 1st International Symposium on Access Spaces (IEEE-ISAS 2011)*, page 127-131, June 17-19, 2011.

7. Sandy Martedi, Hideaki Uchiyama, Guillermo Enriquez, Hideo Saito, Tsutomu Miyashita and Takenori Hara, ”Foldable augmented maps,” In *Proceedings of the 9th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, page 65-72, October 2010.

8. Guillermo Ignacio Enriquez, Hideaki Uchiyama, Sandy Martedi, Hideo Saito, Tsutomu Miyashita and Takenori Hara, ”Interactive paper maps: Merging AR visualization using keypoint tracking and hand gesture based interaction,” In *Proceedings of the 3rd Korea-Japan Workshop on Mixed Reality (KJMR2010)*, April 2010.

9. Sandy Martedi, Hideaki Uchiyama and Hideo Saito, ”Clickable augmented documents,” In *Proceedings of the 12th IEEE International Workshop on Multimedia Signal Processing (MMSP)*, page 162-166, October 2010.

10. Sandy Martedi, Hideo Saito, and Myriam Servières, ”Shape measurement system of foot sole surface from flatbed scanner image,” In *Proceedings of the IAPR Conference on Machine Vision Applications (MVA2009)*, page 338-341, May 20-22, 2009.

11. Sandy Martedi, Hideo Saito, and Myriam Servières,”3D Shape Reconstruction of Sole Surface of Human Foot using Flatbed Scanner,” In *Proceedings of the 15th Japan-Korea Joint Workshop on Frontiers of Computer Vision (FCV2009)*, page 118-123, February, 2009.