NAIST-IS-DD1061008

Doctoral Dissertation

Camera Pose Estimation for an Image Sequence with External References

Hideyuki Kume

March 13, 2014

Department of Information Systems
Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

Hideyuki Kume

Thesis Committee:

Professor Naokazu Yokoya	(Supervisor)
Professor Hirokazu Kato	(Co-supervisor)
Associate Professor Tomokazu Sato	(Co-supervisor)

Camera Pose Estimation for an Image Sequence with External References*

Hideyuki Kume

Abstract

Structure-from-Motion (SfM) is one of the key techniques developed in the field of computer vision, and has been used in many applications such as three-dimensional reconstruction, robot navigation, and augmented reality. The most significant problem in SfM is the accumulation of estimation errors in a long image sequence. Although many types of methods for reducing accumulative errors have been proposed, SfM essentially cannot be free from accumulative errors unless certain external references (e.g., GPS, aerial images, or feature landmarks) are given.

To treat various scenes for which the appropriate external references are not unique, this thesis proposes camera pose estimation methods employing the following three types of references: (1) GPS, (2) aerial images, and (3) a 3D point database created by SfM. GPS and aerial images are already available for most outdoor scenes around the world. To reduce the accumulative errors in SfM, we propose bundle adjustment (BA)-based methods that can globally optimize the camera poses using GPS and aerial images. Some applications of robot navigation and augmented reality require estimating the camera poses along a previously taken route. For such applications, we propose an online camera pose estimation method using a 3D point database created by SfM from previously captured images.

In methods using GPS, extended BA that fuses SfM and GPS data has been previously proposed, and it works properly if the GPS data are acquired accurately. However, because existing methods do not consider the confidence of GPS

*Doctoral Dissertation, Department of Information Systems, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD1061008, March 13, 2014.

positioning, the accuracy of estimated camera positions depends largely on the confidence of the GPS positioning data. To solve this problem, we add weighting coefficients depending on the GPS positioning confidence to the energy function for extended BA.

No existing method uses aerial images as external references in BA. We propose a new SfM pipeline that uses feature matches between ground-view and aerial images. To find proper matches from unreliable matches, we newly propose RANSAC-based outlier elimination methods for both the feature matching and BA stages.

As a method using a 3D point database created by SfM, we estimate the camera poses online from the 2D positions of the feature points in the current image and their 3D positions obtained from the database. The challenge here is how to accurately obtain the 3D-2D correspondences. To this end, the proposed method identifies the database image that is most similar to the current image by considering both topological information and image features.

The usefulness of the proposed methods was quantitatively confirmed through experiments using data obtained in real environments.

Keywords:

camera pose estimation, structure-from-motion, bundle adjustment, GPS, aerial image, 3D point database

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Camera pose estimation for an image sequence has been widely investigated and used for many computer vision and virtual reality applications, including the following.

- Augmented reality [1, 2]
- Robot navigation [3]
- Match move [4]
- 3D reconstruction [5–7]
- Free-viewpoint image generation [8, 9]
- Super-resolution [10]
- Video stabilization [11]

Figure 1.1 shows some example applications. For augmented reality and match move, which are applied in navigation systems, educational materials, and film making, camera pose estimation is required for superimposing virtual objects into geometrically correct positions. To virtualize a real environment, 3D reconstruction and free-viewpoint image generation techniques have been thoroughly investigated, and are based on the camera pose information used to integrate many input images. One of the requirements of robot navigation is the ability to determine the robot's location, which can be realized through camera pose

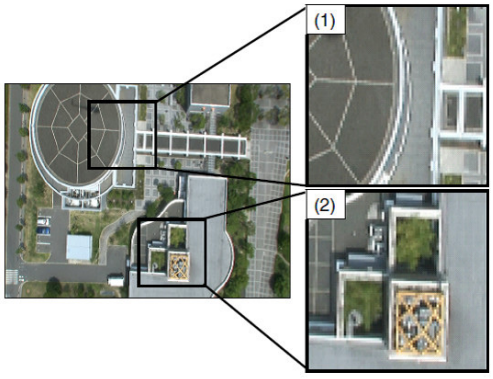
(a) Augmented reality [1]

(b) 3D reconstruction [7]

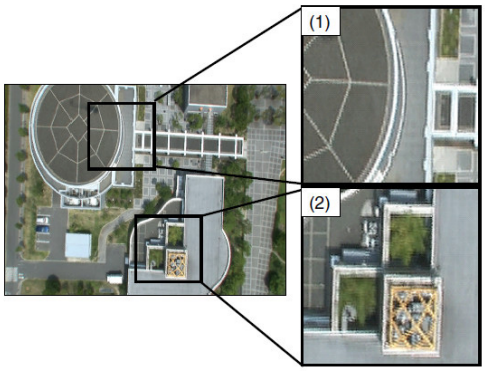
Free-viewpoint image

Novel camera position (red)

(c) Free-viewpoint image generation [9]



Input image



Super-resolved image

(d) Super-resolution [10]

Figure 1.1: Applications of camera pose estimation.

estimation. In this manner, camera pose estimation is one of the fundamental tasks in computer vision, virtual reality, and robotics applications.

For estimating camera poses, Structure-from-Motion (SfM) methods, which estimate camera poses and 3D positions of feature points from an image sequence, are frequently used. The most significant problem of SfM is the accumulation of estimation errors in a long image sequence. Although many types of methods for reducing accumulative errors have been proposed, SfM methods essentially cannot be free from accumulative errors unless certain external references (e.g., GPS, aerial images, or feature landmarks) are given. In addition, SfM without external references can only be used to estimate relative camera poses and not absolute camera poses, which are required for certain applications such as robot navigation.

To treat various scenes for which the appropriate external references are not unique, this thesis proposes camera pose estimation methods employing the following three types of references: (1) GPS, (2) aerial images, and (3) a 3D point database created by SfM. GPS and aerial images are already available for most outdoor scenes around the world. To reduce accumulative errors in SfM, we propose bundle adjustment (BA)-based methods that can globally optimize camera poses using GPS and aerial images. Some applications of robot navigation and augmented reality require estimating the camera poses along a previously taken route. For these applications, we propose an online camera pose estimation method using a 3D point database created by SfM from previously captured images.

Various methods for estimating camera poses have been proposed. In this chapter, we first review existing camera pose estimation methods with and without external references. We then describe the contributions of this thesis against these existing methods. Finally, we describe the remaining organization of this thesis.

1.1. Camera Pose Estimation without External References

This section describes SfM methods that estimate camera poses and 3D positions of feature points from the correspondences of feature points among input images. We first review the SfM methods in terms of their estimation approaches. Techniques used to reduce accumulative errors in SfM are then detailed.

1.1.1 Structure-from-Motion

Many SfM methods have been proposed in the fields of computer vision and robotics [12–14]. These methods can be classified as follows based on their estimation approaches.

- Epipolar geometry
- Factorization
- Filtering
- Local bundle adjustment

In the following, we describe the characteristics of these methods.

Epipolar geometry

The epipolar geometry represents the geometric relationship among two, three, and four cameras [12]. Because the epipolar geometry for three and four cameras can be considered as combination of the epipolar geometry for two cameras, the epipolar geometry for two cameras has been thoroughly studied.

The relationship of the 2D positions of the corresponding points between two cameras is represented using a fundamental matrix. If there are eight or more correspondences, the fundamental matrix can be computed using the eight-point algorithm [15]. Hartley [16] proposed the normalized eight-point algorithm to achieve a robust estimation against image noises. Because the rank of the fundamental matrix is two, a fundamental matrix can be computed through seven correspondences using a rank constraint [17].

If the intrinsic camera parameters are known, a fundamental matrix can be converted into an essential matrix, from which the relative camera pose can be computed. In addition, methods that directly compute an essential matrix from five correspondences have been proposed [18–21]. If the intrinsic camera parameters are unknown, self-calibration methods that simultaneously estimate the relative pose and intrinsic parameters are required. A simple method [12] calculates the focal length from a fundamental matrix. Moreover, methods that estimate the relative pose and focal length from six correspondences have also been proposed [20, 22].

These epipolar-geometry-based techniques are useful for certain applications such as 3D reconstruction from two images. More importantly, these methods can be used as initialization procedures in the filter-based and local-BA-based methods described later in this thesis.

Factorization

Factorization [23] is a method for linearly estimating the camera poses and 3D positions of feature points from the 2D correspondences of the feature points among the input images. Factorization was originally developed for orthographic cameras [23], and was later extended to paraperspective cameras [24], projective cameras [25] and perspective cameras [26].

One disadvantage of factorization methods is that they require a complete set of 2D correspondences, i.e., each point must be visible in each frame. Although methods that can deal with missing components of correspondences have been proposed [27, 28], they require iterations to solve this problem. Moreover, it is difficult for factorization methods to handle incorrect correspondences, which often occur for long video sequences.

Filtering

To estimate the camera poses and 3D positions of feature points sequentially for an image sequence, stochastic filters are sometimes used. Davison et al. [3] proposed a method based on the extended Kalman filter (EKF) that can deal with the non-linearity of perspective projections in the observation model. Civera et al. [29] proposed RANSAC for EKF-based SfM to achieve robust estimation

against incorrect correspondences of 2D feature points. Eade and Drummond [30] proposed a method based on FastSLAM using the Rao-Blackwellized particle filter, which is more accurate than EKF SLAM in the field of robotics. One problem of filter-based SfM is that it has difficulty handling large-scale environments with many feature points. To solve this problem, Eade and Drummond [31] proposed a method for dividing large-scale environments into smaller sub-spaces.

These filter-based methods tend to be employed in real-time applications such as robot navigation and augmented reality because they can estimate camera poses with low computational cost. However, Strasdat et al. [32] showed that the local-BA-based methods described later in this thesis can achieve a more accurate estimation than filter-based methods.

Local Bundle Adjustment

BA optimizes the camera poses and 3D positions of feature points so as to non-linearly minimize the sum of the reprojection errors. BA was originally used as the final optimization process for SfM methods owing to its high computational costs.

Thanks to recent advances in the computational power of PCs, some methods [1, 33–40] have employed BA in the incremental processing of an image sequence by limiting the range of BA, which is called local BA. These methods first estimate the initial camera poses and 3D positions of the feature points using the epipolar geometry. When each new frame is input, these methods then (1) estimate the camera pose by solving the PnP problem, (2) estimate the 3D positions of the feature points through triangulation, and (3) apply local BA.

As an initial attempt, Nister et al. [33] proposed a real-time method that employs this incremental framework except for applying local BA. Zhang and Shan [34] and Engels et al. [35] employed local BA that uses only a small number of the most recent images. Mouragnon et al. [36] and Klein and Murray [1] employed key-frames that are spatio-temporally distant from each other, instead of the most recent frames. To deal with large-scale environments, Holmes and Murray [37] proposed a method for dividing the target environments into small sub-spaces. In addition, methods that consider environmental changes [38,39] and a rolling shutter camera [40] have been proposed. As mentioned in the previous

(a) (b)

Figure 1.2: SfM from community photos [41].

section, Strasdat et al. [32] showed that local-BA-based methods can achieve a more accurate estimation than filter-based methods.

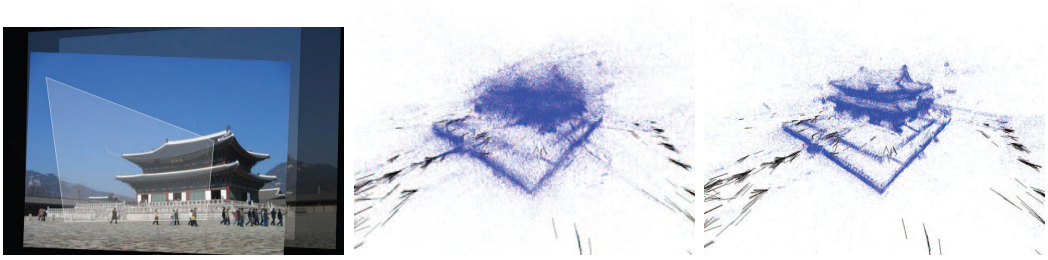
Snavely et al. [41] proposed a local-BA-based method that can deal with unordered images and applied their method to community photos downloaded from the Internet. Figure 1.2 shows example results of SfM from community photos. Because community photos include a significant number of images, the computational cost becomes quite expensive. To solve this problem, as with key-frame approaches, Snavely et al. [42] selected a smaller number of images for the original estimation and folded in the remaining images at the very end. Wu [43] developed VisualSFM, a state-of-the-art implementation of local-BA-based SfM that can handle unordered images as well as image sequences.

1.1.2 Techniques to Reduce Accumulative Errors

Since SfM methods essentially suffer from accumulative errors owing to their incremental manner, techniques to reduce accumulative errors have also been investigated and applied. Such techniques are based on BA and loop closing.

Bundle Adjustment

BA is a procedure for optimizing camera poses and 3D positions of feature points so as to non-linearly minimize the sum of squared reprojection errors. BA is the most accurate way to address the SfM problem because BA can be considered as the maximum likelihood estimation when errors in the 2D positions of the



(a) Image of scene (b) Before BA (c) After BA

Figure 1.3: Example of the effectiveness of BA [45].

correspondences can be assumed to be normally-distributed [44]. Figure 1.3 shows an example of the effectiveness of BA.

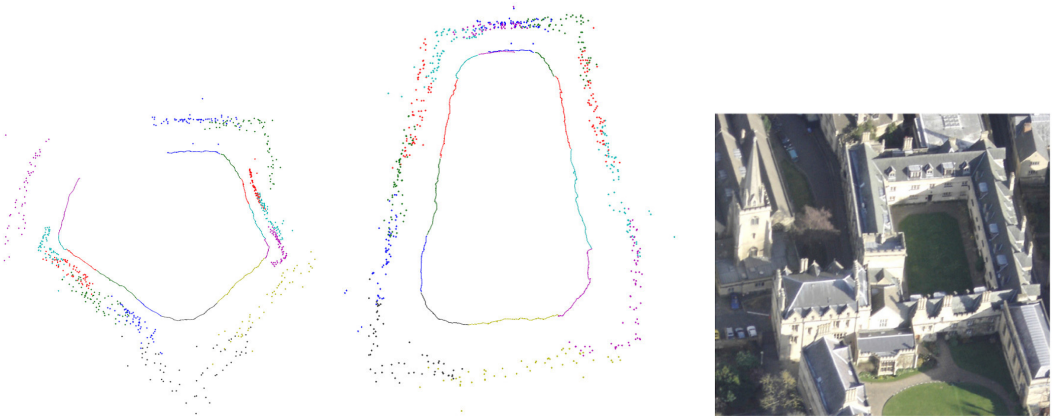
BA requires solving the large non-linear least squares problem and is computationally expensive. To reduce the computational cost, some methods [44–47] exploit the sparsity of BA, i.e., each reprojection error depends only on one camera pose and one 3D position of a point.

On the other hand, Hedborg et al. [40] proposed BA considering a rolling shutter model. To improve the robustness against the outliers of the correspondences, Dai et al. [48] proposed BA using the L1 norm instead of the squared (L2) norm. Implementations of BA have been recently made available [49–52].

Loop Closing

As shown in Figure 1.4, loop closing is a technique for reducing accumulative errors by detecting loops, i.e., camera returns to a position where the camera passes before. Once the loops are detected, the accumulative errors can be reduced by the BA. The problem then is how to detect the loops. For offline SfM methods, it is easy to detect loops by matching the images with each other. In contrast, for online real-time SfM methods, this is relatively difficult owing to the real-time constraint. For efficient loop detection, Angeli et al. [53] employed bag-of-visual-words, which are often used for image retrieval, and Strasdat et al. [54] employed a key-frame approach. In addition, Williams et al. [55] compared loop closing techniques and summarized their characteristics.

As an approach related to loop closing, Cohen et al. [56] exploited symmetries that often exist in man-made structures, e.g., two identical wings of a large



(a) Before loop closing (b) After loop closing (c) Aerial view of courtyard

Figure 1.4: Loop closing for the trajectory around a courtyard [55].

building complex. They employed BA by considering the detected symmetries to reduce accumulative errors (Figure 1.5).

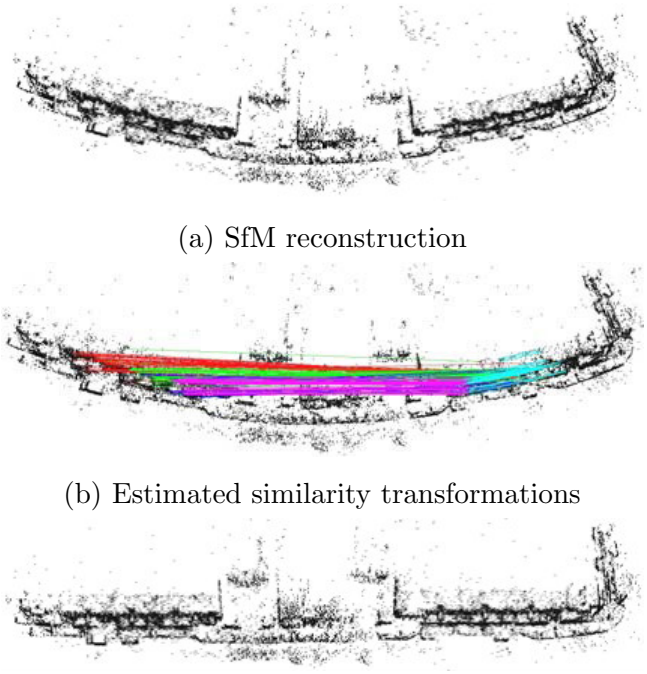
Although these loop closing techniques can reduce accumulative errors, the applicable environments and movements of a camera are limited.

1.2. Camera Pose Estimation with External References

As described in the previous section, while many SfM methods have been proposed, they suffer from accumulation of estimation errors in a long image sequence. Although loop closing techniques can reduce accumulative errors, the applicable environments and camera movements are limited, and SfM methods cannot be free from accumulative errors unless certain external references are provided.

This section reviews camera pose estimation methods using external references. These external references can be classified into the following categories.

- Sensors
- Pre-knowledge of the target environments



(a) SfM reconstruction

(b) Estimated similarity transformations

(c) Result

Figure 1.5: BA exploiting symmetry [56].

- Fiducial markers
- Image databases
- 3D models
- 3D point databases
- Aerial images
- Road maps

In the following, we describe the characteristics and problems of methods using external references.

1.2.1 Sensors

Sensors used for estimating camera poses can be classified into two types: (1) sensors that measure relative poses, such as odometry, and (2) sensors that measure absolute poses, such as RFID and GPS.

Measuring Relative Poses

Sensors that measure relative poses, such as odometry, can estimate accurate poses in a short period of time. Eudes et al. [57] and Michot et al. [58] proposed methods fusing odometry estimates into SfM to reduce the accumulation of errors in the scale parameters. Michot et al. [58] also proposed a method using a gyroscope to reduce the accumulation of errors in the posture parameters. However, sensors that measure relative poses also suffer from accumulative errors over long periods of time. Thus, it is difficult for these sensors to remove accumulative errors in a long image sequence.

Measuring Absolute Poses

To measure absolute poses, GPS and other kinds of infrastructure sensors such as RFID and infrared beacon can be used.

Some methods estimate camera poses directly from sensors. Tenmoku et al. [59] used RFID and infrared beacons (Figure 1.6(a)) to measure absolute poses in an indoor environment. They also used a pedometer to estimate the relative poses in cases when the RFID and infrared beacons cannot be used. Piekarski et al. [60] used a backpack with GPS and an Inertial Measurement Unit (IMU) (Figure 1.6(b)) to obtain camera poses in an outdoor environment. Kourogi et al. [61] proposed a method using RFID, GPS, and odometry to handle both indoor and outdoor environments. Methods for estimating camera poses directly from sensors are robust against rapid camera movements. However, for vision applications such as 3D reconstruction and augmented reality, it is difficult for these methods to achieve pixel-level registration owing to calibration errors between sensors and the camera.

Some methods fuse sensors into SfM to reduce the accumulative errors. Ramachandran et al. [62] proposed an optimization procedure based on reprojection errors and the direction of gravity measured by inertial sensors. Although this method produces better solutions than ordinary BA, it is difficult to reduce the accumulation of errors in the position and scale parameters.

In outdoor environments, some methods use GPS together with SfM owing to its availability. These methods can be classified in terms of their fusion type as follows.

Infrared beacons RFID tags and a tag reader

(a) Positioning infrastructures [59]

(b) Backpack including GPS and IMU [60]

Figure 1.6: Sensor examples.

- Epipolar-geometry-based method [63]
- Fitting-based methods [64–66]
- Filter-based methods [67–69]
- BA-based methods [70–73]

Carceroni et al. [63] proposed a method for estimating the essential matrices from the correspondences of the 2D points and given camera positions using GPS. Like the SfM methods without references described in the previous section, this method is useful for the initialization of BA-based methods.

Fitting-based methods [64–66] fuse GPS data into SfM by fitting the camera positions estimated by SfM to those by GPS using a similarity transform. Although Bok et al. [65] and Wei et al. [66] employed local fitting, the accumulative

errors cannot be removed through a similarity transform because errors are not uniformly accumulated in SfM.

Filter-based methods [67–69] tend to be employed for real-time applications because GPS and vision data can instantly be fused from the previous state and the current measurement, for example, using the Kalman filter. One problem of filter-based methods is the difficulty of global optimization owing to the sequential updating strategy of the filter design.

Some methods [70–73] employ extended BA that minimizes the energy function defined as the sum of reprojection errors and the penalty term of the GPS. The extended BA can globally optimize the camera poses, which works fine if GPS data are accurately acquired. However, the accuracy of estimated camera positions depends largely on the confidence of GPS positioning data because existing extended-BA methods do not consider the GPS positioning confidence, and the error of GPS positioning easily grows to a level of 10 [m] in urban areas.

1.2.2 Fiducial Markers

Fiducial markers are artificial 2D objects whose appearances are designed to be easily determined through image processing techniques. Once fiducial markers are determined, camera poses in the marker coordinate system can be estimated by solving the PnP problem. Kato and Billinghurst [74] proposed one of the most famous camera pose estimation methods based on fiducial markers called ARToolKit (Figure 1.7(a)). This method is simple and useful, especially for augmented reality applications. However, to estimate the camera poses in large-scale environments using fiducial markers, many markers should be installed in the environment, which requires significant manual intervention and results in a disturbance in the scenery. To avoid disturbing the scenery, invisible markers [75] (Figure 1.7(b)) and markers designed to blend in with the scenery as wall paper [76] (Figure 1.7(c)) and posters [77] (Figure 1.7(d)) were proposed. However, the problem of manual intervention still remains.

(a) ARToolKit marker [74]

Capturing without a flash Capturing with a flash

(b) Invisible markers [75]

(c) Wall paper [76]

(d) Posters [77]

Figure 1.7: Examples of fiducial markers.

1.2.3 Image Databases

As shown in Figure 1.8, the image database consists of images and their camera poses estimated beforehand by sensors or SfM. Methods utilizing an image database [78–89] estimate camera poses by identifying the database image that has the most similar appearance to the current image. These methods can be classified into methods for a single image and methods for an image sequence.

For methods using a single image, Wang et al. [78] proposed a method using vocabulary trees [79], which is an efficient technique for image retrieval, to identify the most similar image. After identifying the most similar database image, some methods [78, 80, 81] use epipolar geometry to estimate the relative poses between the current image and the identified database image. Through recent advances in image retrieval techniques and community photos whose camera poses are tagged

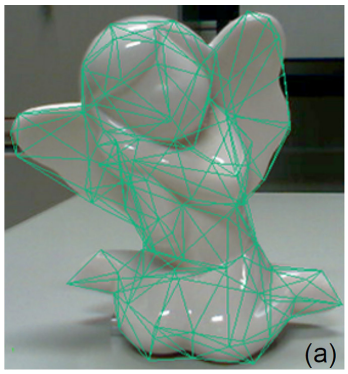
(a) Positions of images (b) Examples of images

Figure 1.8: Example of an image database [88].

by sensors, this approach can handle environments from the city scale [82, 83] up to the entire planet [84]. In addition, for community photos without pose information, Kalantidis et al. [85] proposed a method to automatically determine landmarks using Wikipedia [wikipedia.org].

For methods using an image sequence, the methods using a single image described above can be used by applying the particular method to each image of the image sequence. However, estimated camera poses may change discontinuously between successive images because each image is treated independently. Therefore, the methods for an image sequence consider spatio-temporal information between successive images. Yagi et al. [86] created a route panorama from database images. The camera poses are then continuously estimated by tracking the image pattern on the route panorama that is similar to the current image through the use of the active contour model. Some methods [87–89] use a topological graph to identify the database image that is the most similar to the current image by considering the spatio-temporal connection of the database images. Badino et al. [88] fused metric-scale information into a topological graph. Vaca-Castano et al. [89] used street-view images available on the Internet, such as from Google Street View [`maps.google.com/streetview`], as database images.

These methods utilizing an image database [78–89] require the image database to be prepared beforehand, which is costly. Although community photos and street-view images can be used as a database, the available areas are still limited



Comport et al. [90] Oikawa et al. [94]

(a) Wire-frame models

(b) Textured models [92]

Figure 1.9: Examples of 3D models.

to famous landmarks and large cities. However, preparing a database is acceptable for certain applications in which the camera iteratively passes along the same route. Additionally, these methods are efficient for a camera that follows a previously taken route by considering the spatio-temporal information.

1.2.4 3D Models

Three-dimensional models of scenes such as wire-frame models (Figure 1.9(a)) and textured models (Figure 1.9(b)) are used as external references. Some methods [90–94] estimate camera poses directly from 3D models by matching the features between input images and 3D models. Comport et al. [90] used wire-frame models of scenes. To improve the robustness of the matching, texture information is also used [91–93] along with wire-frame models. On the other hand, Oikawa et al. [94]

addressed the problem of tracking textureless rigid curved objects using quadrics to approximate the object contours.

Some methods fuse 3D models into SfM [95–98]. These methods can reduce the accumulative errors in SfM using the following 3D models.

- Wire-frame models [95]
- Plane-based models [96, 97]
- Textured 3D models [98]
- Digital elevation models (DEM) [97]

Approaches fusing 3D models into SfM can be classified into switching [95, 96] and extended BA [97, 98]. Bleser et al. [95] and Lothe et al. [96] simply switch the SfM and model-based estimation depending on the availability of the models. As with methods using GPS with SfM, some methods [97, 98] have employed extended BA because it can globally optimize the camera poses using sparsely acquired information on the camera poses estimated from the models.

One disadvantage of 3D-model-based methods is that the manual intervention required to create the 3D models is costly. Although some models are already available in the GIS database [96, 97], the available areas are still limited to large cities.

1.2.5 3D Point Databases

As shown in Figure 1.10, some methods [99–108] employ a database that consists of 3D points and their image features and estimate camera poses by matching the feature points between input images and the database. Since 3D points can be estimated from images using SfM methods, less manual intervention is required than in 3D-model-based methods. The main problem of this approach is how to accurately and efficiently estimate the feature matches. Approaches to this problem depend on whether the methods estimate the camera pose for a single image or for an image sequence.

Regarding single-image-based methods, an earlier method [99] simply found feature matches using a brute force search. Irschara et al. [100] clustered 3D points

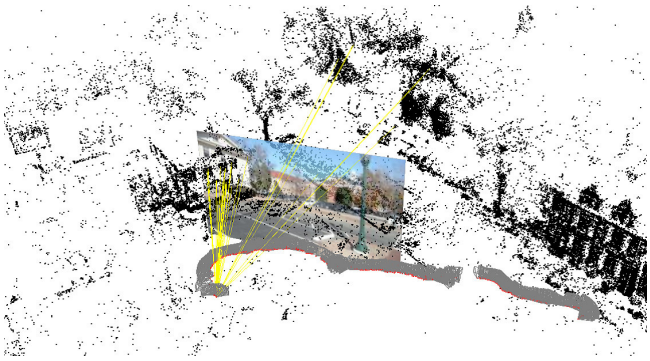


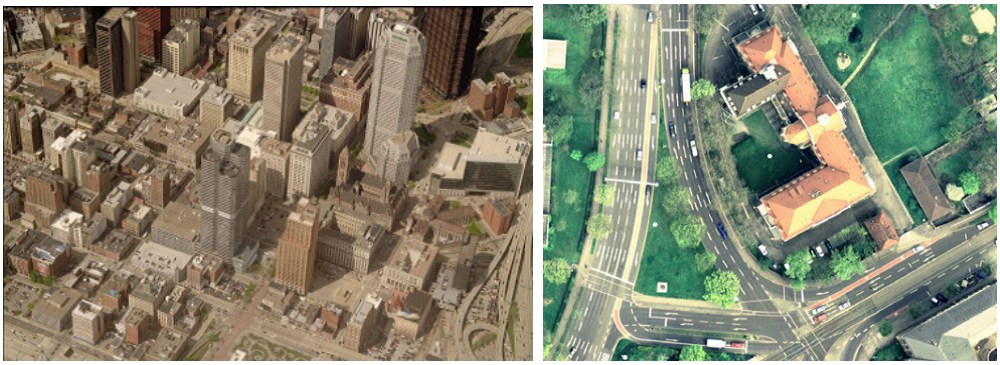
Figure 1.10: 3D point database [100]. Camera poses are estimated by matching the feature points between the input images and 3D point database.

by generating virtual views and projecting 3D points with image features into these views. The search space is then limited using an image retrieval technique that identifies a virtual view including the image features most similar to those of the current image. Li et al. [101] introduced priorities of feature points to accelerate the search process. When a match is found, the priorities increase for the points that are visible with the matched point in a database image. Sattler et al. [102] matched feature points between the input image and the database using a vocabulary tree [79].

In image-sequence-based methods, temporal information is helpful to limit the search space. Arth et al. [103] limited the search space by clustering 3D points using the visibility of the feature points. Wientapper et al. [104] estimated a tentative camera pose through the use of the EKF with an inertial sensor. Matches are then searched for only among those feature points that are projected onto the tentative camera's field of view. Taketomi et al. [105] also estimated a tentative camera pose by tracking the feature points temporally. Lim et al. [106] also tracked the feature points temporally and searched only images in which the tracked feature points are detected.

As with the methods using other references, some methods [107, 108] fuse the 3D point database into SfM. These methods employ BA-based optimization because it can globally optimize the camera poses.

These methods utilizing a 3D point database require the database to be prepared in advance, which is costly even if an SfM method can be used. However,



(a) Perspective image [109] (b) Orthogonal image [115]

Figure 1.11: Examples of aerial images.

as with the methods utilizing an image database, preparing the database is admissible for certain applications in which the camera iteratively travels along the same route.

1.2.6 Aerial Images

Aerial images have recently become available for most outdoor scenes around the world, such as Google Maps [maps.google.com] and Microsoft Bing Maps [bing.com/maps]. Focusing on this availability, camera pose estimation methods using aerial images have been proposed. There are two types of aerial images: perspective images (Figure 1.11(a)) and orthogonal images (Figure 1.11(b)).

Bansal et al. [109] proposed a method for estimating camera poses by matching façades in the ground-view input image with perspective aerial images. Although perspective aerial images are available on Google Maps and Microsoft Bing Maps, the available areas are still limited to large cities.

Most methods using aerial images employ orthogonal aerial images that are available for most outdoor scenes and correspond to absolute 2D positions, i.e., latitudes and longitudes. These methods can be classified into learning-based [110], edge-based [111, 112], and feature-point-based [113–115] methods.

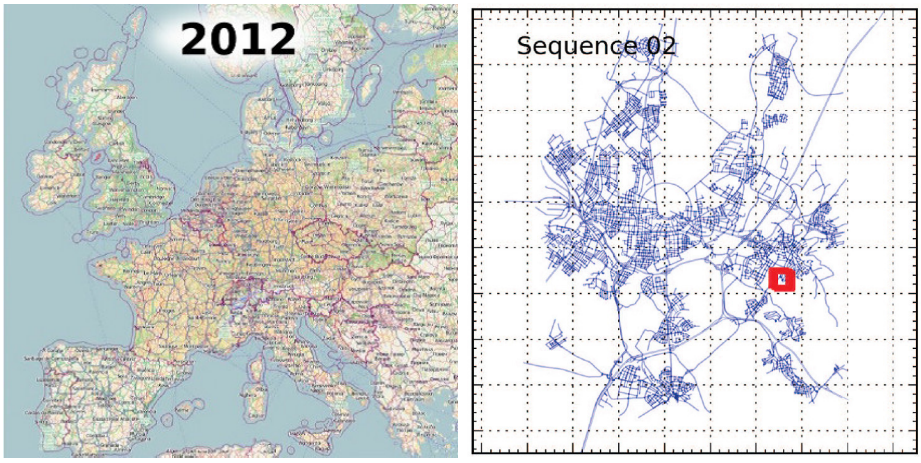
Lin et al. [110] proposed a method using community photos with position information and aerial images. Their method first searches for community photos that are similar and dissimilar to input ground-view images. Then, using aerial

images at positions in the identified community photos, a support vector machine (SVM) is trained and applied to sliding windows over aerial images to estimate the camera position of the input ground-view image. Although this method estimates the camera positions from large regions (1,600 [km^2] in their experiments), only rough positions of the cameras can be estimated.

Kim et al. [111] and Leung et al. [112] proposed methods for estimating camera poses by matching building edges in the ground-view images to those in the aerial images. As a related approach, Cham et al. [116] estimated camera poses for an omnidirectional image using a 2D building map instead of aerial images.

Other methods [113–115] match the feature points between input ground-view images and aerial images. However, it is not easy to find good matches for all images in a long video sequence, especially for scenes where unique landmarks cannot be observed. Mills [117] and Toriya et al. [113] proposed robust feature point matching procedures that compare the orientation and scale of the matches, which are from feature descriptors such as SIFT [118] and SURF [119], with the dominant orientation and scale identified through a histogram analysis. However, this does not work well when a very large number of outliers exist. Toriya et al. [113] and Noda et al. [114] relaxed the problem by generating mosaic images of the ground from multiple images for feature matching. However, the accumulative errors in a mosaic image are not considered in these methods. To resolve this problem, Pink et al. [115] fuse sparsely obtained camera poses estimated from aerial images into SfM using the Kalman filter. However, it is difficult for the Kalman filter to globally optimize the camera poses.

Methods based on matches of the feature points first estimate the homography matrices, and the camera poses are then extracted from the homography. It should be noted that a homography can represent the relationship between ground-view and aerial images only when the ground is flat. To overcome this limitation, Sekii et al. [120] treated feature points on aerial images as 3D points whose horizontal 2D positions are known and altitudes are unknown. Camera poses are then estimated from these 3D points and their 2D positions on the ground-view image.



(a) OpenStreetMap (OSM) (b) Road map extracted from OSM

Figure 1.12: Example road maps [121].

1.2.7 Road maps

In addition to aerial images, road maps are also available for most outdoor scenes around the world. Brubaker et al. [121] proposed a method that employs SfM with OpenStreetMap [openstreetmap.org], which is a community-developed road map (Figure 1.12). Although this method can reduce accumulative errors by matching the trajectory from SfM to the road maps, there are ambiguities for certain types of scenes such as straight roads or Manhattan worlds.

1.3. Contributions of this Thesis

In the previous sections, existing camera pose estimation methods were reviewed; in addition, we indicated that external references should be used to remove accumulative errors. Many kinds of external references have been proposed, and they should be selected depending on the particular situation or application. Figure 1.13 summarizes the characteristics of external references in terms of the amount of manual intervention and the extent of the applicable environments. In this thesis, we focus on the following two situations.

- Situations in which it is necessary to estimate the camera poses without a pre-measurement of the target environments. This type of situation applies

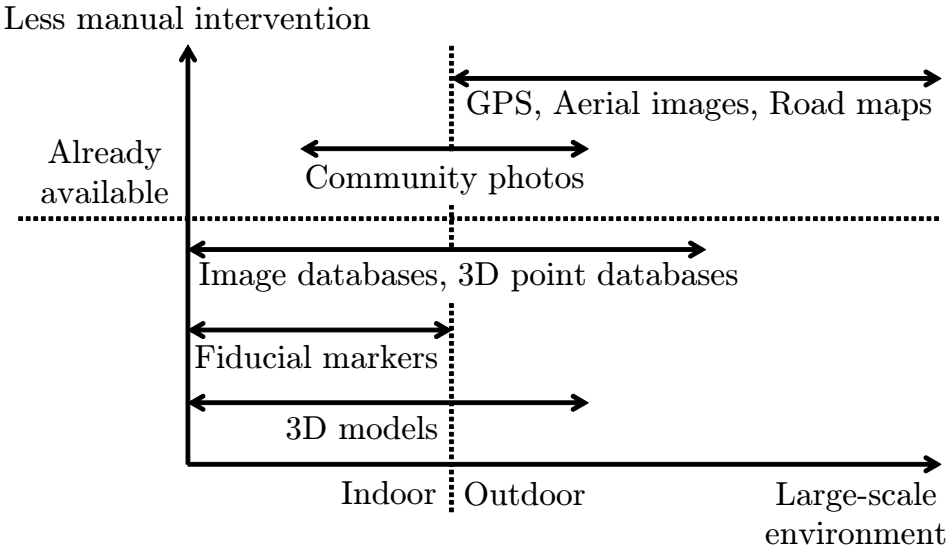


Figure 1.13: Characteristics of external references in terms of the amount of manual intervention and the extent of the applicable environments.

to a 3D reconstruction, match move, free-viewpoint image generation, and so on.

- Situations in which it is necessary to estimate the camera poses along a previously taken route. This type of situation applies to robot navigation, augmented reality, and so on.

In the following, we detail the contributions of the present study for each situation by selecting the appropriate external references.

1.3.1 Camera Pose Estimation without a Pre-Measurement of the Target Environments

As shown in Figure 1.13, the following external references are already available without a pre-measurement of the target environments by users.

- GPS

- Pre-knowledge of a target environment created from community photos and street-view images such as image databases, 3D models, or 3D point databases
- Aerial images
- Road maps

Although community photos and street-view images have been increasing in number, the available areas are still limited to famous landmarks and large cities. In contrast, GPS, aerial images, and road maps are available for most outdoor scenes around the world. Road-map-based methods suffer from ambiguities in certain types of scenes such as straight roads or Manhattan worlds. Therefore, we focus on GPS and aerial images as external references.

It is not easy to estimate camera poses for all images in a long video sequence directly from GPS and aerial images owing to the sampling rate and matching difficulty. We thus propose techniques that fuse SfM with GPS and aerial images using BA because BA-based methods can globally optimize camera poses using sparsely acquired information on camera poses from external references.

In methods using GPS, extended BA that fuses SfM and GPS data has been previously proposed [70–73] and it works fine if the GPS data are acquired accurately. However, the accuracy of the estimated camera position depends largely on the confidence of the GPS positioning data because such methods do not consider the GPS positioning confidence. To solve this problem, we add weighting coefficients depending on the GPS positioning confidence to the energy function for extended BA.

No existing method uses aerial images as external references in the BA. We propose a new SfM pipeline that uses feature matches between ground-view and aerial images. To find good matches from unreliable matches, we newly propose RANSAC-based [122] outlier elimination methods in both feature matching and BA stages.

1.3.2 Camera Pose Estimation along a Previously Taken Route

Some robot navigation and augmented reality applications require estimating the camera poses along a previously taken route. For these applications, most of the references described in the previous section can be used because they can be prepared beforehand. We focus on methods using a 3D point database and an image database because, as shown in Figure 1.13, such databases can be created for large-scale environments with relatively less human effort using SfM methods.

As mentioned in Section 1.2.5, the main problem of methods using a 3D point database is how to obtain matches between the input images and the database. To limit the search space, existing methods track the feature points temporally. However, tracking sometimes fails owing to occlusions and a rapid camera movement. On the other hand, without tracking the feature points, methods using an image database can efficiently identify the database image that is most similar to the current image by considering a spatio-temporal connection. We thus propose a method using a 3D point database that employs an image-database method to limit the search space.

1.4. Organization of this Thesis

The rest of this thesis is organized as follows. Chapter 2 describes a camera pose estimation method that fuses SfM and GPS data using extended BA while considering the confidence of GPS positioning. Chapter 3 describes a new SfM pipeline that uses feature matches between ground-view and aerial images, with a robust feature matching procedure employing a two-stage RANSAC. Chapter 4 describes a method for estimating camera poses online from 2D positions of the feature points in the current image and their 3D positions obtained from the database by considering both the topological information and the image features. Finally, Chapter 5 summarizes this thesis.

Chapter 2

Extended Bundle Adjustment using GPS Positioning and Its Confidence

2.1. Introduction

This chapter describes a camera pose estimation method using GPS positioning data as external references that are already available in most outdoor scenes around the world. As described in Chapter 1, the most significant problem in SfM is the accumulation of estimation errors in a long image sequence. Although many kinds of methods that reduce accumulative errors have been proposed, SfM methods essentially cannot be free from accumulative errors unless certain external references are given.

In this study, to fuse GPS data into SfM, we employ the framework of extended BA that can globally optimize camera poses using sparsely acquired camera positions from GPS. Although extended-BA methods using GPS data have been previously proposed [70–73] and work fine if GPS data are accurately acquired, existing methods have the following problems.

- The accuracy of the estimated camera position depends largely on the confidence of the GPS positioning data because such methods do not consider the GPS positioning confidence.

- The solution often converges to a local minimum when GPS data cannot be acquired for a long period of time.

To resolve these problems, we add weighting coefficients depending on the GPS positioning confidence to the energy function for extended BA. To avoid the local minima, camera positions estimated without GPS data are fitted to the GPS positions prior to the optimization.

The proposed method basically follows the framework of existing extended-BA method [70]. As shown in Figure 2.1, the camera pose estimation (A) and 3D position estimation of the feature points (B) are repeated sequentially for each frame, from the first frame to the last [5]. In this repetition, the local optimization process (D) is applied for each frame in which GPS data are obtained to reduce the accumulative errors. Here, to avoid the local minima, parameter fitting to the GPS positions (C) is applied for frames in which the GPS positioning is recovered after a GPS outage. After estimating the initial camera poses using processes (A) through (D), the estimated poses are globally refined (E). In processes (D) and (E), a common energy function is minimized. In the following, the energy function is first defined using reprojection errors and the penalty terms for the GPS positioning. The optimization process with parameter fitting (C) is then detailed.

2.2. Energy Function Considering GPS Positioning Confidence

To fuse GPS data into SfM, as with existing extended-BA methods [70–73], the energy function is minimized. The energy function $\mathrm{E}_{\mathrm{gps}}$ is defined using the sum of reprojection errors Φ and the penalty term of GPS Ψ as follows:

$$\mathrm{E}_{\mathrm{gps}}(\{\boldsymbol{R}_i, \boldsymbol{t}_i\}_{i=1}^{I}, \{\boldsymbol{p}_j\}_{j=1}^{J}) = \Phi(\{\boldsymbol{R}_i, \boldsymbol{t}_i\}_{i=1}^{I}, \{\boldsymbol{p}_j\}_{j=1}^{J}) + \omega_{\Psi}\Psi(\{\boldsymbol{R}_i, \boldsymbol{t}_i\}_{i=1}^{I}), \quad (2.1)$$

where $\boldsymbol{R}_i$ and $\boldsymbol{t}_i$ represent the rotation and translation from the world coordinate system to the camera coordinate system for the i-th frame, respectively; $\boldsymbol{p}_j$ is the 3D position of the j-th feature point; I and J are the numbers of frames and feature points, respectively, and ω_{Ψ} is a weight that balances Φ and Ψ.

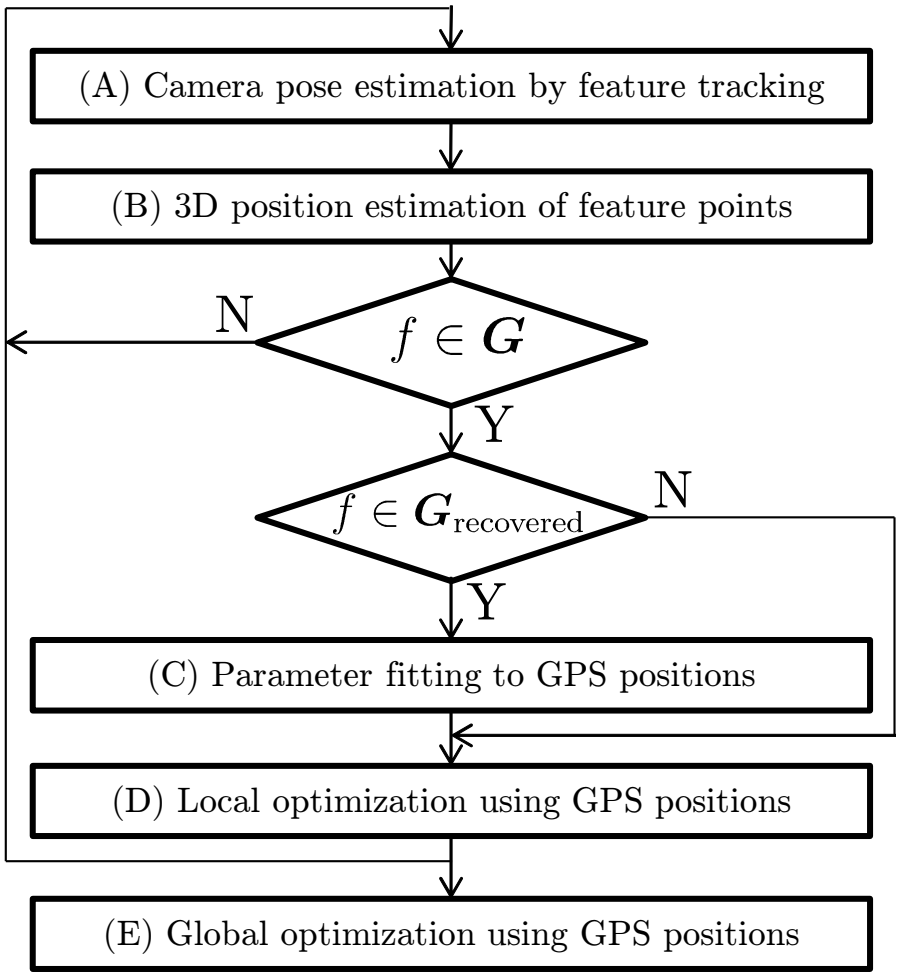


Figure 2.1: Flow diagram of the proposed method using GPS, where f is the frame index, $\boldsymbol{G}$ is a set of frames in which GPS data are obtained, and $\boldsymbol{G}_{\text{recovered}}$ is a set of frames in which GPS positioning is recovered after a GPS outage.

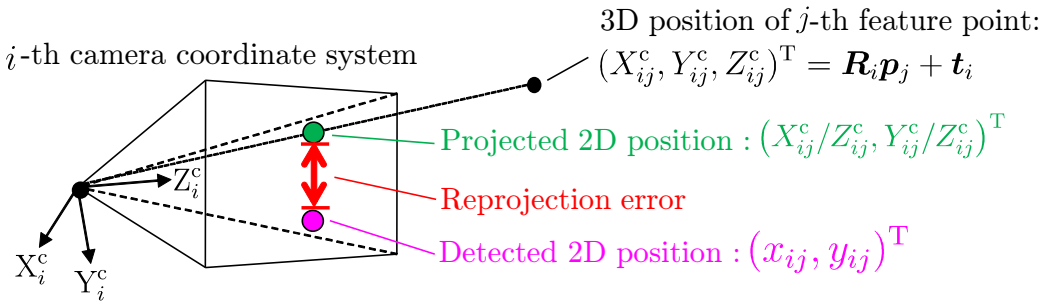


Figure 2.2: Reprojection error.

Here, to treat unreliable GPS data, we newly add weighting coefficients depending on the GPS positioning confidence to the penalty term of GPS Ψ. In the following, the energy associated with the reprojection error Φ and the penalty energy for GPS positioning Ψ are both detailed.

2.2.1 Reprojection Errors

As shown in Figure 2.2, the reprojection error is the distance between the detected 2D position of the feature point and the projected position of the corresponding 3D feature point. The reprojection error has often been used in SfM. In this study, the energy term associated with the reprojection error Φ is defined as follows:

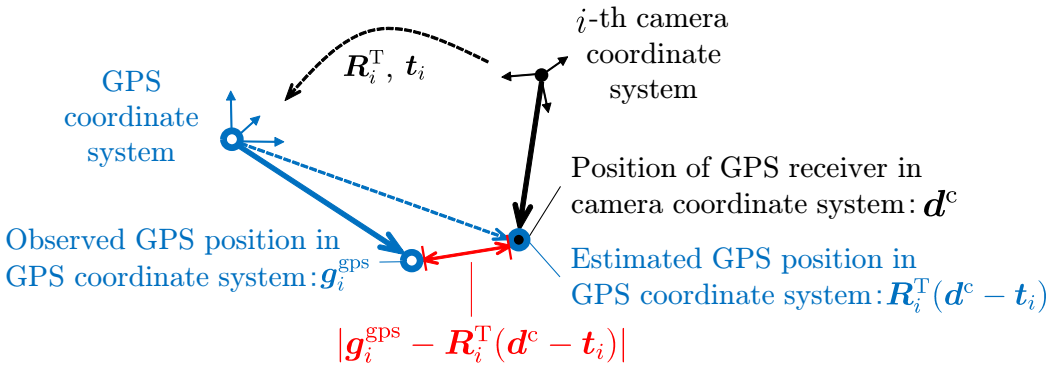
$$\Phi(\{\boldsymbol{R}_i, \boldsymbol{t}_i\}_{i=1}^{I}, \{\boldsymbol{p}_j\}_{j=1}^{J}) = \frac{1}{\sum_{i=1}^{I} |\boldsymbol{P}_i|} \sum_{i=1}^{I} \sum_{j \in \boldsymbol{P}_i} \mu_j \left| \begin{pmatrix} x_{ij} \\ y_{ij} \end{pmatrix} - \begin{pmatrix} \frac{X_{ij}^{\mathrm{c}}}{Z_{ij}^{\mathrm{c}}} \\ \frac{Y_{ij}^{\mathrm{c}}}{Z_{ij}^{\mathrm{c}}} \end{pmatrix} \right|^2, \tag{2.2}$$

$$(X_{ij}^{\mathrm{c}}, Y_{ij}^{\mathrm{c}}, Z_{ij}^{\mathrm{c}})^{\mathrm{T}} = \boldsymbol{R}_i \boldsymbol{p}_j + \boldsymbol{t}_i, \tag{2.3}$$

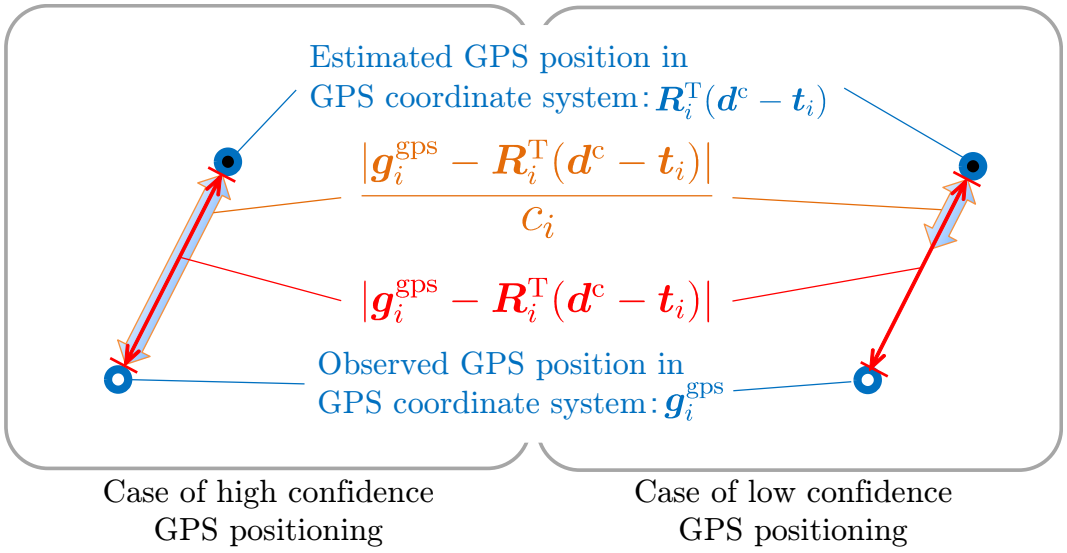
where $\boldsymbol{P}_i$ is a set of the feature points detected in the i-th frame, $(x_{ij}, y_{ij})^{\mathrm{T}}$ is the detected 2D position of the j-th feature point in the i-th frame, and μ_j represents the confidence of the j-th feature point, which is computed from the reprojection errors in the sequential process [5].

2.2.2 Penalty Term for GPS Positioning

As shown in Figure 2.3(a), existing extended-BA methods using GPS [70–73] define the penalty term for GPS positioning through the distance between the



(a) Distance between observed and estimated GPS positions



(b) Weighting coefficient depending on the GPS positioning confidence

Figure 2.3: Energy term with respect to GPS positioning.

observed and estimated GPS positions as follows:

$$\hat{\Psi}(\{\boldsymbol{R}_i, \boldsymbol{t}_i\}_{i=1}^{I}) = \frac{1}{|\boldsymbol{G}|} \sum_{i \in \boldsymbol{G}} |\boldsymbol{g}_i^{\mathrm{gps}} - \boldsymbol{R}_i^{\mathrm{T}}(\boldsymbol{d}^{\mathrm{c}} - \boldsymbol{t}_i)|^2, \tag{2.4}$$

where $\boldsymbol{G}$ is a set of frames in which the GPS positioning data are obtained, $\boldsymbol{g}_i^{\mathrm{gps}}$ is the observed GPS position for the i-th frame in the GPS coordinate system, and $\boldsymbol{d}^{\mathrm{c}}$ indicates the position of the GPS receiver in the camera coordinate system. It should be noted that Lhuillier [72] and Larnaout et al. [73] set $\boldsymbol{d}^{\mathrm{c}} = 0$.

As shown in Figure 2.3(b), we newly define the weighting coefficients depending on the GPS positioning confidence c_i. The penalty term for GPS positioning is defined as follows:

$$\Psi(\{\boldsymbol{R}_i, \boldsymbol{t}_i\}_{i=1}^{I}) = \frac{1}{|\boldsymbol{G}|} \sum_{i \in \boldsymbol{G}} \left(\frac{|\boldsymbol{g}_i^{\mathrm{gps}} - \boldsymbol{R}_i^{\mathrm{T}}(\boldsymbol{d}^{\mathrm{c}} - \boldsymbol{t}_i)|}{c_i} \right)^2. \tag{2.5}$$

Here, c_i should be determined depending on the dilution of precision (DOP), a type of solution in RTK-GPS (RTK-fix, RTK-float), and other error factors. In the experiments described later, c_i is determined by observing the GPS positioning data at a fixed point for a long time period.

2.3. Optimization by Minimizing Energy Function

2.3.1 Range of Optimization

In processes (D) and (E) shown in Figure 2.1, to optimize the camera poses and 3D positions of the feature points, the energy function $\mathrm{E}_{\mathrm{gps}}$ defined in Equation (2.1) is non-linearly minimized. The difference in processes (D) and (E) is the range of optimized frames. In process (D), to reduce accumulative errors in the sequential process, the parameters from the $(f-l)$-th frame to the current frame (f-th frame) are refined. In process (E), the parameters for all frames are refined to globally optimize the camera poses.

2.3.2 Parameter Fitting to GPS Positions

During a long GPS outage, the local optimization process (D) does not work well. Therefore, in the optimization process after GPS positioning is recovered from a GPS outage, the energy often converges to a local minimum because the initial parameters include large accumulative errors (Figure 2.4(a)). Concretely, cameras and feature points around only a frame in which the GPS positioning is recovered are drawn to the recovered GPS position, and most cameras and feature points remain at the original positions (Figure 2.4(b)). In this study, to avoid the local minima, the consistency between the estimated camera poses and the recovered GPS position is improved before the optimization by fitting the camera positions and 3D positions of the feature points to the recovered GPS position. This parameter fitting process is applied except when the confidence of the recovered GPS position is significantly low because confidence of the recovered GPS position is usually low and the next GPS outage may occur before the high-confidence GPS position is obtained. It should be noted that if a high-confidence GPS position is obtained after a GPS outage, the influence of the low-confidence GPS position is suppressed by the optimization when considering the GPS positioning confidence.

For parameter fitting, a similarity transform is not appropriate because reprojection errors become significantly large in frames around the start of the GPS outage. We therefore fit the parameters to gradually decrease the changes from the end of the GPS outage to the start. Concretely, the 3D positions of the feature points and camera poses are corrected through the following steps.

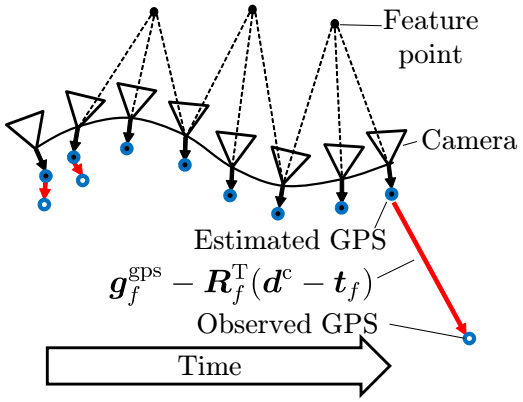
1. As shown in Figure 2.4(c), the 3D position of the j-th feature point observed during the GPS outage is updated as follows:

$$\boldsymbol{p}_j \leftarrow \boldsymbol{p}_j + b_j(\boldsymbol{g}_i^{\mathrm{gps}} - \boldsymbol{R}_i^{\mathrm{T}}(\boldsymbol{d}^{\mathrm{c}} - \boldsymbol{t}_i)), \tag{2.6}$$

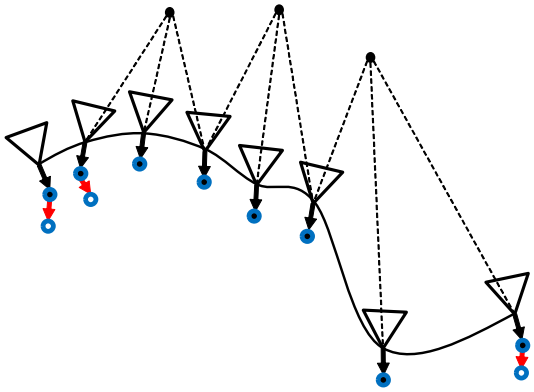
$$b_j = \begin{cases} \frac{m_j - f_s}{f - f_s} & ; m_j - f_s > 0 \\ 0 & ; \text{otherwise} \end{cases}, \tag{2.7}$$

   where f_s is the starting frame of the GPS outage, and m_j represents the median of the frame indexes in which the j-th feature point is observed.

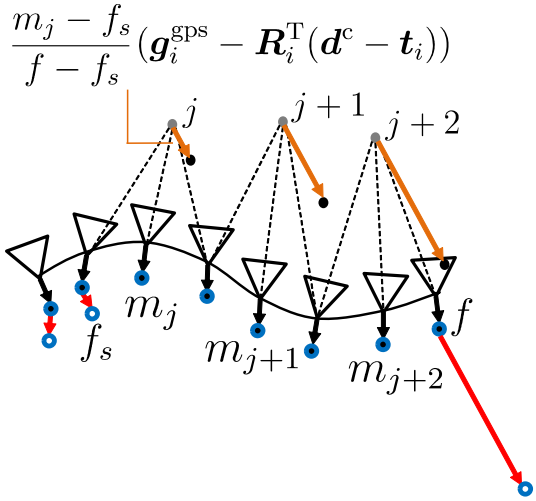
2. As shown in Figure 2.4(d), for the frames during the GPS outage, camera



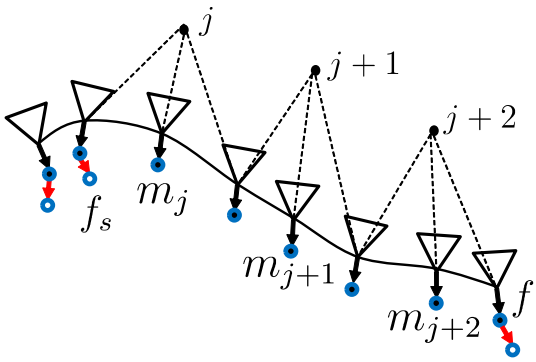
(a) Before optimization



(b) After optimization without parameter fitting



(c) Parameter fitting 1: Translating points



(d) Parameter fitting 2: Estimating camera poses

Figure 2.4: Parameter fitting to GPS positions.

poses are estimated by solving the PnP problem using the 3D positions of the feature points updated by the previous step.

Note that the range of local optimization is set to whole frames during the GPS outage (from the f_s-th frame to the f-th frame) when this fitting process is applied.

2.4. Experiments

To validate the effectiveness of introducing weighting coefficients depending on the GPS positioning confidence and parameter fitting, we conducted two experiments using two datasets: (1) data including many low-confidence GPS positions, and (2) data including a long GPS outage. The accuracy of the proposed method is compared with that of the existing extended-BA method [70] quantitatively using a real video sequence. In the following, we first describe the common aspects of the two experiments and how to determine the weighting coefficients depending on the GPS positioning confidence. The results of each experiment are then detailed.

2.4.1 Experimental Setup

In the experiments, camera poses were estimated for an image sequence (1,600 [pixel] $\times$ 1,200 [pixel], 2,756 frames, and 194 [s]) captured by a moving video camera (Point Grey Research Grasshopper2) mounted on the roof of a vehicle, as shown in Figure 2.5. Figure 2.6 shows examples of the captured images. An RTK-GPS receiver (TOPCON GR-3) was attached to the camera, and the positioning data were acquired at 1 [Hz] during the video capture. Table 2.1 shows the specifications of the RTK-GPS receiver. Approximately 91% of the GPS positioning data were acquired as high-confidence GPS solutions (RTK-fix) and used as the ground truth data. Figure 2.7 shows the positions of the ground truth data. From these data, two datasets including low-confidence GPS solutions (RTK-float) and a GPS outage were generated by masking the GPS satellite data using post-process software (TOPCON Tools), and the generated data were used as the input in the experiments. The details of datasets are described in Sections

Figure 2.5: Camera and RTK-GPS mounted on the roof of a vehicle.

2.4.3 and 2.4.4. It should be noted that the GPS signals used were actually acquired in the experimental environment and included the GPS error sources such as multipaths from buildings and signal decay from trees.

The other conditions were as follows. The intrinsic camera parameters and the position of the GPS receiver in the camera coordinate system were calibrated in advance, and these parameters were fixed during the video capture. The video frames and GPS input were manually synchronized. To fix the accuracy of the feature tracking, we used feature tracks obtained by Sato et al. [5] for all methods

Table 2.1: Specifications of the RTK-GPS receiver.

Model	GR-3
Signal	GPS/GLONASS L1/L2/L5 C/A and P Code & Carrier
Horizontal accuracy (RTK)	10 mm + 1 ppm
Vertical accuracy (RTK)	15 mm + 1 ppm

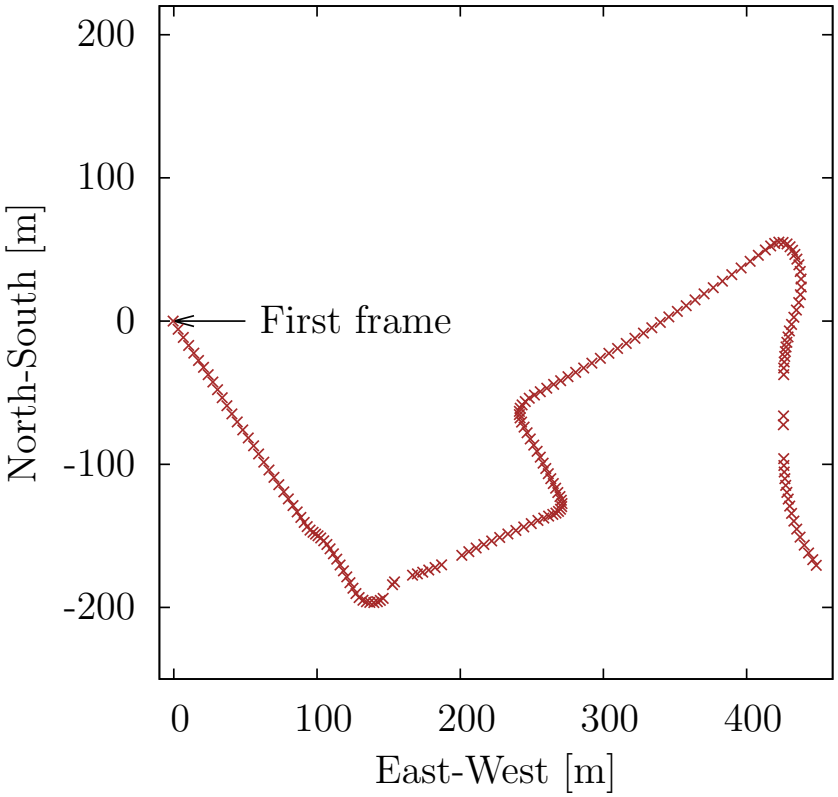
(a) 1st frame (b) 400th frame

(c) 800th frame (d) 1,200th frame

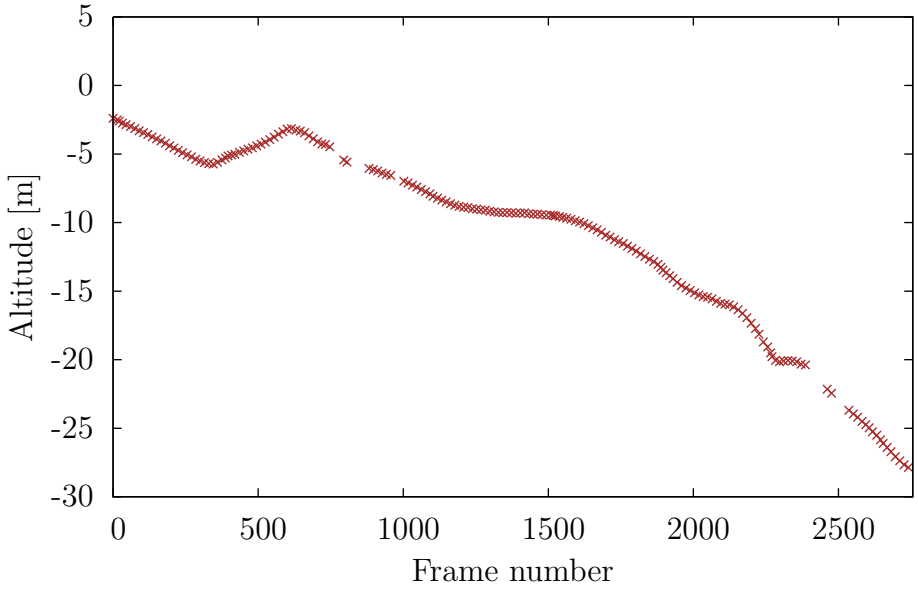
(e) 1,600th frame (f) 2,000th frame

(g) 2,400th frame (h) 2,756th frame

Figure 2.6: Example input images.



(a) Horizontal 2D positions



(b) Altitudes

Figure 2.7: Ground truth GPS positions.

compared. Position errors were calculated by comparing the ground truth GPS positions with the estimated GPS positions $\boldsymbol{R}_i^{\mathrm{T}}(\boldsymbol{d}^{\mathrm{c}} - \boldsymbol{t}_i)$. We set the range of local optimization to $l = 200$, and the weight of the penalty term for GPS positioning to $\omega_\Phi = 10^{-9}$ according to Yokochi et al. [70]. For a non-linear minimization of the energy function, we used sparseLM [50].

2.4.2 Determination of Weighting Coefficients Depending on GPS Positioning Confidence

In our experiments, we used RTK-GPS whose accuracy depends largely on the types of solutions (RTK-fix, RTK-float). Thus, using these solution types, we defined the weighting coefficients depending on the GPS positioning confidence c_i as follows:

$$c_i = \begin{cases} 1 & ; i \in \boldsymbol{G}_{\mathrm{fix}} \\ c_{\mathrm{float}} & ; \mathrm{otherwise} \end{cases}, \tag{2.8}$$

where $\boldsymbol{G}_{\mathrm{fix}}$ is a set of frames in which RTK-fix solution data are obtained. Here, the weight c_{float} for the RTK-float solution data was determined experimentally. Concretely, we first calculated the RMS for both RTK-fix (rms_{fix}) and RTK-float (rms_{float}) by observing the GPS positioning data at a fixed point for a long time period. Then, c_{float} was experimentally determined as $c_{\mathrm{float}} = rms_{\mathrm{float}}/rms_{\mathrm{fix}} = 107.4$.

2.4.3 Quantitative Evaluation using Data Including Many Low-Confidence GPS Positions (Experiment 1)

To validate the effectiveness of introducing weighting coefficients depending on the GPS positioning confidence, as described in Section 2.2.2, the accuracy of the camera positions estimated through the following methods was compared.

Method A: The existing method [70] in which the solution types of GPS data are not considered.

Method B: The existing method [70] using only RTK-fix solution data.

Method C: The proposed method considering the solution types of GPS data.

Method D: An SfM method that does not use GPS data.

Because SfM cannot estimate the absolute camera poses, we fitted the camera positions estimated from Method D to the ground truths using a similarity transform.

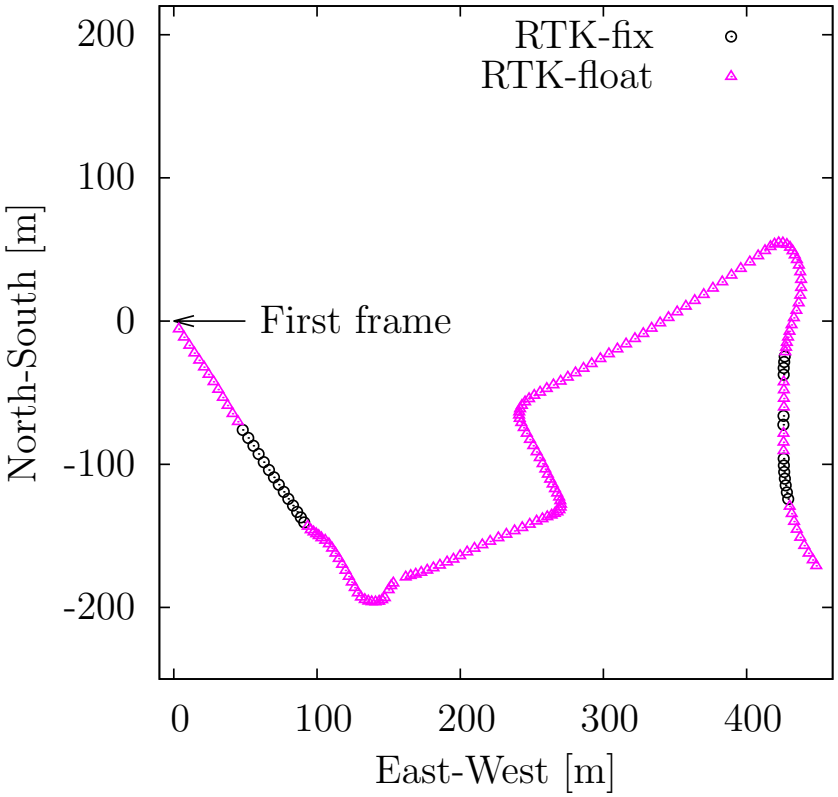
In this experiment, to generate the GPS positioning data through a simulation, we assumed city environments where the confidence of the GPS positioning is occasionally low owing to the occlusion of GPS signals from buildings or trees. As shown in Figure 2.8, the data generated consist of sparsely obtained high-confidence (RTK-fix) GPS positions and many (86% of GPS positions) low-confidence (RTK-float) GPS positions.

Comparison of Position Errors with the Existing Method

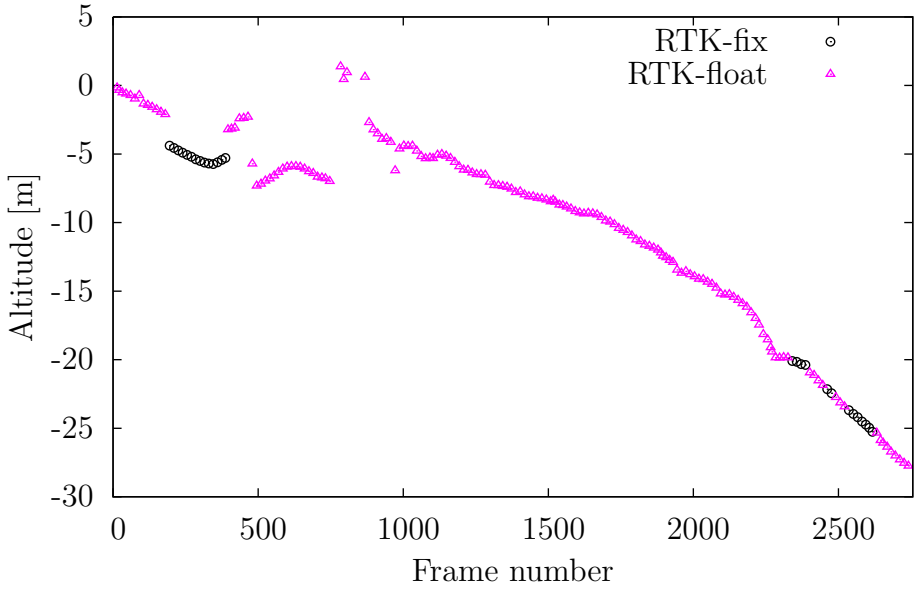
Figures 2.9 and 2.10 show the estimated GPS positions and position errors in each frame, respectively. Table 2.2 shows the statistics of the position errors. These results demonstrate that the vision-based method (Method D) was affected by the accumulative errors even when the estimated positions were fitted to the ground truths through a similarity transform. Methods A, B, and C reduced the accumulative errors using GPS data. However, Method A, which does not consider GPS positioning confidence, was affected by RTK-float solution data. In Method B, which used only RTK-fix solution data, the errors were large while the RTK-float solution data were obtained (500-2,000th frames). The proposed method (Method C) obtained the most accurate positions, as shown in Table 2.2, using the weighting coefficients depending on the GPS positioning confidence.

Discussion about Influence of Parameters

Table 2.3 shows the average position errors from the proposed method with a variable range of local optimization, l, which demonstrates that the optimization within a short range ($l = 50$) did not reduce the accumulative errors. Except for a short range, position errors did not largely depend on the range of local optimization.

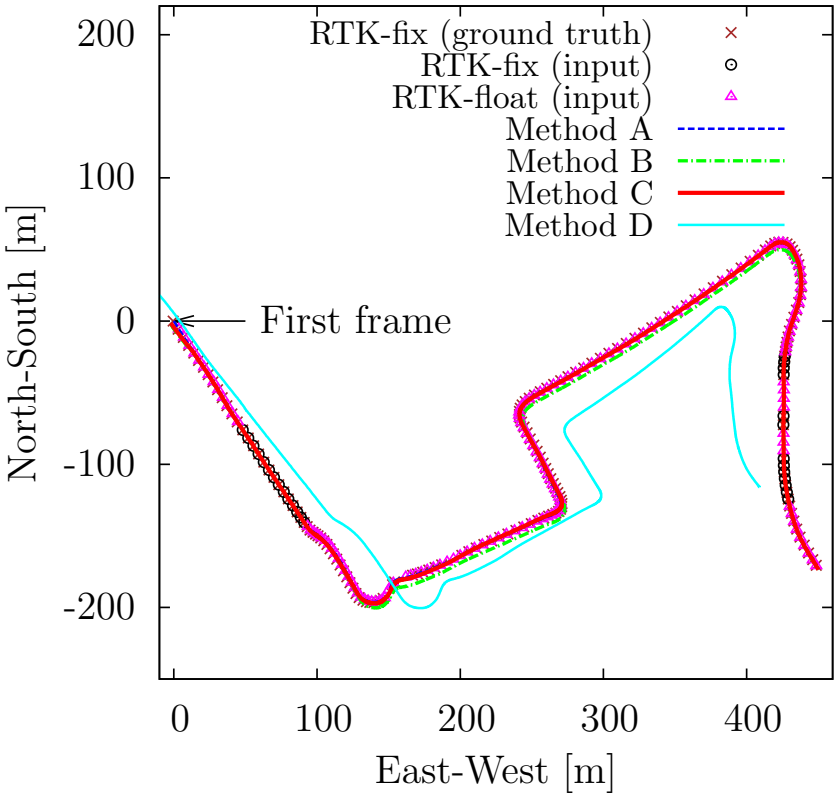


(a) Horizontal 2D positions

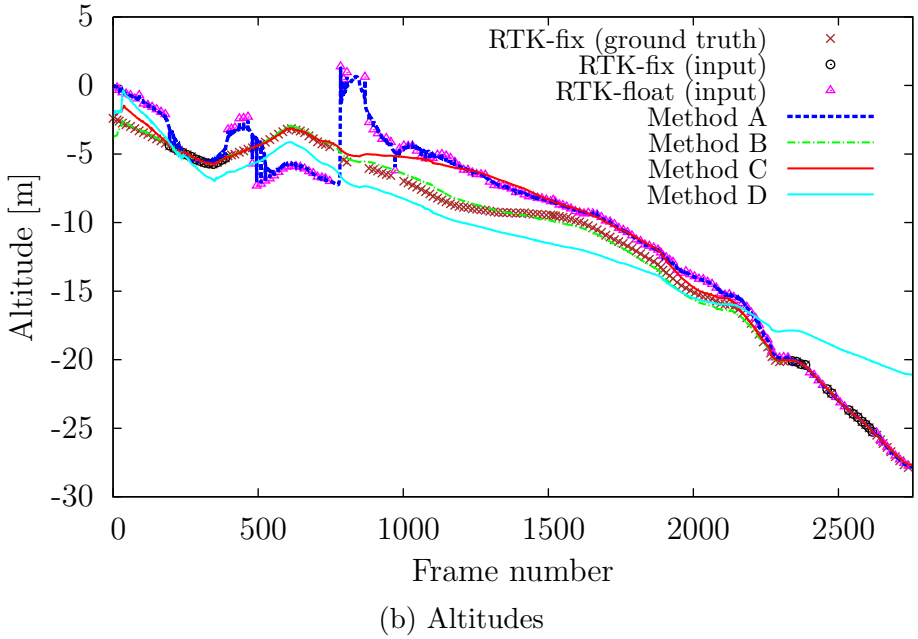


(b) Altitudes

Figure 2.8: Input GPS positions (experiment 1).



(a) Horizontal 2D positions



(b) Altitudes

Figure 2.9: Estimated GPS positions (experiment 1).

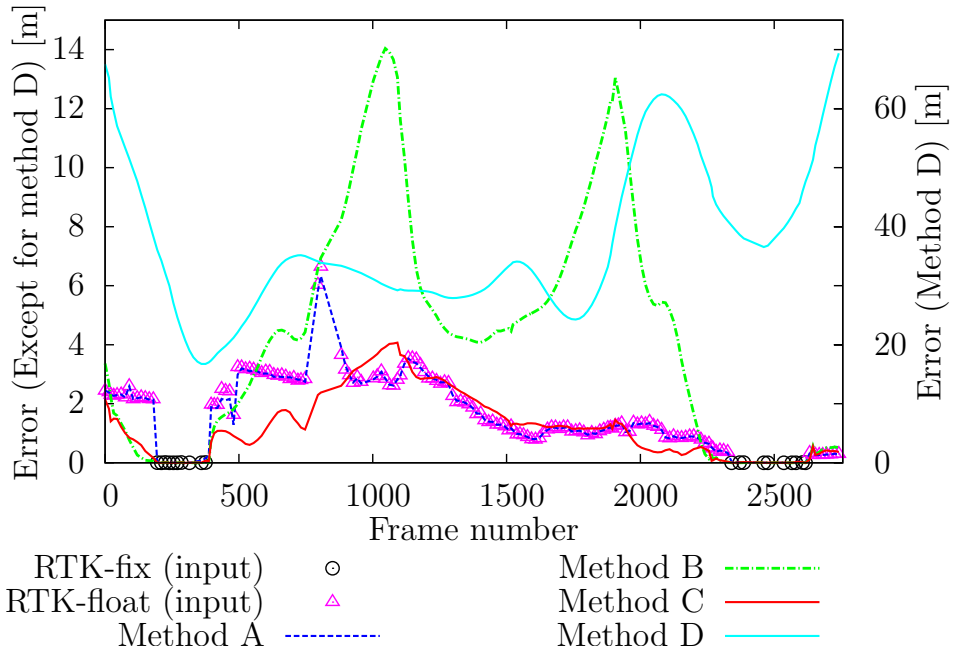


Figure 2.10: Position errors in each frame (experiment 1).

Table 2.2: Comparison of position errors (experiment 1) [m].

Method	Average	Std. dev.	Max
RTK-fix	0.003	0.003	0.010
RTK-float	1.844	1.093	6.641
A	1.553	1.182	6.326
B	4.298	3.944	14.041
C	1.217	1.070	4.073
D	37.189	13.033	69.388

Table 2.3: Relationship between l and average position errors (experiment 1) [m].

l	50	100	200	400	800
Avg. error	13.431	1.284	1.217	1.232	1.219

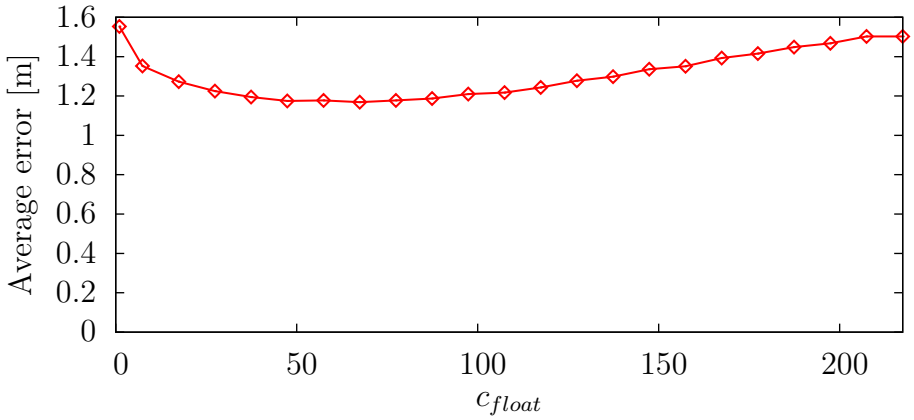


Figure 2.11: Relationship between weight c_{float} and average position errors (experiment 1).

Figure 2.11 shows the average position errors from the proposed method with variable weighting coefficients for low-confidence GPS positioning around $c_{\text{float}} = 107.4$. Note that $c_{\text{float}} = 1$ indicates Method A, which does not consider the GPS positioning confidence. The results demonstrate that position errors do not largely depend on c_{float}, and that c_{float} obtained from a fixed point observation is not the best value but is still sufficient.

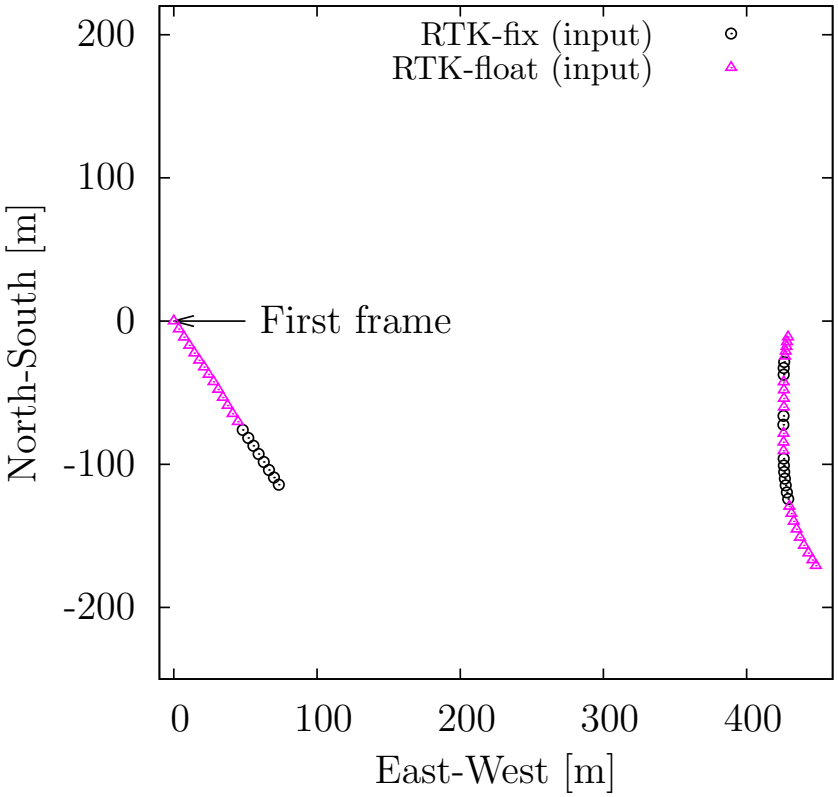
2.4.4 Quantitative Evaluation using Data Including a Long GPS Outage (Experiment 2)

To validate the effectiveness of the parameter fitting to avoid the local minima described in Section 2.3.2, the accuracy of the estimated camera positions from the following methods was compared.

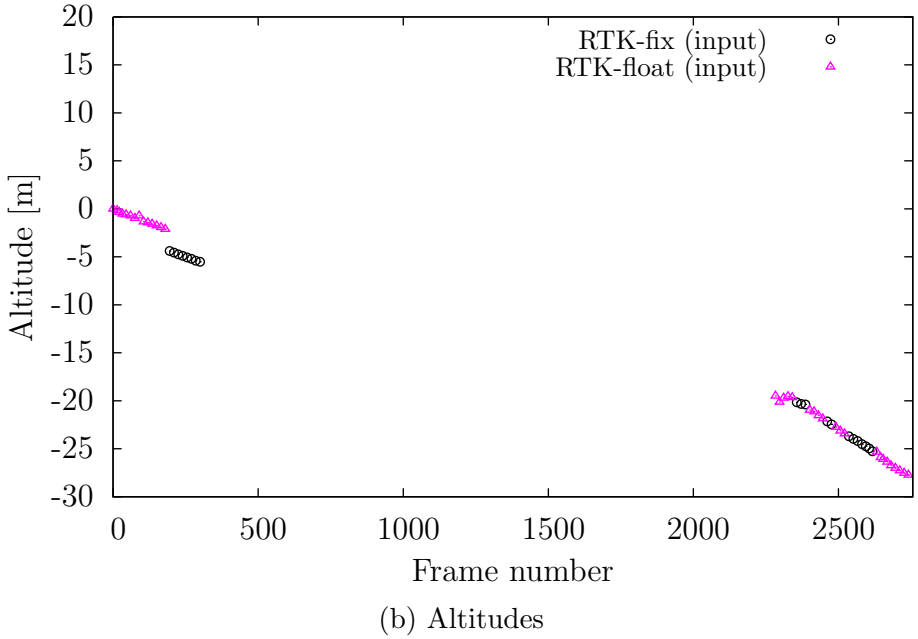
Method C: The proposed method.

Method C': The proposed method without parameter fitting.

In this experiment, to generate the GPS positioning data through a simulation, we assumed city environments in which GPS outages occasionally occur owing to occlusions from buildings and trees. As shown in Figure 2.12, the data generated include a long GPS outage (139 [s], 315-2,271th frames, and 72 % GPS positioning). We employed parameter fitting even when the confidence of the recovered



(a) Horizontal 2D positions



(b) Altitudes

Figure 2.12: Input GPS positions (experiment 2).

GPS positioning was RTK-float because the errors in the RTK-float positioning are usually smaller than the accumulative errors of SfM.

Figure 2.13 shows the estimated GPS positions when the GPS positioning was recovered. For Method C, the results before optimization (right after parameter fitting) and after optimization are shown. The results demonstrate that cameras around only the frame in which the GPS positioning was recovered were drawn to the obtained GPS position in Method C', which does not apply parameter fitting. In Method C, the positions close to the ground truth positions were obtained through parameter fitting and optimization.

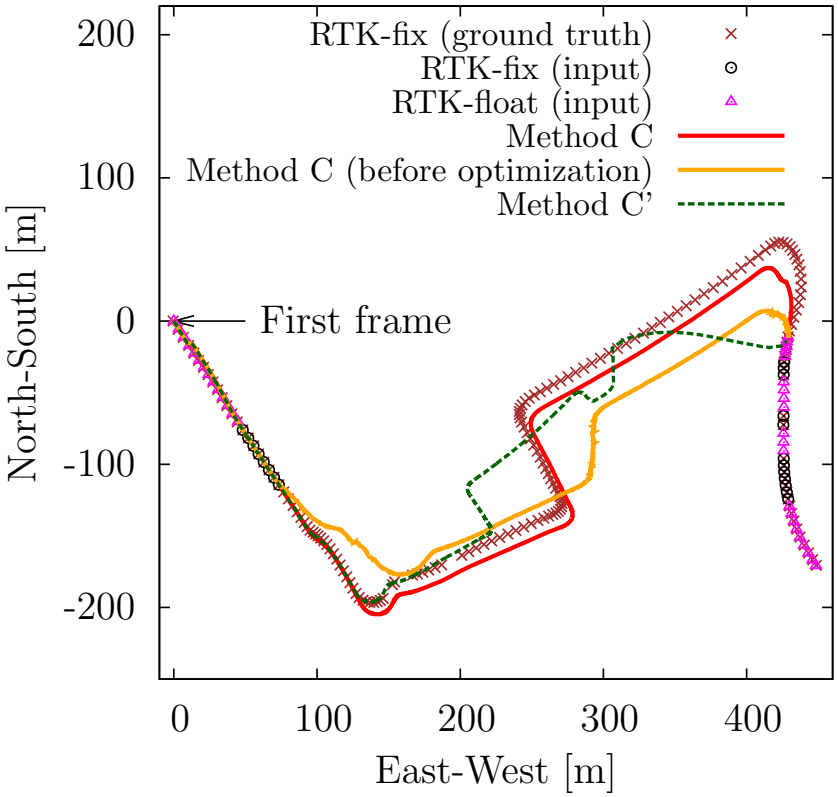
Figures 2.14 and 2.15 show position errors in each frame after global optimization and a change in energy during the sequential process, respectively. Table 2.4 shows the statistics of the position errors. These results show that the errors from Method C' were large during a GPS outage because the energy converged to a local minimum and optimization did not work well. The errors from Method C' were also large at around the last frame because an inconsistency between the 3D positions of the feature points and the camera poses arose from the optimization. In Method C, the accumulative errors were reduced using GPS positioning around the first and last frames. However, a comparison of the results from experiment 1 shows that the accumulative errors were still large during a GPS outage.

2.5. Conclusions

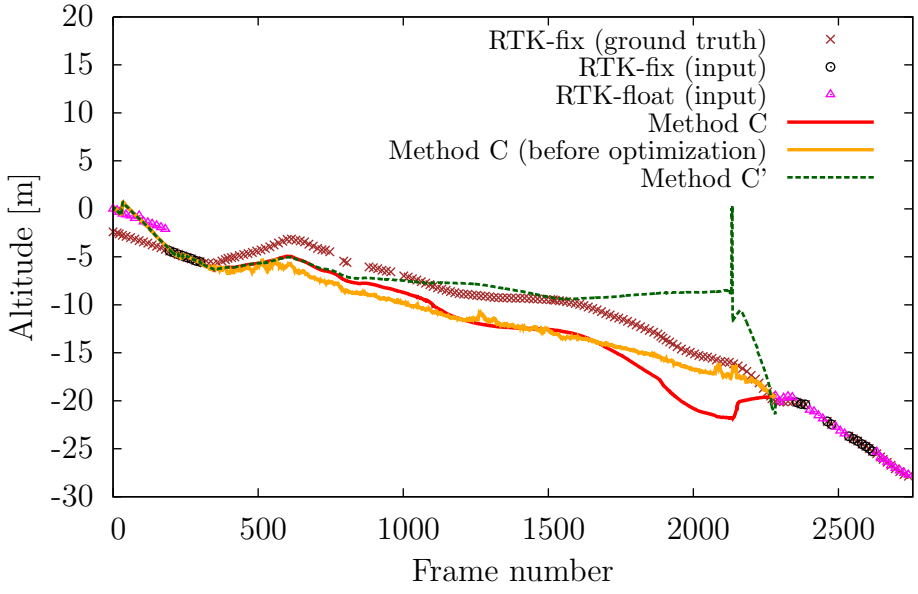
In this chapter, we proposed a method using images and GPS data for accurately estimating camera poses even when the GPS positioning accuracy drops to a

Table 2.4: Comparison of position errors (experiment 2) [m].

Method	Average	Std. dev.	Max
RTK-fix	0.006	0.009	0.038
RTK-float	1.315	0.989	2.594
C	3.593	3.617	14.935
C'	60.299	67.829	277.530



(a) Horizontal 2D positions



(b) Altitudes

Figure 2.13: Estimated GPS positions after a GPS outage (experiment 2).

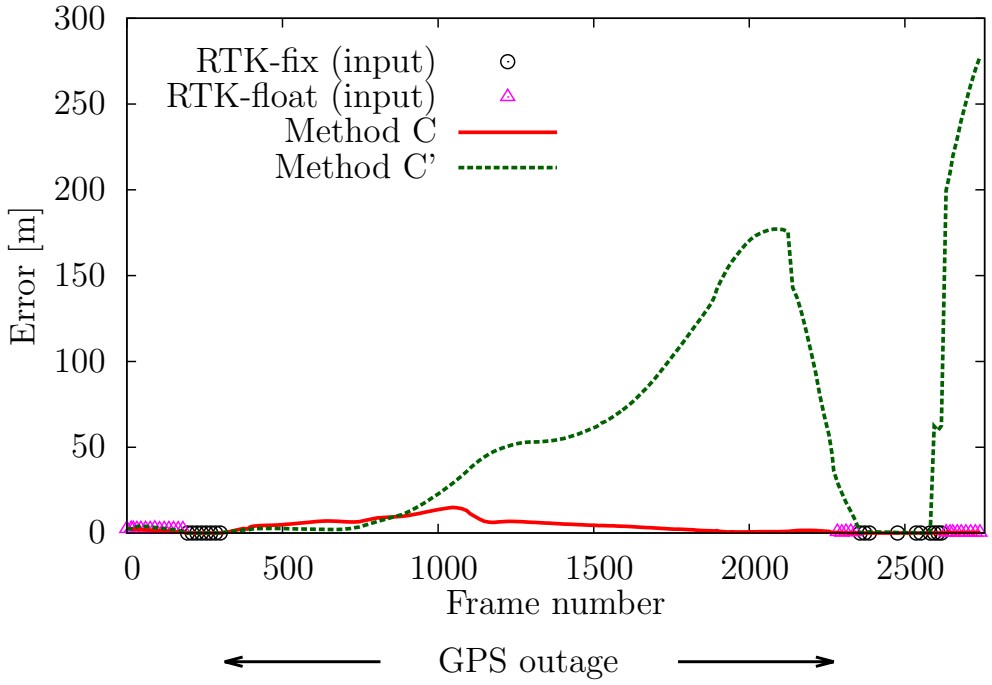


Figure 2.14: Position errors in each frame (experiment 2).

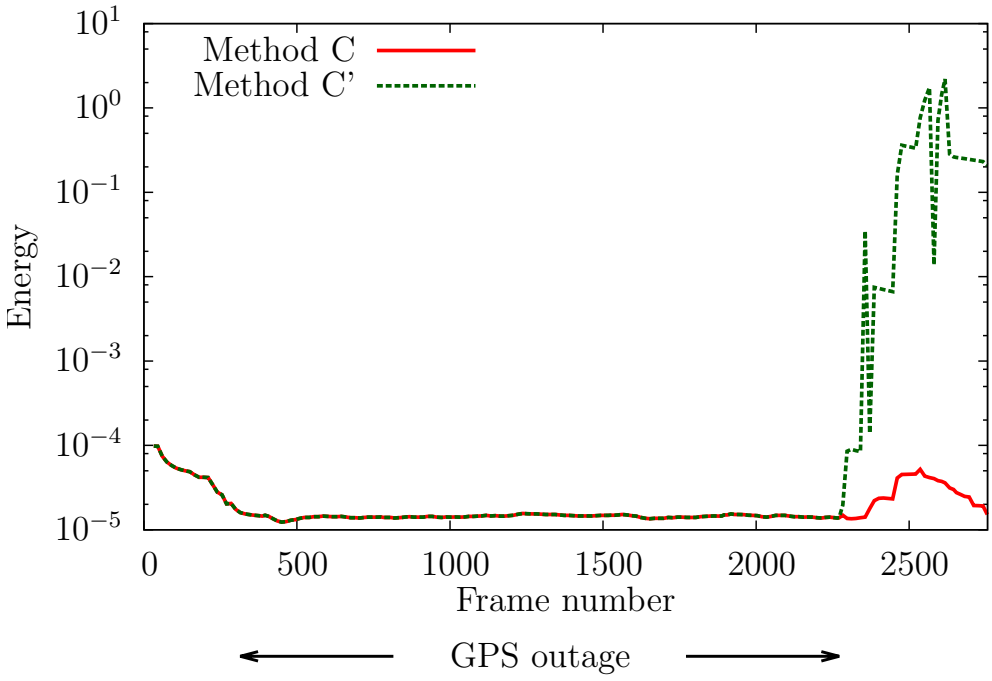


Figure 2.15: Change in energy during sequential process (experiment 2).

low level by introducing weighting coefficients depending on the GPS positioning confidence for extended BA. We also introduced parameter fitting using GPS positions to avoid the local minima during optimization after a GPS outage. We confirmed experimentally that the proposed method provides more accurate camera positions compared with an existing extended-BA method. For more practical use of the proposed method, an automatic weight determination method should be investigated because appropriate weights are dependent on the environment. In addition, combining other sensors such as an IMU is important to reduce accumulative errors during a long GPS outage.

Chapter 3

Sampling-Based Bundle Adjustment using Feature Matches Between Ground-View and Aerial Images

3.1. Introduction

This chapter describes a camera pose estimation method using aerial images as external references that are already available for most outdoor scenes around the world. As mentioned in Chapter 1, the most significant problem in SfM is the accumulation of estimation errors during a long image sequence. To reduce the accumulative errors, we employed the framework of extended BA to the SfM problem using an aerial image as an external reference. Although many kinds of methods using aerial images have been proposed [109–115, 120], to the best of our knowledge, ours is the first method using aerial images as external references in BA. For the successful use of an aerial image as a reference in SfM, successful matching between the aerial image and the ground-view image is very important. To find good matches from unreliable matches, in addition to the use of GPS and gyroscope sensors embedded in most recent smartphones, we use two new methods: (1) RANSAC-based [122] outlier elimination in the feature matching stage

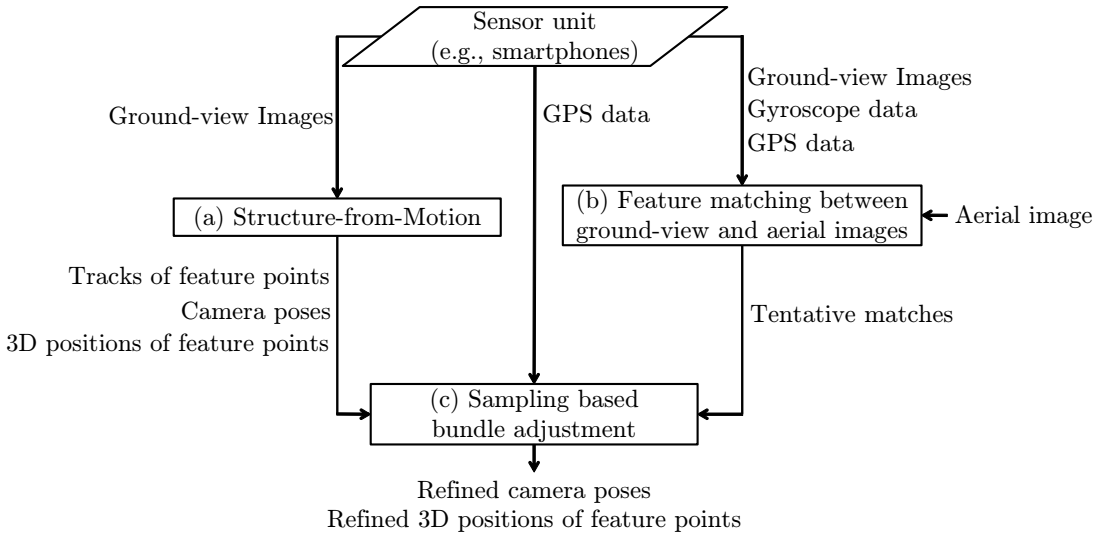


Figure 3.1: Flow of the proposed method using aerial images.

by focusing on the consistency of the orientation and scale extracted from the images by feature descriptors such as SIFT [118], and (2) RANSAC-based outlier elimination in the BA stage using the consistency of the estimated geometry and matches.

As shown in Figure 3.1, the proposed method consists of three processes: (a) SfM, (b) feature matching between the ground-view and aerial images, and (c) sampling-based BA. For process (a), any SfM method can be employed. In the experiments described later, we employed VisualSFM [43] as a state-of-the-art SfM implementation. In the following, we describe processes (b) and (c), the feature matching between the ground-view and aerial images, and sampling-based BA, respectively.

3.2. Feature Matching Between Ground-View and Aerial Images

In this section, we propose a robust method for finding matches between ground-view and aerial images. As shown in Figure 3.2, the method is composed of (1) ground-view image rectification using homography, (2) feature matching, and

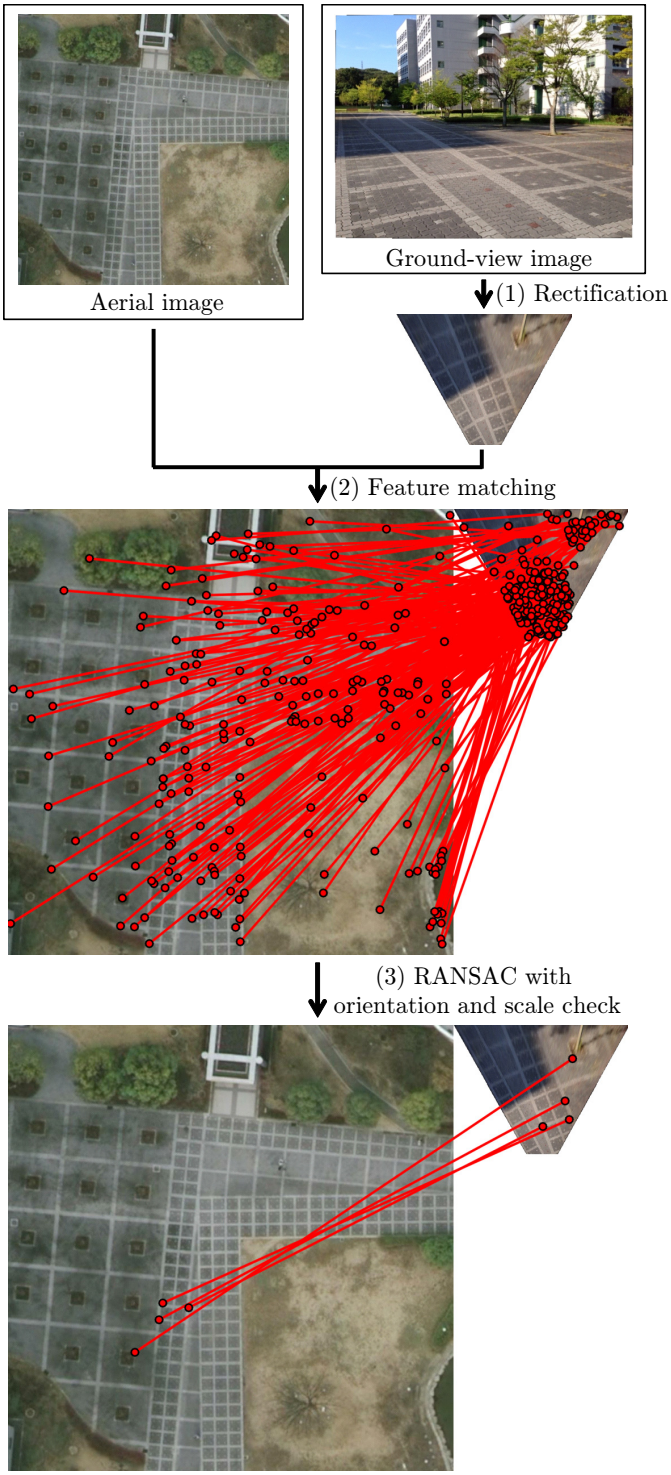


Figure 3.2: Flow of the feature matching.

(3) RANSAC. Here, to achieve robust matching, we propose new criteria for RANSAC using a consistency check of the orientation and scale from a feature descriptor. It should be noted that matching for all input frames is not necessary in our pipeline. Even if we can find only several matched frame candidates, they can be effectively used as references during the BA stage.

3.2.1 Image Rectification using Homography

Before calculating the feature matches using a feature detector and a descriptor, as shown in Figure 3.2, we rectify the ground-view images so that the texture patterns are similar to those of the aerial image. In most cases, aerial images are taken very far away from the ground and can thus be assumed to have been captured by an orthographic camera whose optical axis is directed toward the direction of gravity. To rectify the ground-view images, we also assume that these images contain the ground plane whose normal vector is directed toward the direction of gravity. We then compute a homography matrix using the gravity direction in the camera coordinate system which can be estimated from the vanishing points of parallel lines in the ground-view images or from a gyroscope.

3.2.2 Feature Matching

The feature matches between the rectified ground-view images and the aerial image are calculated. Here, we use GPS data corresponding to the ground-view images to limit the search area in the aerial image. More concretely, we select the region whose center is the GPS position and size is $\iota \times \iota$. In the experiments described later, ι was set to 50 [m]. The feature matches are then calculated using a feature detector and a descriptor. We employed SIFT [118] in our experiments because of its robustness to changes in scale, rotation and illumination.

3.2.3 RANSAC with Orientation and Scale Check

As shown in Figure 3.2, the results of feature matching often include many incorrect matches. To remove these matches, we use RANSAC with a consistency check of the orientation and scale parameters.

For matches between the rectified ground-view images and the aerial image, we can use a similarity transform composed of scale s, rotation θ, and translation $\boldsymbol{\tau}$. During the RANSAC procedure, we randomly sample two matches (the minimum number required to estimate the similarity transform) to compute the similarity transform $(s, \theta, \boldsymbol{\tau})$. Here, we count the number of inlier matches that satisfy

$$|\boldsymbol{a}_k - (s\mathrm{R}(\theta)\boldsymbol{\gamma}_k + \boldsymbol{\tau})| < d_{\mathrm{th}}, \tag{3.1}$$

where $\boldsymbol{a}_k$ and $\boldsymbol{\gamma}_k$ are the 2D positions of the k-th match in the aerial image and rectified ground-view image, respectively, $\mathrm{R}(\theta)$ is the 2D rotation matrix with rotation angle θ, and d_{th} is the threshold. After repeating the random sampling process, the sampled matches with the largest number of inliers are selected.

The problem here is that the distance-based criterion described above cannot successfully find correct matches when a very large number of incorrect matches exist. To achieve more robust matching, we modify the criterion of RANSAC by checking the consistency of the orientation and scale from a feature descriptor. Concretely, we count the number of inliers that simultaneously satisfy Equation (3.1) and the following two conditions.

$$\max\left(\frac{s_{\mathrm{g}k} \cdot s}{s_{\mathrm{a}k}}, \frac{s_{\mathrm{a}k}}{s_{\mathrm{g}k} \cdot s}\right) < s_{\mathrm{th}}, \tag{3.2}$$

$$\mathrm{aad}(\theta_{\mathrm{g}k} + \theta, \theta_{\mathrm{a}k}) < \theta_{\mathrm{th}}, \tag{3.3}$$

where $(s_{\mathrm{a}k}, s_{\mathrm{g}k})$ and $(\theta_{\mathrm{a}k}, \theta_{\mathrm{g}k})$ are the scale and orientation of the feature points for the k-th match on the aerial image and rectified ground-view image, respectively. The function aad returns the absolute angle difference in the domain $[0°, 180.0°]$. Additionally, s_{th} and θ_{th} are the thresholds for the scale and angle, respectively.

3.3. Sampling-Based Bundle Adjustment

Even using the modified RANSAC proposed in the previous section, it is not possible to remove all incorrect matches in principle because repetitive and/or similar patterns may exist, e.g., road signs in actual environments, as shown in Figure 3.3. To overcome this difficulty, we also employ RANSAC for the BA stage by focusing on the consistency between the feature matches and the estimated camera poses from the images.

Figure 3.3: Examples of road signs in an aerial image from Google Maps [maps.google.com].

3.3.1 Definition of Energy Function

To consider the matches between the ground-view and aerial images, the energy function is newly defined. As shown in Figure 3.4, since we deal with perspective ground-view images and an orthographic aerial image, two types of reprojection errors should be considered. The energy function $\mathrm{E}_{\mathrm{aerial}}$ is defined using the reprojection errors for the ground-view (perspective) images $\hat{\Phi}$, and the aerial (orthographic) image Ω, as follows:

$$\mathrm{E}_{\mathrm{aerial}}(\{\boldsymbol{R}_i, \boldsymbol{t}_i\}_{i=1}^{I}, \{\boldsymbol{p}_j\}_{j=1}^{J}) = \hat{\Phi}(\{\boldsymbol{R}_i, \boldsymbol{t}_i\}_{i=1}^{I}, \{\boldsymbol{p}_j\}_{j=1}^{J}) + \omega_{\Omega}\Omega(\{\boldsymbol{p}_j\}_{j=1}^{J}), \quad (3.4)$$

where $\boldsymbol{R}_i$ and $\boldsymbol{t}_i$ represent the rotation and translation from a world coordinate system into a camera coordinate system for the i-th frame, respectively; $\boldsymbol{p}_j$ is the 3D position of the j-th feature point; I and J are the numbers of frames and feature points, respectively; and ω_{Ω} is the weight that balances $\hat{\Phi}$ and Ω. Because the energy function is non-linearly minimized in BA, good initial parameter values

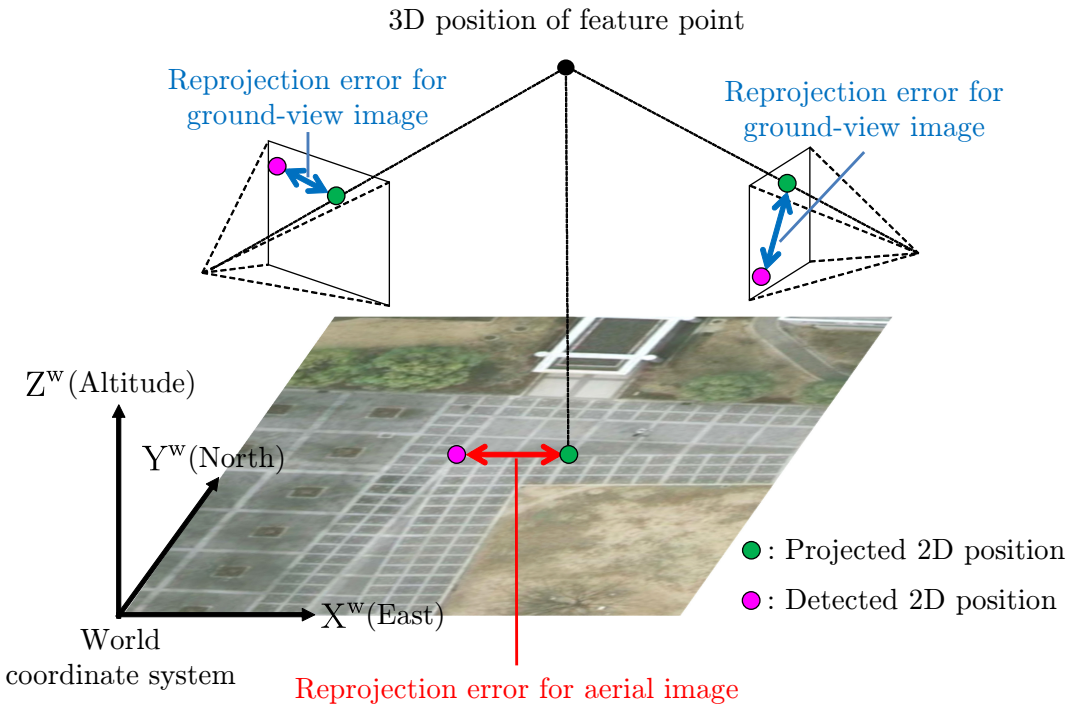
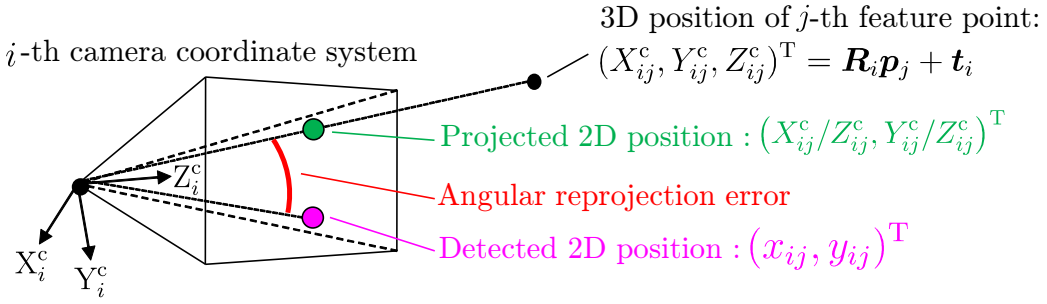


Figure 3.4: Reprojection errors for ground-view (perspective) images and an aerial (orthographic) image.

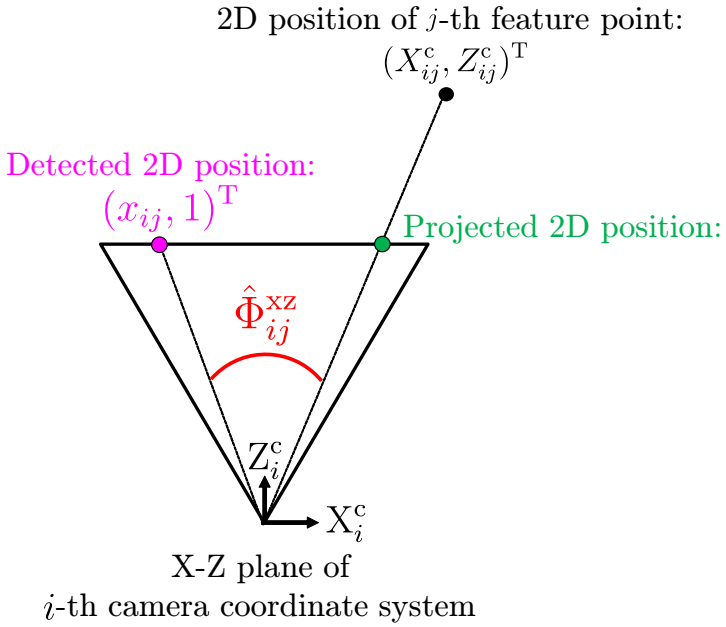
are required to avoid the local minima. Before minimizing the energy function, we fit the parameters estimated by SfM to the GPS positions using a 3D similarity transform. In the following, the energy associated with the reprojection errors $\hat{\Phi}$ and Ω is detailed.

Reprojection Errors for Ground-view Images

The commonly used reprojection errors are defined by employing a pinhole camera model that cannot deal with projections from behind the camera. Such behind projections often occur in BA with external references owing to dynamic movements of the camera poses caused by these references. Here, as shown in Figure 3.5, instead of common squared distance errors on the image plane, we employ



(a) Angular reprojection error



(b) Split angular reprojection error

Figure 3.5: Reprojection error for ground-view (perspective) image.

reprojection errors using the angles of rays as follows:

$$\hat{\Phi}(\{\boldsymbol{R}_i, \boldsymbol{t}_i\}_{i=1}^{I}, \{\boldsymbol{p}_j\}_{j=1}^{J}) = \frac{1}{\sum_{i=1}^{I} |\boldsymbol{P}_i|} \sum_{i=1}^{I} \sum_{j \in \boldsymbol{P}_i} \left((\hat{\Phi}_{ij}^{\mathrm{xz}})^2 + (\hat{\Phi}_{ij}^{\mathrm{yz}})^2 \right) \tag{3.5}$$

$$\hat{\Phi}_{ij}^{\mathrm{xz}} = \angle \left(\begin{pmatrix} x_{ij} \\ 1 \end{pmatrix}, \begin{pmatrix} X_{ij}^{\mathrm{c}} \\ Z_{ij}^{\mathrm{c}} \end{pmatrix} \right) \tag{3.6}$$

$$\hat{\Phi}_{ij}^{\mathrm{yz}} = \angle \left(\begin{pmatrix} y_{ij} \\ 1 \end{pmatrix}, \begin{pmatrix} Y_{ij}^{\mathrm{c}} \\ Z_{ij}^{\mathrm{c}} \end{pmatrix} \right) \tag{3.7}$$

$$(X_{ij}^{\mathrm{c}}, Y_{ij}^{\mathrm{c}}, Z_{ij}^{\mathrm{c}})^{\mathrm{T}} = \boldsymbol{R}_i \boldsymbol{p}_j + \boldsymbol{t}_i \tag{3.8}$$

where $\boldsymbol{P}_i$ is a set of feature points detected in the i-th frame. Function $\angle$ returns an angle between two vectors. Note that $(x_{ij}, y_{ij})^{\mathrm{T}}$ is the detected 2D position of the j-th feature points in the i-th frame.

Here, as mentioned in [36], convergence of energy is very poor with angular reprojection error $(\hat{\Phi}_{ij})^2 = \angle((x_{ij}, y_{ij}, f_i)^{\mathrm{T}}, (X_{ij}, Y_{ij}, Z_{ij})^{\mathrm{T}})^2$. Then, as shown in Figure 3.5(b), we split the angular reprojection error into the xz and yz components to simplify the Jacobian matrix of $\mathrm{E}_{\mathrm{aerial}}$ required by non-linear least squares methods such as the Levenberg-Marquardt method. In this definition, $\hat{\Phi}_{ij}^{\mathrm{xz}}$ and $\hat{\Phi}_{ij}^{\mathrm{yz}}$ do not depend on the y and x components of $\boldsymbol{t}_i$, respectively. We confirmed experimentally that this splitting largely affects the convergence performance.

Reprojection Errors for an Aerial Image

As shown in Figure 3.6, the reprojection errors for an aerial (orthographic) image can be defined as follows:

$$\Omega(\{\boldsymbol{p}_j\}_{j=1}^{J}) = \frac{1}{\sum_{i \in \boldsymbol{M}} |\boldsymbol{A}_i|} \sum_{i \in \boldsymbol{M}} \sum_{j \in \boldsymbol{A}_i} \left| \boldsymbol{a}_j - (X_j^{\mathrm{w}}, Y_j^{\mathrm{w}})^{\mathrm{T}} \right|^2, \tag{3.9}$$

$$(X_j^{\mathrm{w}}, Y_j^{\mathrm{w}}, Z_j^{\mathrm{w}})^{\mathrm{T}} = \boldsymbol{p}_j \tag{3.10}$$

where $\boldsymbol{M}$ is a set of frames in which the feature matches between the ground-view and aerial images are obtained, $\boldsymbol{A}_i$ is a set of feature points matched to the aerial image in the i-th frame, and $\boldsymbol{a}_j$ is the 2D position of the j-th feature point in the aerial image.

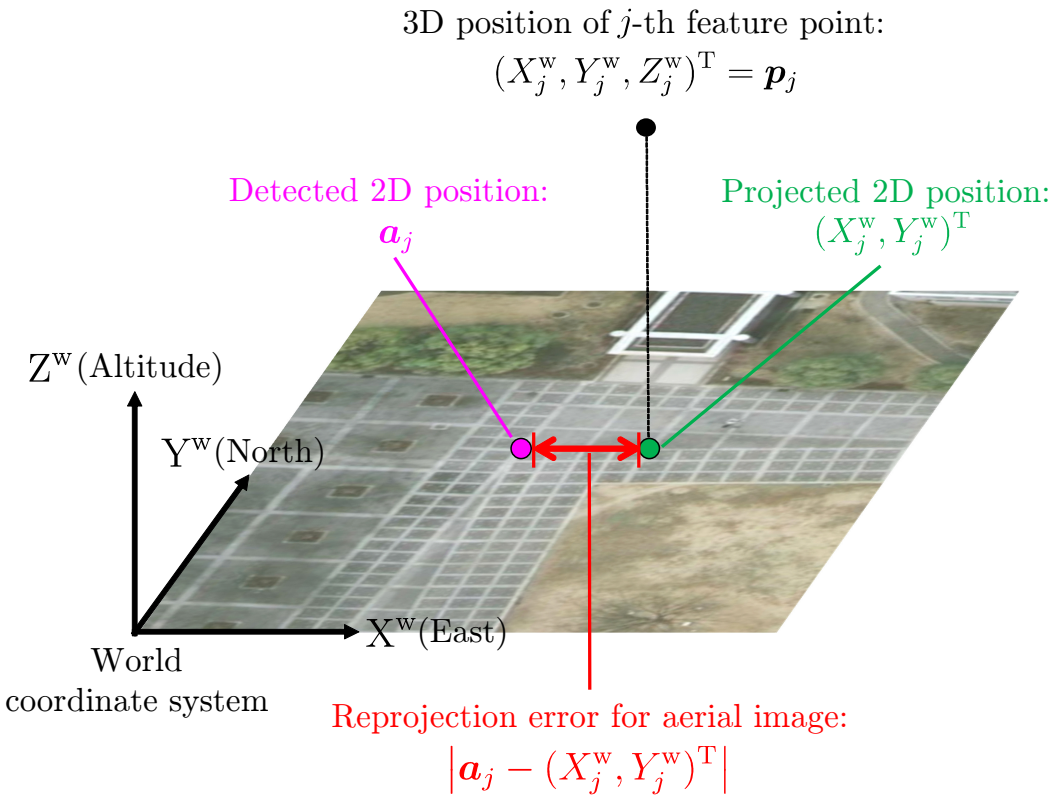


Figure 3.6: Reprojection error for an aerial (orthographic) image.

3.3.2 RANSAC for Bundle Adjustment

The RANSAC scheme is introduced in BA using the consistency between the feature matches and the estimated camera poses from the images. First, we randomly sample n frames from the matched frame candidates and apply BA using feature matches included in the sampled frames, i.e., using a set of selected frames $\boldsymbol{M}'$ instead of $\boldsymbol{M}$ in Equation (3.9). We then count the number of inlier frames that satisfy the following condition.

$$\underset{j \in \boldsymbol{A}_i}{\mathrm{average}}(\alpha_{ij}) < \alpha_{\mathrm{th}} \tag{3.11}$$

where α_{ij} is an angular reprojection error of the j-th feature point in the aerial image coordinate system, as shown in Figure 3.7, and α_{th} is the threshold. Here, α_{ij} can be computed as follows:

$$\alpha_{ij} = \angle \left(\boldsymbol{a}_j - \mathrm{pr}_{\mathrm{xy}}(-\boldsymbol{R}_i^{\mathrm{T}}\boldsymbol{t}_i), \mathrm{pr}_{\mathrm{xy}}\left(\boldsymbol{R}_i^{\mathrm{T}}(x_{ij}, y_{ij}, 1)^{\mathrm{T}}\right)\right), \tag{3.12}$$

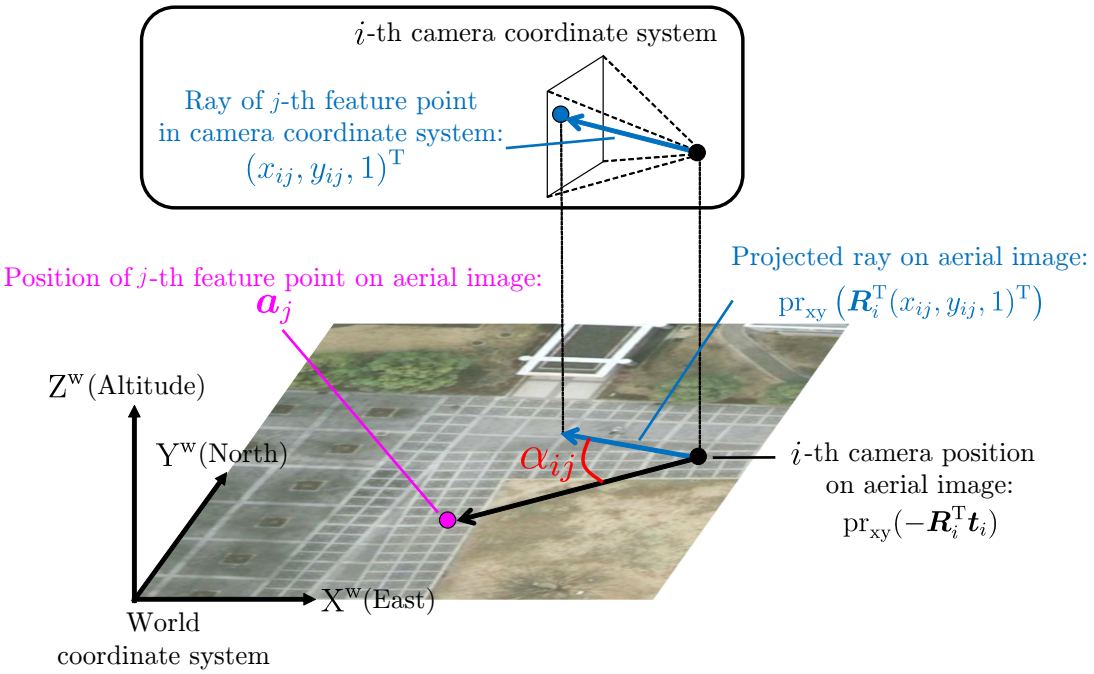


Figure 3.7: Criterion used in RANSAC for BA.

where $\mathrm{pr}_{\mathrm{xy}}$ is a function projecting a 3D point onto the x-y plane (the aerial image coordinate system).

After repeating the random sampling process at the given times, sampled frames with the largest number of inlier frames are selected. Finally, camera poses are refined by reapplying BA using the feature matches with the selected inlier frames as references.

3.4. Experiments

To validate the effectiveness of the proposed method, we quantitatively evaluated the performances of the sampling-based BA as well as the feature matching process using two datasets: (1) data captured by a hand-held sensor unit on the textured ground, and (2) data captured by a car-mounted sensor unit on a roadway. In the following, we first describe the common setup for the two experiments. The results of each experiment are then detailed.

3.4.1 Experimental Setup

We used an iPhone 5 (Apple) as a sensor unit including a camera, GPS, and a gyroscope. The GPS and gyroscope measured the position at 1 [Hz] and the direction of gravity for every frame, respectively. We also used an RTK-GPS (Topcon GR-3, 1 [Hz], horizontal positioning accuracy in specification sheet is 0.01 [m]) to obtain the ground truth positions. The positions from the GPS data were assigned temporally to the nearest frame. As the external references, we downloaded the aerial images covering the area used in the experiments from Google Maps [maps.google.com], whose coordinate system is associated with the metric scale.

To obtain the initial values for the BA, we employed VisualSFM [43] as a state-of-the-art SfM implementation. For non-linear minimization, we used Ceres-Solver [52]. We experimentally set $d_{\mathrm{th}} = 2$ [pixel], $s_{\mathrm{th}} = 2$ and $\theta_{\mathrm{th}} = 40$ [°] for the feature matching, and $\omega_{\Omega} = 10^{-5}$ and $\alpha_{\mathrm{th}} = 5.0$ [°] for the BA. We evaluated the accuracy of the proposed method by comparing the following methods.

- BA without references [43]
- BA with references without RANSAC, using all the matches obtained by the feature matching process
- BA with references and RANSAC

Since the BA without references cannot estimate absolute camera poses, we fitted the camera positions estimated using SfM to the ground truths through a similarity transform.

3.4.2 Quantitative Evaluation using Data Captured on Textured Ground (Experiment 1)

In this experiment, we used video images (640 [pixel] × 480 [pixel], 2,471 frames, 494 [s]) captured by a hand-held sensor unit on a textured ground. Figures 3.8 and 3.9 show example input ground-view images and an aerial image (19.2 [pixel] = 1 [m].)

(a) 1st frame (b) 350th frame

(c) 700th frame (d) 1,050th frame

(e) 1,400th frame (f) 1,750th frame

(g) 2,100th frame (h) 2,450th frame

Figure 3.8: Example input ground-view images (experiment 1).

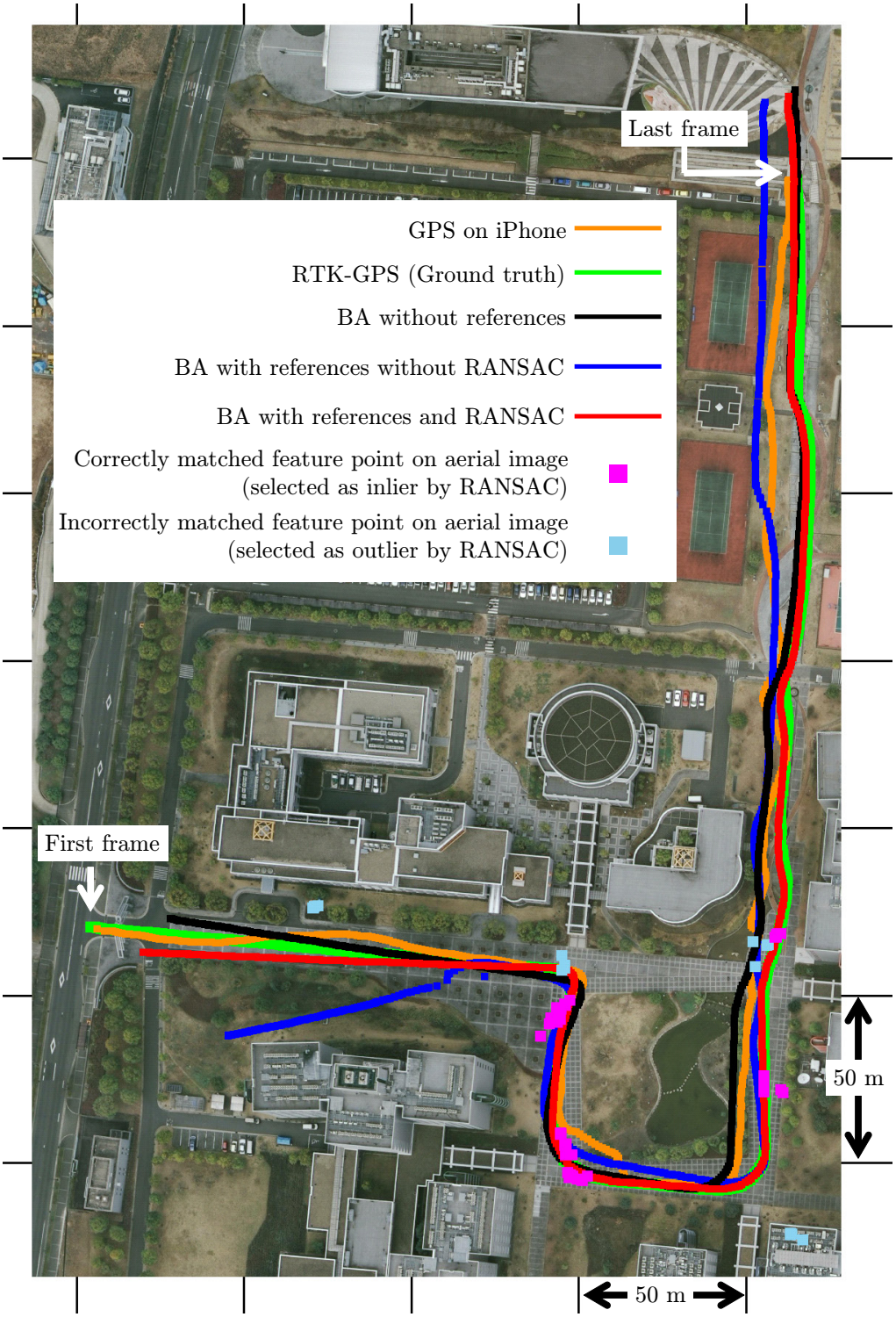


Figure 3.9: Experimental environment and results (experiment 1).

Quantitative Evaluation of Feature Matching

In this experiment, we first evaluated the effectiveness of the proposed feature matching process including RANSAC using the scale and orientation check described in Section 3.2. Here, we tested RANSAC with variable thresholds s_{th} and θ_{th}. To count the number of correctly matched frames, we first selected frames that have four or more inlier matches after RANSAC. From these frames, we manually counted frames whose matches were correct.

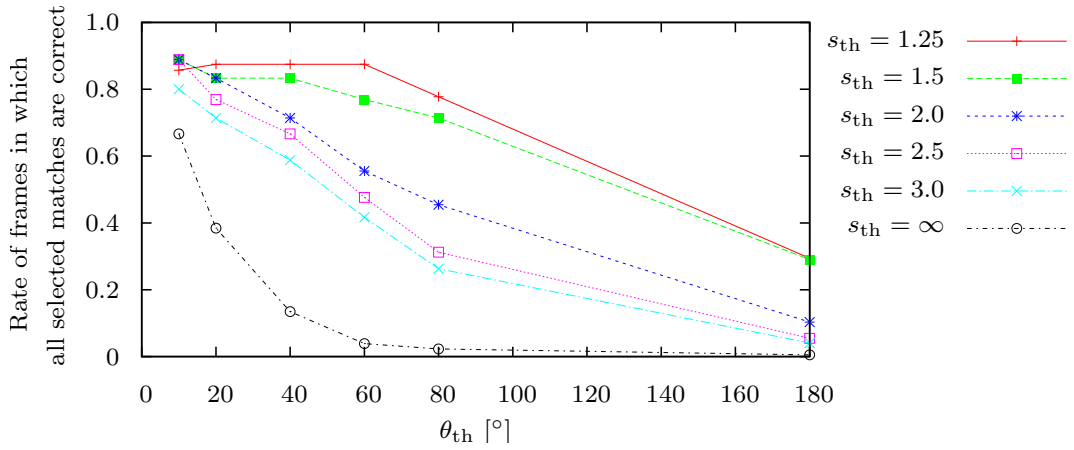
Figure 3.10 shows the rate and number of frames in which all selected matches were correct. Note that $s_{\mathrm{th}} = \infty$ and $\theta_{\mathrm{th}} = 180.0$ [°] mean that the orientation check and scale check were disabled, respectively. The results indicate that the rate was significantly improved through the scale and orientation check. We can also confirm that small values of s_{th} and θ_{th} tend to increase this rate. However, the number of correctly matched frames, which is important for optimizing the camera poses using BA, was decreased when using small thresholds. In the following experiments, we employed feature matches with $s_{\mathrm{th}} = 2$ and $\theta_{\mathrm{th}} = 40$ [°].

Figure 3.11 shows the effects of the scale and orientation check for two sampled images. In both cases, RANSAC without scale and orientation check could not select any correct matches, whereas the proposed RANSAC with scale and orientation check was able to do so. However, as shown in Figure 3.12, incorrect matches still remain even when we used both scale and orientation check because similar patterns exist.

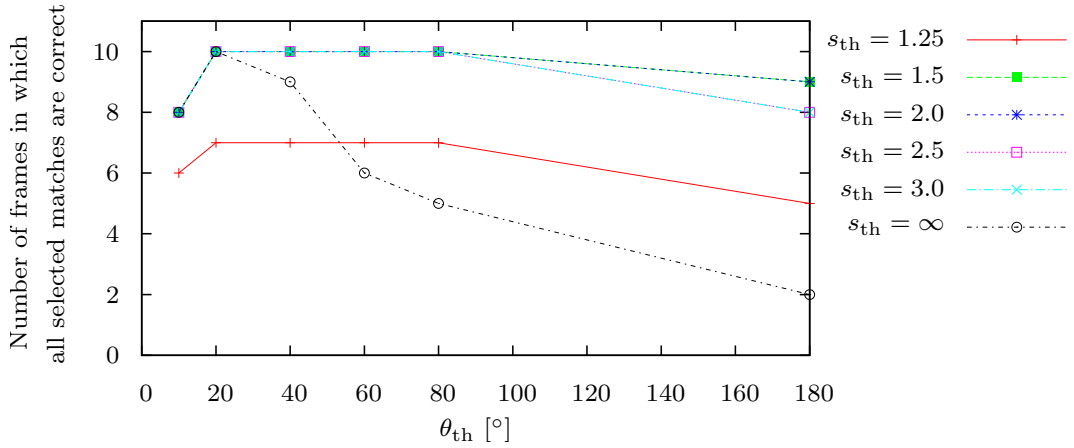
Quantitative Evaluation of Bundle Adjustment

In this experiment, we evaluated the effectiveness of BA with RANSAC, as described in Section 3.3. In this stage, frames with GPS data were sampled (650 out of 2,471 frames) and used to reduce the computational time. As external references, we used the frames and feature matches selected through the orientation and scale check described in the previous section. Here, ten out of fourteen frames had correct matches.

We first investigated the influence of weight ω_Ω for balancing two types of reprojection errors in the energy function of the BA. Figure 3.13 shows the average position errors from the BA with variable weight ω_Ω using all of the correctly matched frames. This result demonstrates that position errors did not largely

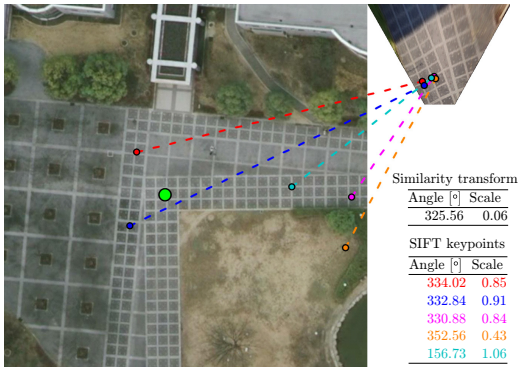


(a) Rate of frames in which all selected matches are correct

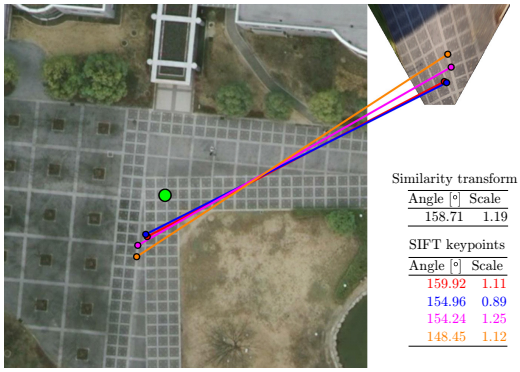


(b) Number of frames in which all selected matches are correct

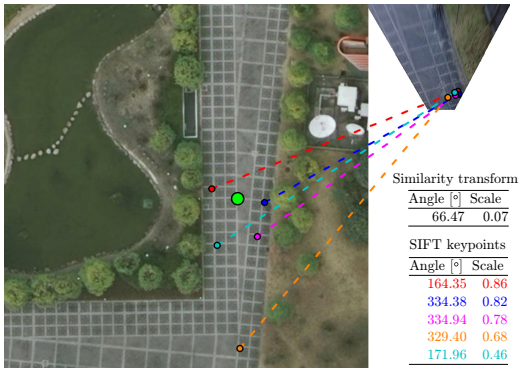
Figure 3.10: Rates and numbers of frames in which all selected matches are correct (experiment 1).



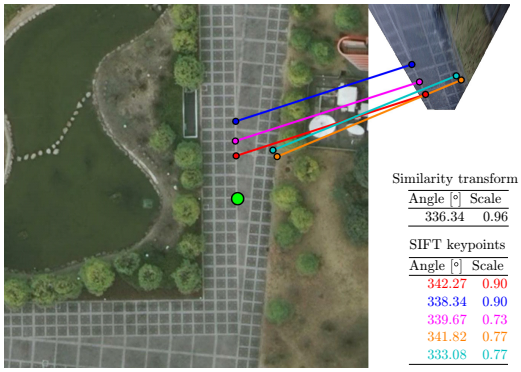
(a) Without orientation and scale check



(b) With scale check, with/without orientation check

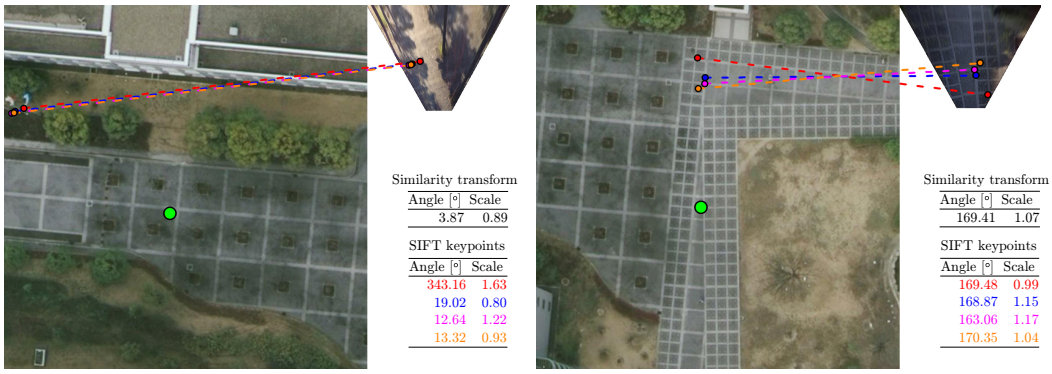


(c) Without orientation and scale check



(d) With orientation check, with/without scale check

Figure 3.11: Selected inliers for example images (experiment 1). The solid and dashed lines represent correct and incorrect matches, respectively. The relative angle and scale of the matched feature points are shown in bottom-right table along with the corresponding line colors. The green points are the ground truths of the camera positions. Note that RANSAC with/without orientation check for (b) and scale check for (d) gave the same results.



(a) (b)

Figure 3.12: Examples of incorrect matches by RANSAC using orientation and scale check (experiment 1). The interpretations of the symbols are the same as in Figure 3.11.

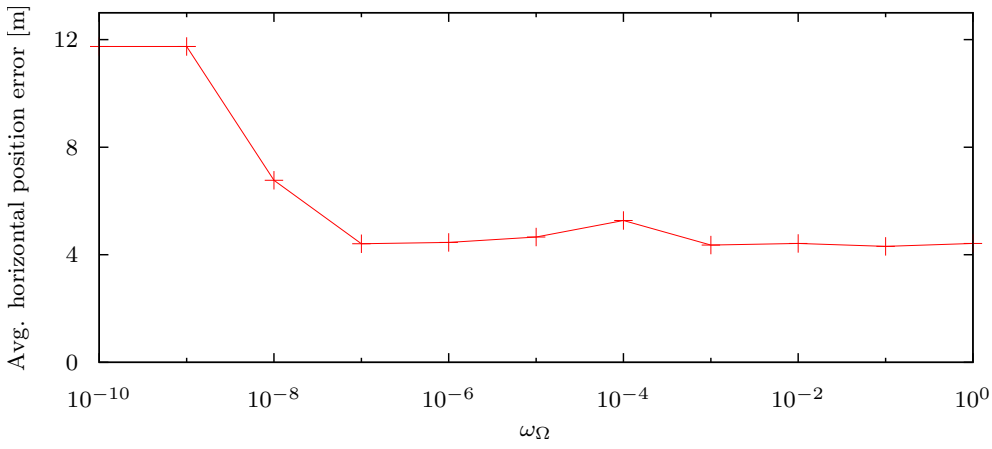


Figure 3.13: Relationship between weight ω_Ω and average horizontal position error (experiment 1).

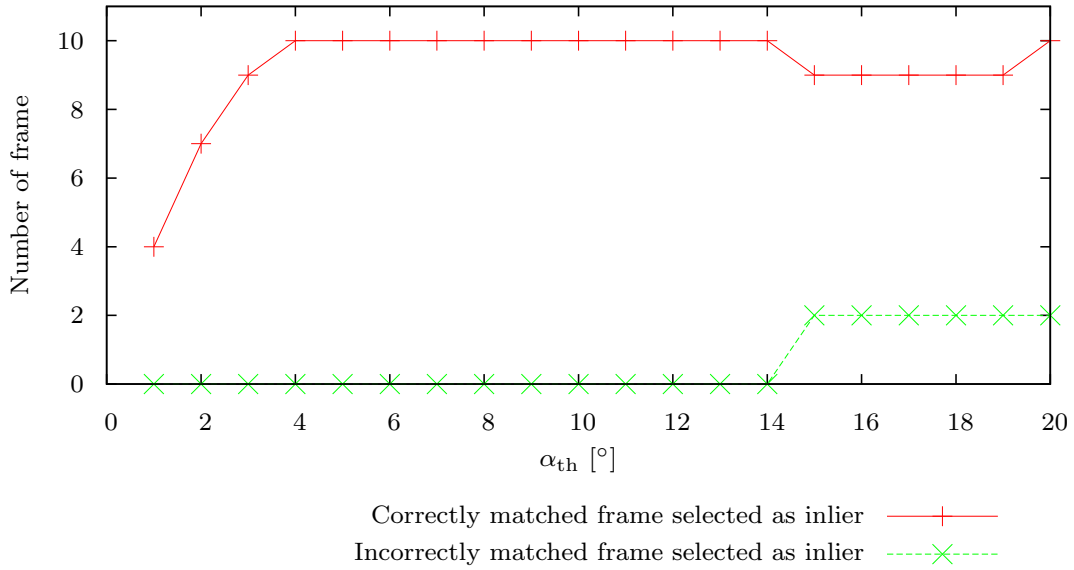


Figure 3.14: Number of inlier frames with variable threshold α_{th} (experiment 1).

depend on weight ω_Ω except when small values were applied. In the following experiments, we employed $\omega_\Omega = 10^{-5}$.

We next evaluated the proposed RANSAC in terms of its capability to select frames with correct matches. Here, we experimentally set $n = 4$ and tested 100 trials. To efficiently optimize the camera poses using feature matches between ground-view and aerial images, we modified the random sampling process of frames in RANSAC so that the distances between average positions of matches on an aerial image were 25 [m] or more. Figure 3.14 shows the numbers of inlier frames from RANSAC with variable threshold α_{th}. The results demonstrate that incorrectly matched frames were selected as inliers by large values of α_{th}, and that the numbers of correctly matched frames decreased by small values of α_{th}. In the following experiments, we employed $\alpha_{\mathrm{th}} = 5.0$ [°].

We also checked the number of inlier frames selected in each trial with $\alpha_{\mathrm{th}} = 5.0$ [°]. Figure 3.15 shows the number of trials and inlier frames derived by each trial. From this figure, we can see that the sampled frames without incorrect matches tend to increase the number of inlier frames. This result demonstrates that the criterion of RANSAC in the BA stage works successfully. We also confirmed that the trials that derived the largest number of inlier frames successfully

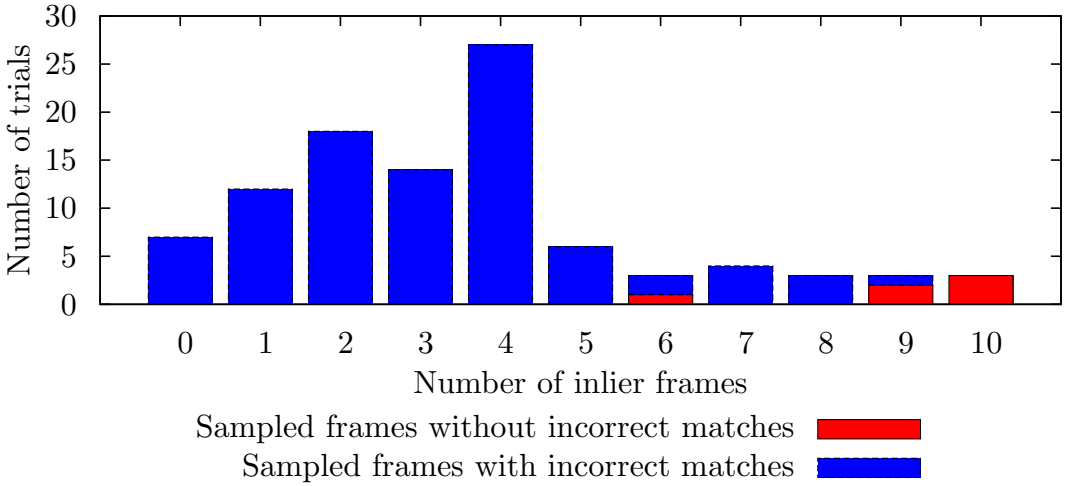


Figure 3.15: Number of trials and inlier frames derived by each trial (experiment 1).

selected all of the correct matches.

Figures 3.9 and 3.16 show the estimated camera positions and horizontal position errors for each frame, respectively. These results demonstrate that the estimated camera positions from BA without references were affected by the accumulative errors. The BA without RANSAC was affected by the incorrect matches. The proposed BA with RANSAC reduced the accumulative errors. It should be noted that, at the end of the sequence, the accumulative errors still remained because the ground was not level, and no matches were therefore found.

3.4.3 Quantitative Evaluation using Data Captured on Roadways (Experiment 2)

In this experiment, we used video images (640 [pixel] $\times$ 480 [pixel], 7,698 frames, 396 [s]) captured by a car-mounted sensor unit on a roadway. Figures 3.17 and 3.18 show example input ground-view images and an aerial image (approximately 22.3 [pixel] = 1 [m]). Note that we manually excluded frames captured when the car was stopped at a traffic light.

We first applied the feature matching process including RANSAC through scale and orientation check. After selecting frames with four or more inlier

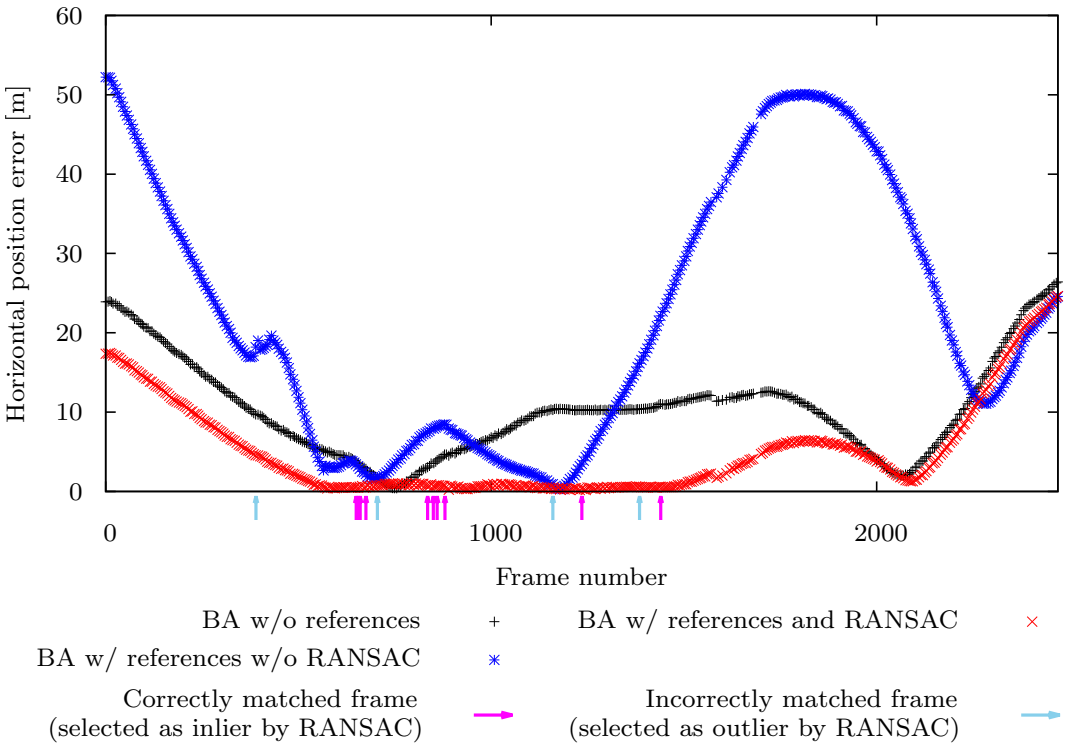


Figure 3.16: Horizontal position error in each frame (experiment 1).

matches, we obtained 37 frames (28 frames without incorrect matches and nine frames with incorrect matches). We then applied RANSAC during the BA stage using frames with GPS data (739 out of 7,698 frames). Here, we experimentally set $n = 7$. To efficiently optimize the camera poses using feature matches between ground-view and aerial images, we modified the random sampling process of frames in RANSAC so that the distances between average positions of matches on an aerial image were 100 [m] or more. After 100 trials, the trial that derived the largest number of inlier frames selected 22 frames as inliers (nineteen frames without incorrect matches and three frames with incorrect matches) and fifteen frames as outliers (nine frames without incorrect matches and six frames with incorrect matches). Figures 3.19 and 3.20 show example frames selected as inliers and outliers, respectively. As shown in Figure 3.19, the frames with incorrect matches were selected as inlier frames by RANSAC because the positions of the incorrect matches on the aerial image were close to the correct positions. Figures 3.18 and 3.21 show the estimated camera positions and horizontal position errors

(a) 1st frame

(b) 1,100th frame

(c) 2,200th frame

(d) 3,300th frame

(e) 4,400th frame

(f) 5,500th frame

(g) 6,600th frame

(h) 7,698th frame

Figure 3.17: Example input ground-view images (experiment 2).

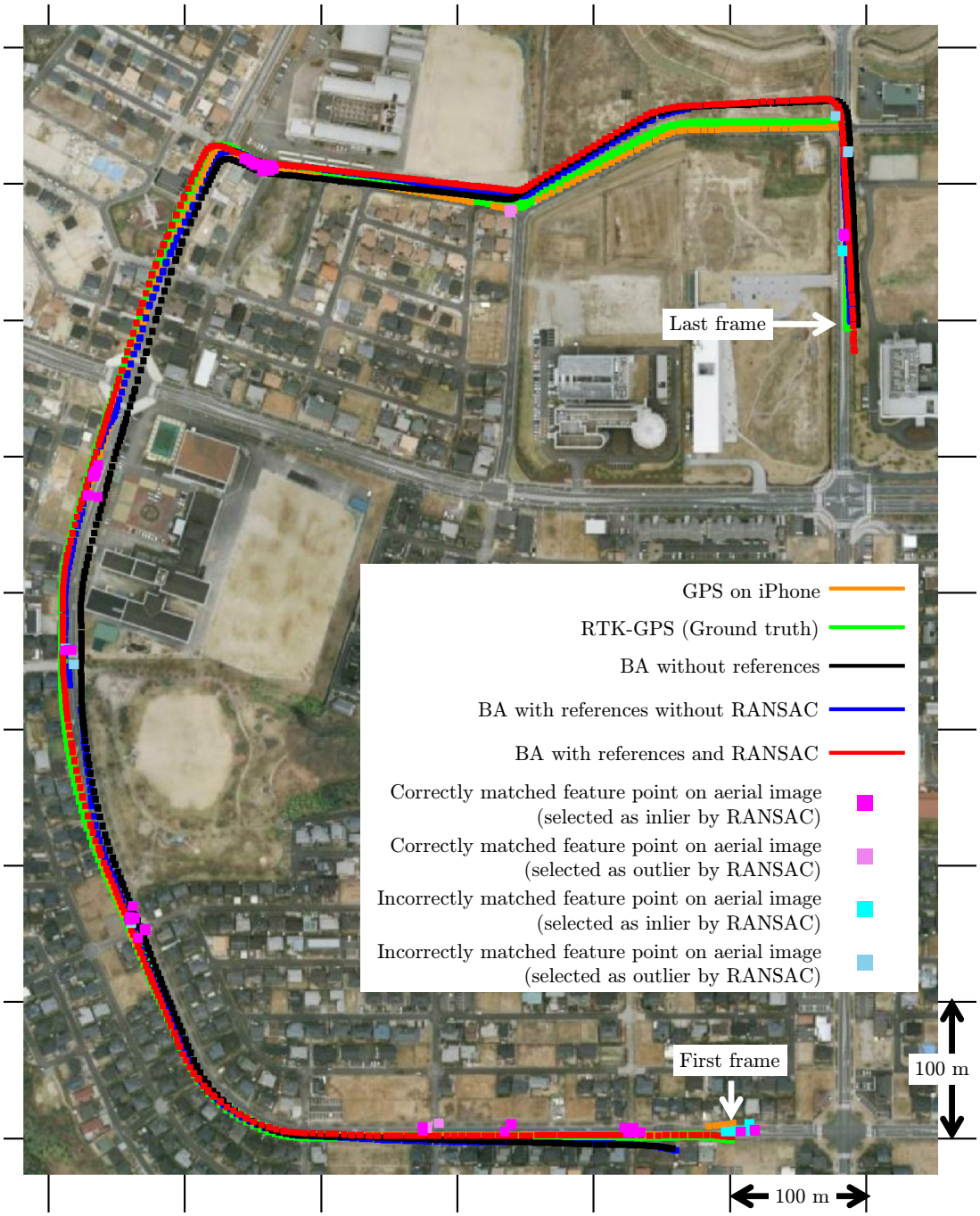
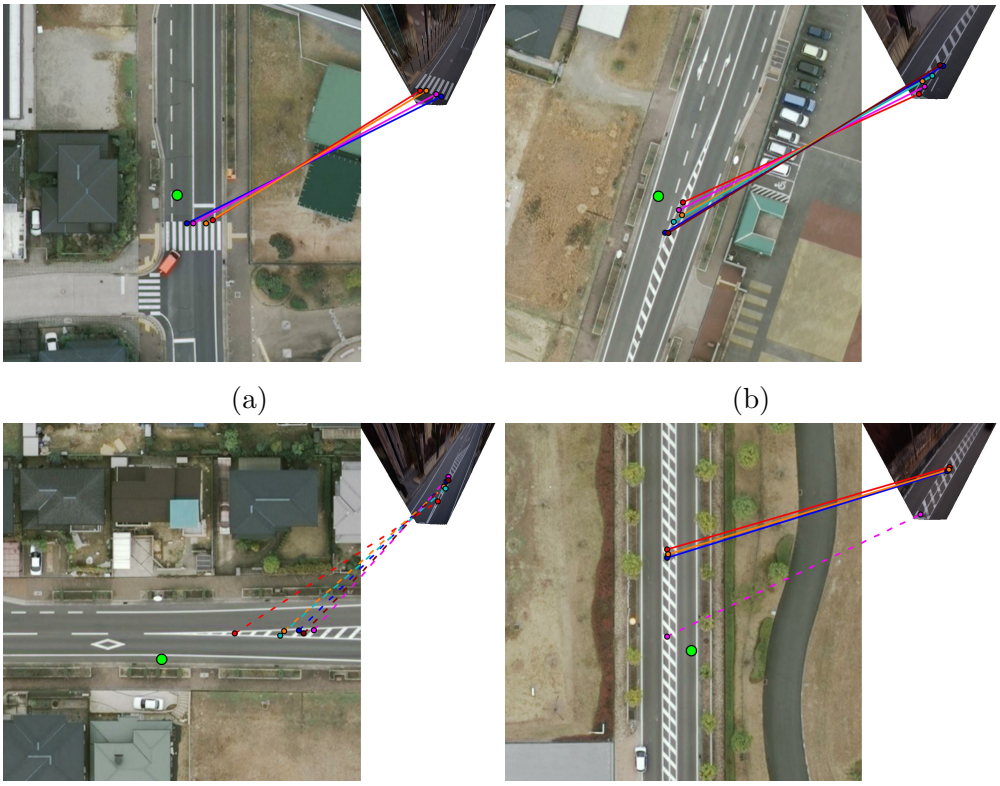


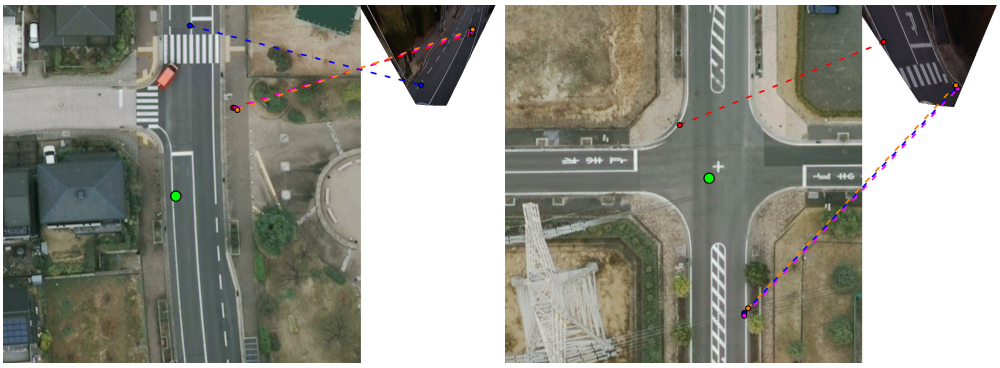
Figure 3.18: Experimental environment and results (experiment 2).



(a) (b)

(c) (d)

Figure 3.19: Examples of frames selected as inliers by RANSAC during the BA stage (experiment 2). The solid and dashed lines represent correct and incorrect matches, respectively.



(a) (b)

Figure 3.20: Examples of frames selected as outliers by RANSAC during the BA stage (experiment 2). The dashed lines represent incorrect matches.

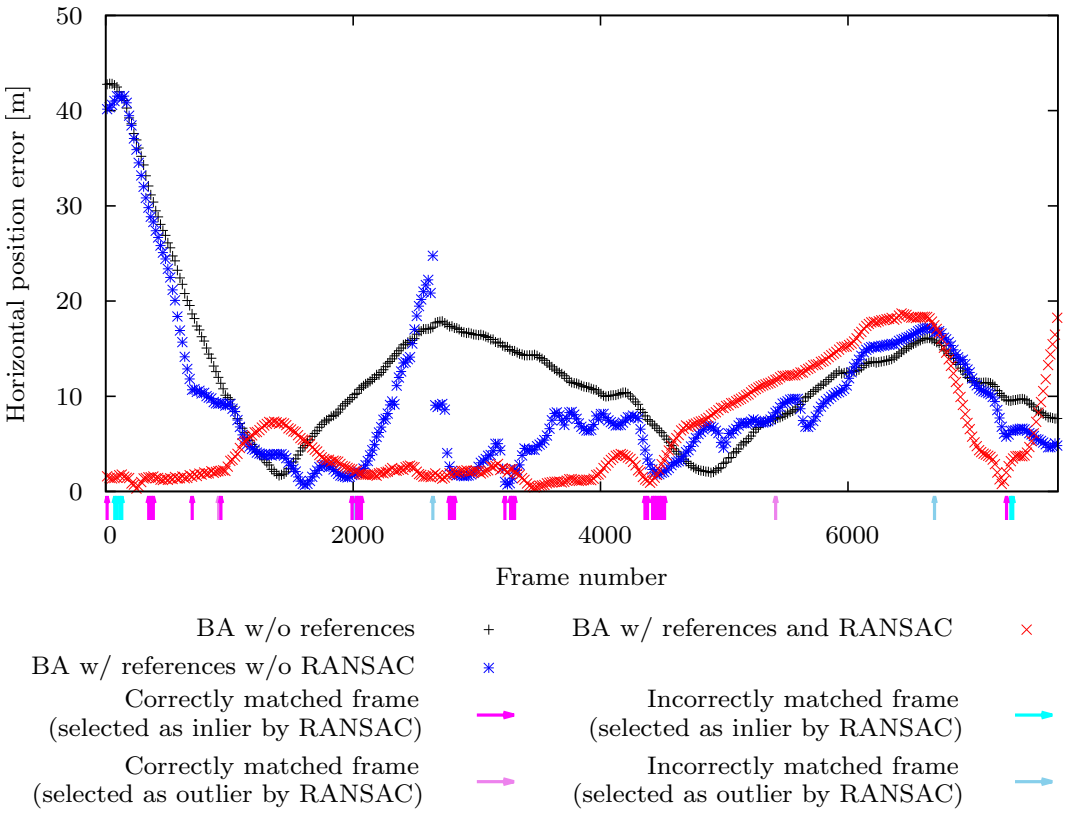


Figure 3.21: Horizontal position error in each frame (experiment 2).

for each frame, respectively. Although frames with incorrect matches still remained even when using a two-stage RANSAC, the proposed BA with RANSAC reduced the accumulative errors. However, at around the 6,000th frame, the accumulative errors are still large because there were only a small number of matches.

3.5. Conclusions

In this chapter, we proposed a method for removing accumulative errors in SfM using aerial images as external references that already exist for many places around the world. To this end, we proposed BA that uses feature matches between the ground-view and aerial images. To find correct matches from unreliable matches, we introduced new RANSAC schemes to both the feature matching and

bundle adjustment stages. In the experiments, we confirmed that the proposed method is effective for estimating the camera poses of real video sequences taken in outdoor environments. However, the accumulative errors still remain when there are no available matches during a long period of time. To find matches where the ground is not level, affine and/or perspective invariant features such as ASIFT [123] and Ferns [124] can be used with homography as a geometric transformation in RANSAC.

Chapter 4

Online Camera Pose Estimation using 3D Point Database Created from Structure-from-Motion

4.1. Introduction

Some robot navigation and augmented reality applications require estimating the camera poses along a previously taken route. In this chapter, we propose an online camera pose estimation method for such applications using a 3D point database created using SfM from previously captured images. Unlike the methods based on SfM described in Chapters 2 and 3, the method proposed in this chapter estimates the camera poses directly from a database. Errors are therefore not accumulated during the online camera pose estimation. Although the database created by SfM is affected by accumulative errors, the methods that reduce accumulative errors, e.g., loop closing, can be applied during the offline stage.

As mentioned in Section 1.2.5, many methods based on a 3D point database have been previously proposed [99–106]. The main problem of methods using a 3D point database is how to obtain matches between the input images and the database. To limit the search space, existing methods track the feature points temporally [104–106]. However, tracking sometimes fails owing to occlusions and rapid movement of the cameras. On the other hand, methods using an image database, as described in Section 1.2.3, can efficiently identify the database image

that is the most similar to the current image by considering the spatio-temporal connections between the database and the input images. By combining these two approaches, we propose a method utilizing a 3D point database that employs an image-database method to limit the search space. For a state-of-the-art image-database method, we employ topometric-localization [88], which considers topological information such as the spatio-temporal connections between sequential images at the metric scale.

As an application, we focus in particular on vehicle navigation for autonomous driving. The proposed method estimates the camera pose related to the vehicle pose by assuming that the transformation between the camera coordinate system and the vehicle coordinate system is known. As shown in Figure 4.1, the proposed method consists of two stages:

Offline creation of 3D point database: A 3D point database is created from images that are captured when a vehicle drives along a route for the first time. This database consists of images, 3D positions of the feature points estimated by SfM [43], and a topological graph for topometric localization [88].

Online localization: The vehicle is localized using a 3D point database and the current image through the following three steps. First, we identify the database image that is the most similar to the current image using topometric localization [88]. Next, we estimate the 2D-2D correspondences of the feature points between the current image and the identified database image to obtain 3D-2D correspondences of the feature points for the current image. The vehicle pose is finally estimated from these 3D-2D correspondences by solving the PnP problem.

4.2. Offline Creation of 3D Point Database

In our method, the database is created from images captured when the vehicle drives along a route for the first time. The database consists of images, the 3D positions of the feature points estimated using SfM, and a topological graph for topometric localization.

Structure-from-motion: We use VisualSFM [43], which is a state-of-the-art SfM implementation, to obtain the 3D positions of the feature points and camera

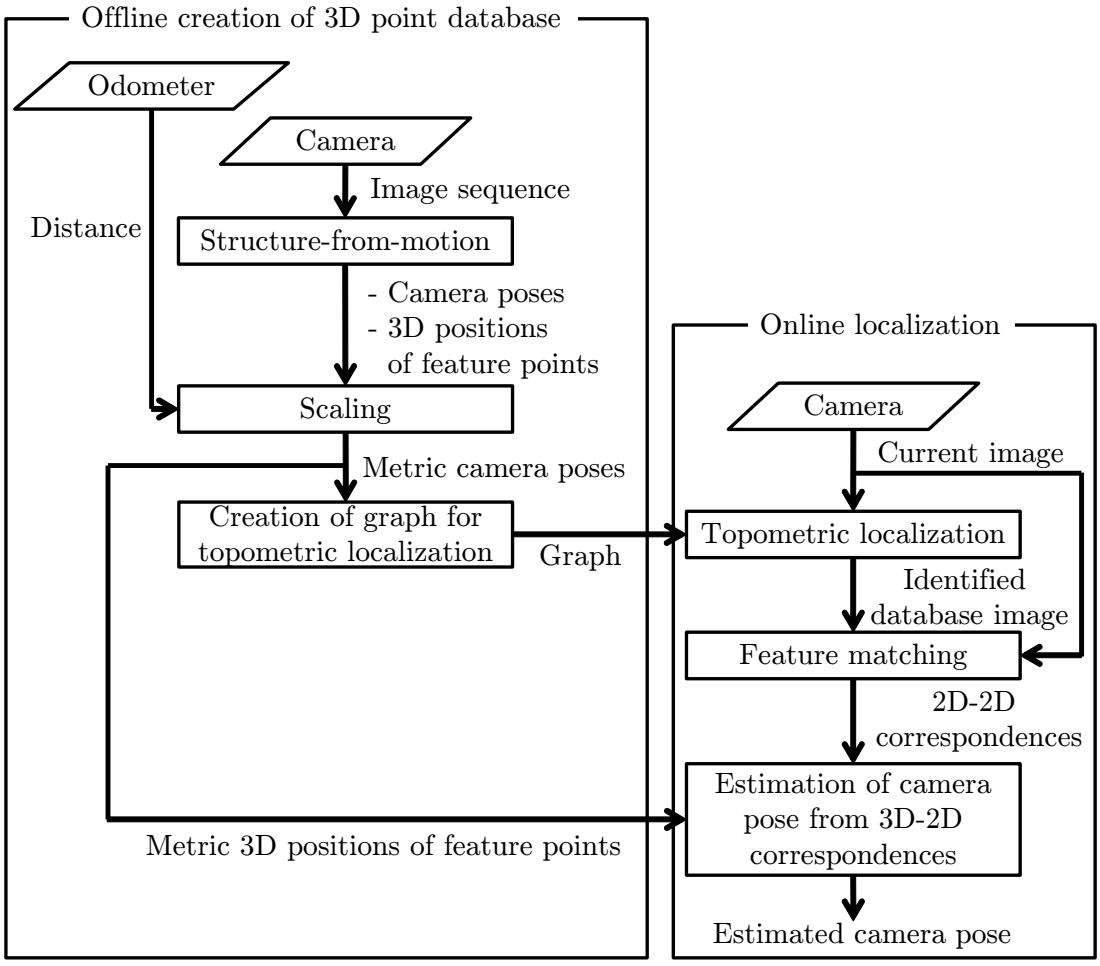


Figure 4.1: Flow of the proposed method using a 3D point database created using SfM.

poses. Since SfM cannot estimate the camera pose at the metric scale, which is necessary to estimate the vehicle pose at the metric scale for online localization, we measure the total driving distance using an odometer, which is standard equipment in ordinary vehicles. Specifically, the scale is determined by adjusting the total driving distance obtained using SfM based on the distance obtained from the odometer. In this way, metric 3D positions of the feature points and camera poses are obtained.

Creation of graph for topometric localization: We prepare a graph for topometric localization, which is used in the online process to identify the database image that is the most similar to the current image. The graph is created as described in [88], except that the camera positions are estimated using SfM instead of GPS. The edges of graph represent spatio-temporal connections of the images, and the nodes represent the images with their SURF features [119].

4.3. Online Camera Pose Estimation

The vehicle is localized using a 3D point database and the current image through the following three steps.

Topometric localization: We first identify the database image that is the most similar to the current image using topometric localization [88], which considers both the image features and spatio-temporal connections between sequential images represented as the graph.

Feature matching: We next estimate the 2D-2D correspondences of the feature points between the current image and the identified database image to obtain the 3D-2D correspondences of the feature points for the current image. We use SiftGPU [125], a GPU implementation of the SIFT feature, to achieve real-time processing.

Estimation of camera pose from 3D-2D correspondences: In the feature matching stage, 2D-2D correspondences between the current image and the identified database image are obtained. We also have the 3D positions of the feature points for the database image estimated during the offline stage. By combining them, we can obtain the 3D-2D correspondences of the feature points for the cur-

Figure 4.2: Evaluation vehicle.

rent image. We then estimate the camera pose from these 3D-2D correspondences by solving the PnP problem. To do so, we simply use a solver implemented in OpenCV [126], which non-linearly minimizes the reprojection errors. We also apply RANSAC to reject the incorrect feature matches. If the number of inlier matches is smaller than the threshold, the estimated camera pose is ignored as a failure. In our experiment, the threshold was set to 6, which is the minimum number needed to linearly solve the PnP problem.

4.4. Experiments

To test the effectiveness of the proposed method, we evaluated its accuracy quantitatively using image sequences captured in an indoor parking lot.

4.4.1 Experimental Setup

To evaluate our proposed method, we used a vehicle equipped with a sensor suite. As shown in Figure 4.2, a camera was mounted on the roof of the vehicle, and was oriented approximately 45 [°] to the right of the straightforward direction, and configured to acquire 1,024 [pixel] × 768 [pixel] images. The vehicle was able to output its driving distance like an ordinary vehicle.

Two image sequences were captured in an indoor parking lot on different days

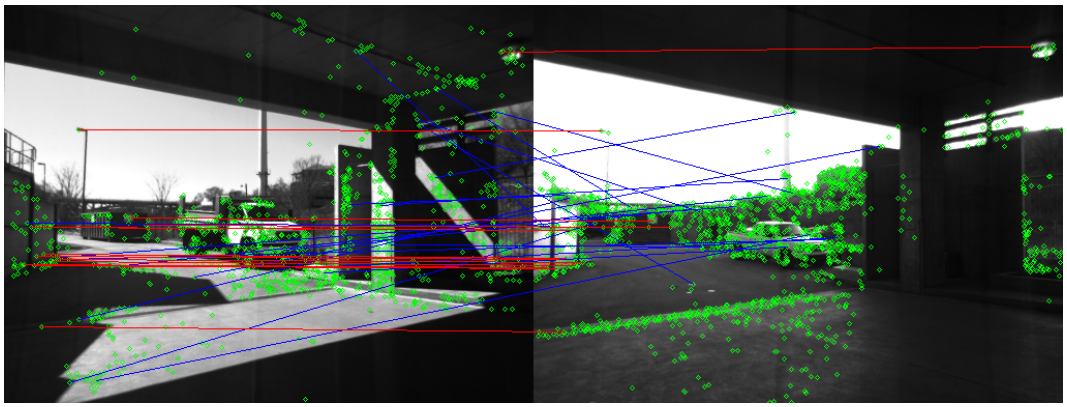
by manually driving along a route that makes a loop between the entrance and a parking space. One of the sequences was used for creating the database, and the other was used as input for online localization. The reference poses used to evaluate the accuracy of the estimated poses were estimated using an offline SfM [43]. This offline SfM used the feature matches among all of the images and took a long time to achieve an accurate estimation.

4.4.2 Quantitative Evaluation

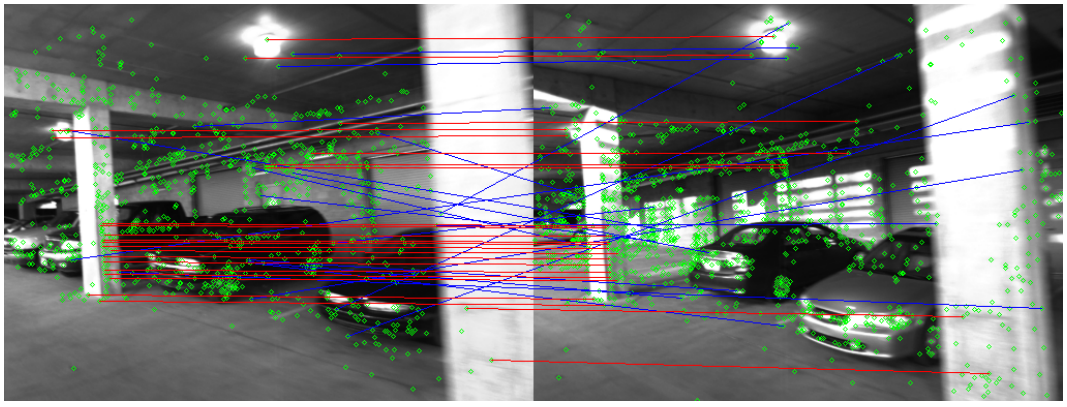
Figure 4.3 shows examples of the input images, the database images identified through topometric localization, and the results of feature matching between these images. Figure 4.4 shows the vehicle poses and 3D positions of the feature points for the database images, the vehicle poses estimated using the proposed method, and the reference vehicle poses. Except for the last quarter, the estimated vehicle poses were almost the same as the reference vehicle poses along the entire route, despite changes in the illumination (Figure 4.3(b)) and the environment (Figure 4.3(c)). Figure 4.5 shows histograms of the errors by comparing the estimated vehicle poses with the reference vehicle poses. Table 4.1 shows the average computation time of the proposed method, which was obtained using a PC with a 3.40 [GHz] Intel Core i7-2600k CPU and an NVIDIA GeForce GTX 580 GPU. From these results, it was confirmed that the proposed method estimated the vehicle pose at 8 [Hz] within a position error of 0.1 [m] and a posture error of 0.3 [°] in approximately 70% of the input images.

We investigated the causes of the errors for frames whose position errors are larger than 0.1 [m] or posture errors are larger than 0.3 [°]. Approximately 51% (54/104) of the frames were affected by the feature matches of the vehicle parked in different places in the database and the input images (Figure 4.3(d)). The other frames can be affected by the errors of SfM during the offline database creation. We also investigated the causes of the failures. Approximately 24% (19/78) of the frames were affected by the errors in the topometric localization (Figure 4.3(e)), and the other frames were affected by significant changes in the illumination and the environment (Figure 4.3(f)). Although the proposed method occasionally provides unstable estimates and failures owing to changes in the environment, it can be used in autonomous vehicles by combining sensors providing relative

(a)

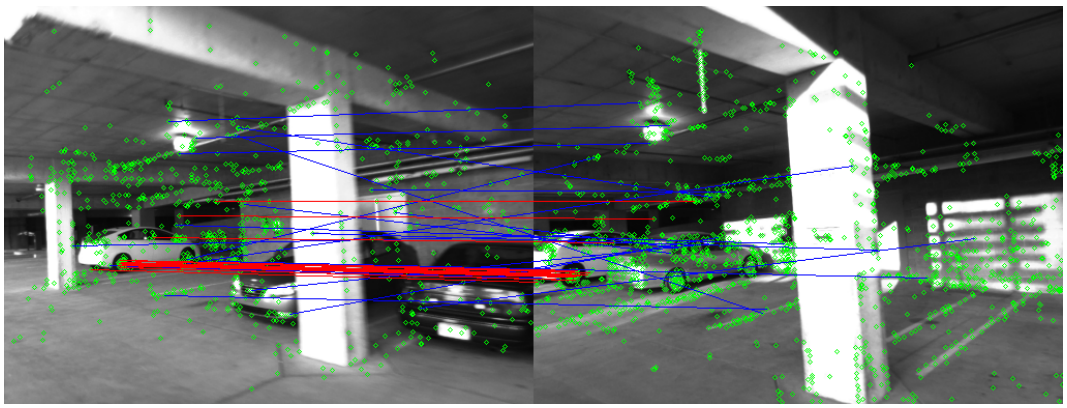


(b)

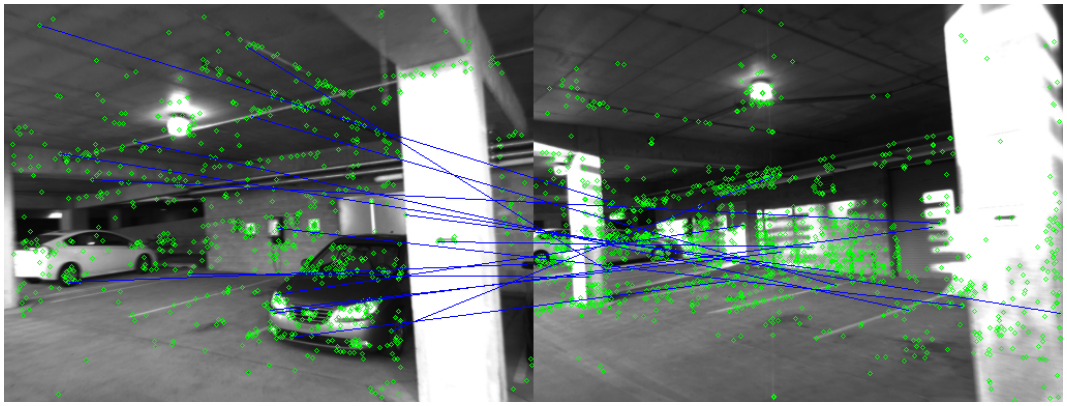


(c)

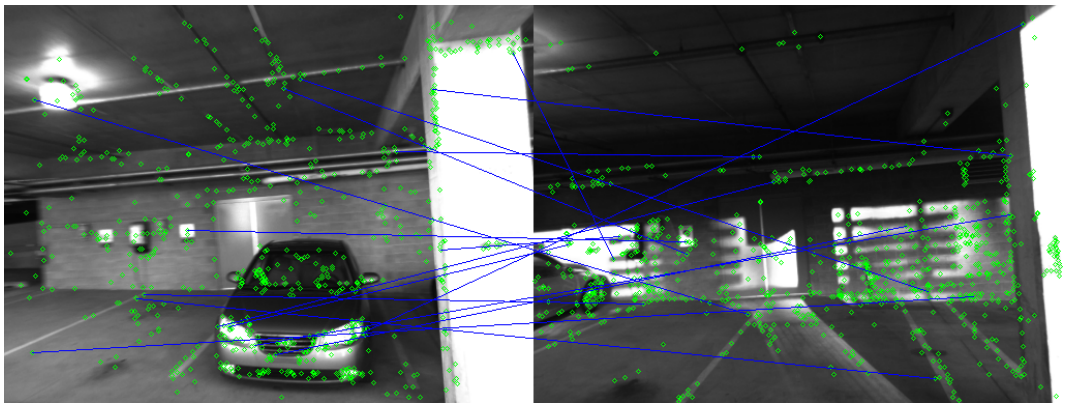
Figure 4.3: Examples of input images (right), the database images identified through topometric localization (left), and the results of feature matching between these images (red line, inlier; blue line, outlier). (1/2)



(d)



(e)



(f)

Figure 4.3. Continued. (2/2)

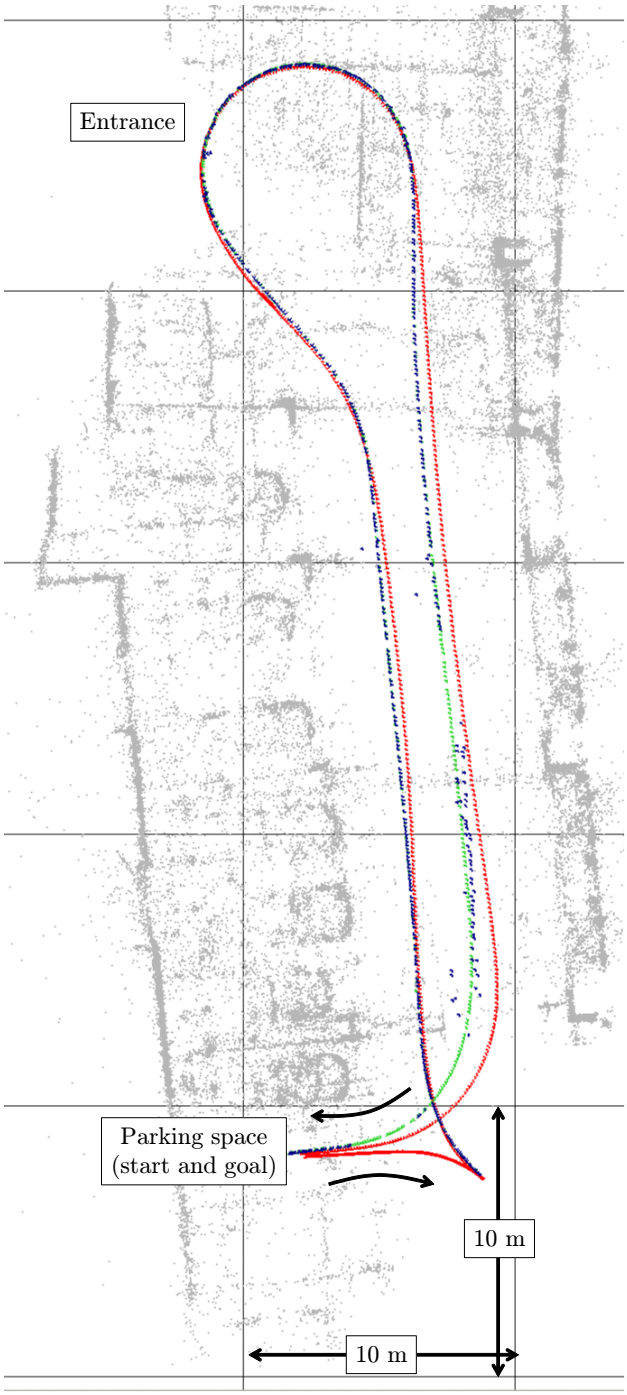
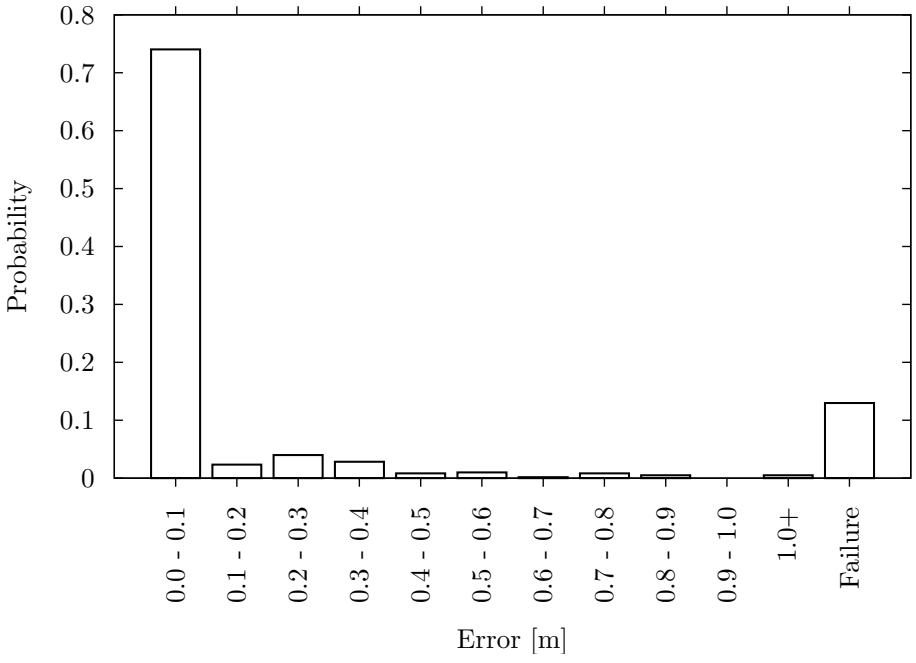
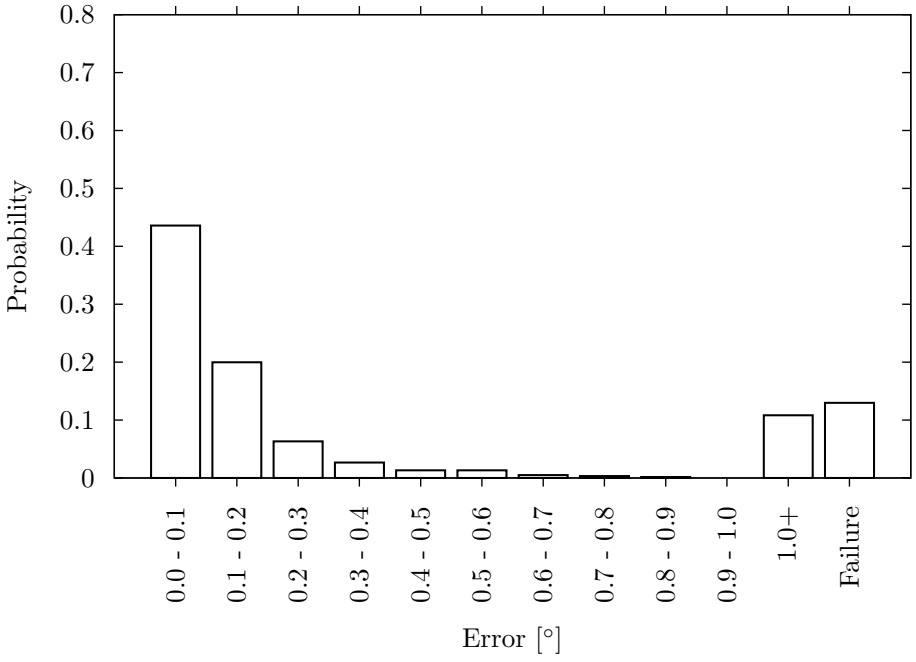


Figure 4.4: The vehicle poses (red) and 3D positions of the feature points (gray) for the database images, the vehicle poses estimated by the proposed method (blue), and the reference vehicle poses (green).



(a) Position



(b) Posture

Figure 4.5: Histograms of the errors.

Table 4.1: Computation time of the proposed method [ms].

Topometric localization	Feature matching	Solving PnP problem	Total
8.8	47.8	67.6	124.2

measurements such as odometry and IMU.

4.5. Conclusions

In this chapter, we proposed a method for localizing a vehicle along a previously driven route using a 3D point database created in advance. The proposed method identifies the database image that is the most similar to the current image through topometric localization, and estimates the vehicle poses from the 3D-2D correspondences of the feature points between the database and the current image. Our experiment showed that the method can estimate a vehicle pose within a position error of 0.1 [m] and a posture error of 0.3 [°] in approximately 70% of the input images captured in an indoor parking lot. However, the proposed method sometimes provides unstable estimates and failures owing to changes in the environment. To overcome this problem and achieve an autonomous vehicle, sensors such as odometry and IMU should be combined with the proposed method.

Chapter 5

Conclusions

5.1. Summary

This thesis proposed methods for estimating camera poses without accumulative errors by utilizing external references. For situations in which camera poses should be estimated without a pre-measurement of the target environments, we employed GPS and aerial images as external references by focusing on their availability in outdoor scenes. For situations in which cameras iteratively pass along the same route, we employed a 3D point database, which can be created by SfM, as an external reference.

For the method using GPS, we proposed extended BA by considering the GPS positioning confidence to achieve an accurate estimation even when the GPS positioning confidence is low. For the RTK-GPS, we introduced weighting coefficients depending on the solution types (RTK-fix or RTK-float). We also introduced parameter fitting to avoid the local minima in the extended BA after a long GPS outage. We confirmed experimentally that the proposed method can obtain more accurate camera positions than an existing extended-BA method that does not consider the GPS positioning confidence. However, the accumulative errors are still large during a long GPS outage.

For the method using aerial images, we proposed BA using feature matches between ground-view images and an aerial image. To the best of our knowledge, ours is the first method that uses aerial images as external references in BA. To this end, we proposed a robust feature matching method using RANSAC for both

the feature matching and BA stages. We confirmed experimentally that two-stage RANSAC selects correct matches, and BA using feature matches achieves a more accurate estimation than ordinary BA. However, accumulative errors still remain when there are no available matches for a long period of time.

For the method using a 3D point database, to efficiently limit the search space of the feature matches, we employed the topometric localization [88] that can efficiently identify the database image most similar to the current image. We confirmed experimentally that the proposed method can estimate the camera pose at 8 [Hz] within a position error of 0.1 [m] and a posture error of 0.3 [°] in approximately 70% of the input images using a PC with a 3.40 [GHz] Intel Core i7-2600k CPU and an NVIDIA GeForce GTX 580 GPU. However, the proposed method provides some unstable estimates and failures owing to changes in the environment.

5.2. Future Directions

In this thesis, we proposed three methods employing three types of external references to estimate camera poses without accumulative errors. As mentioned in Section 1.2, other types of external references such as community photos from the Internet and road maps are also available for certain environments. Therefore, to reduce accumulative errors in various environments and improve the robustness of the estimation, combining various types of external references depending on their availability and confidence is important. If more than one type of external reference is available, the confidence of the external references can be mutually estimated.

To more accurately estimate a camera pose, improving the quality of each external reference is also important. Although the resolution of aerial images is increasing, such images are still insufficient for certain areas. In addition, a number of moving objects, e.g., cars and pedestrians, hide the ground in road environments. Since the proposed method using aerial images can robustly match ground-view images with aerial images, it is possible to increase the quality of an aerial image using high-resolution ground-view images without moving objects. This is useful not only for estimating camera poses but also for ordinary map

applications such as Google Maps. For a 3D point database, it is possible to iteratively update the database using images for online camera pose estimation.

Acknowledgements

The research described in this thesis was carried out during my Ph.D. course at the Graduate School of Information Science, Nara Institute of Science and Technology.

First, I would like to express my deepest gratitude to my supervisor and the chair of my thesis committee, Professor Naokazu Yokoya of the Graduate School of Information Science, Nara Institute of Science and Technology, for his patient instruction, encouragement, and valuable comments throughout this research. He also provided me with a number of opportunities and invaluable experiences.

I would also like to express my appreciation to Professor Hirokazu Kato of the Graduate School of Information Science, Nara Institute of Science and Technology, who served on my thesis committee and provided many insightful suggestions and critical comments regarding this thesis.

In addition, I am deeply grateful to Associate Professor Tomokazu Sato of the Graduate School of Information Science, Nara Institute of Science and Technology, who also served on my thesis committee. Without his continuous encouragement, countless lessons, and constructive advice, I would never have completed this thesis.

I would particularly like to thank Professor Takeo Kanade of the Robotics Institute, Carnegie Mellon University, who suggested the research direction during my stay there. The precious lessons he provided me beyond the research have guided and supported every aspect of my life.

Regarding the research on camera pose estimation using GPS, my sincere appreciation goes to Dr. Nobuo Kochi and Mr. Tetsuji Anai of the R&D Center, Topcon Corporation. This research is a joint work with them, and their innovative suggestions guided this research to its successful completion.

I would also like to thank all the past and present members of the Vision and Media Computing Laboratory at the Graduate School of Information Science, Nara Institute of Science and Technology. They offered warm encouragement and friendship that gave me strength through difficult times. Professor Kazumasa Yamazawa of the Faculty of Information Engineering, Fukuoka Institute of Technology, Associate Professor Masayuki Kanbara, Assistant Professor Norihiko Kawai, Assistant Professor Takafumi Taketomi, and Assistant Professor Yuta Nakashima of the Graduate School of Information Science, Nara Institute of Science and Technology, provided helpful comments and invaluable discussions. Secretaries Ms. Mio Takahashi, Ms. Mina Nakamura, Ms. Chika Kijima, and Ms. Yumi Ishitani of the Vision and Media Computing Laboratory gave me administrative support.

I also benefited greatly by meeting and working with a number of people during my stay at the Robotics Institute, Carnegie Mellon University. I would particularly like to thank Mr. Arne Suppé with whom I have cooperatively completed papers involving this thesis. I also thank Ms. Yukiko Kano who helped me as an administrative staff member.

Finally, I wish to thank my family for their love and support.

References

[1] G. Klein and D. Murray, "Parallel Tracking and Mapping for Small AR Workspaces," Proc. IEEE and ACM Int. Symp. on Mixed and Augmented Reality, pp.225–234, 2007.

[2] Z. Li, Y. Wang, J. Guo, L.-F. Cheong, and S.Z. Zhou, "Diminished Reality using Appearance and 3D Geometry of Internet Photo Collections," Proc. IEEE Int. Symp. on Mixed and Augmented Reality, pp.11–19, 2013.

[3] A.J. Davison, I.D. Reid, N.D. Molton, and O. Stasse, "MonoSLAM: Real-Time Single Camera SLAM," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.29, no.6, pp.1052–1067, 2007.

[4] F. Okura, M. Kanbara, and N. Yokoya, "Fly-Through Heijo Palace Site: Augmented Telepresence using Aerial Omnidirectional Videos," Proc. ACM SIGGRAPH Posters, p.78, 2011.

[5] T. Sato, M. Kanbara, N. Yokoya, and H. Takemura, "Dense 3-D Reconstruction of an Outdoor Scene by Hundreds-Baseline Stereo using a Hand-Held Video Camera," Int. J. of Computer Vision, vol.47, no.1-3, pp.119–129, 2002.

[6] S. Agarwal, N. Snavely, I. Simon, S.M. Seitz, and R. Szeliski, "Building Rome in a Day," Proc. IEEE Int. Conf. on Computer Vision, pp.72–79, 2009.

[7] M. Jancosek and T. Pajdla, "Multi-View Reconstruction Preserving Weakly-Supported Surfaces," Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp.3121–3128, 2011.

[8] S. Knorr, M. Kunter, and T. Sikora, "Super-Resolution Stereo- and Multi-View Synthesis from Monocular Video Sequences," Proc. Int. Conf. on 3-D Digital Imaging and Modeling, pp.55–64, 2007.

[9] G. Chaurasia, S. Duchene, O. Sorkine-Hornung, and G. Drettakis, "Depth Synthesis and Local Warps for Plausible Image-Based Navigation," ACM Trans. on Graphics, vol.32, no.3, pp.30:1–30:12, 2013.

[10] Y. Awatsu, N. Kawai, T. Sato, and N. Yokoya, "Spatio-Temporal Super-Resolution using Depth Map," Proc. Scandinavian Conf. on Image Analysis, pp.696–705, 2009.

[11] F. Liu, M. Gleicher, H. Jin, and A. Agarwala, "Content-Preserving Warps for 3D Video Stabilization," ACM Trans. on Graphics, vol.28, no.3, pp.44:1–44:10, 2009.

[12] R.I. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, 2004.

[13] R. Szeliski, Computer Vision: Algorithms and Applications, Springer, 2010.

[14] A. Torii, T. Okatani, and S. Nobuhara, "Recent Research Trends in Multi-View Three-Dimensional Reconstruction," IPSJ SIG Technical Report, vol.CVIM-176, no.1, pp.1–22, 2011 (in Japanese).

[15] H.C. Longuet-Higgins, "A Computer Algorithm for Reconstructing a Scene from Two Projections," Nature, vol.293, pp.133–135, 1981.

[16] R.I. Hartley, "In Defense of the Eight-Point Algorithm," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.19, no.6, pp.580–593, 1997.

[17] R.I. Hartley, "Projective Reconstruction and Invariants from Multiple Images," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.16, no.10, pp.1036–1041, 1994.

[18] D. Nistér, "An Efficient Solution to the Five-Point Relative Pose Problem," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.26, no.6, pp.756–770, 2004.

[19] H. Li and R. Hartley, "Five-Point Motion Estimation Made Easy," Proc. IAPR Int. Conf. on Pattern Recognition, pp.630–633, 2006.

[20] Z. Kukelova, M. Bujnak, and T. Pajdla, "Polynomial Eigenvalue Solutions to Minimal Problems in Computer Vision," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.34, no.7, pp.1381–1393, 2012.

[21] V. Lui and T. Drummond, "An Iterative 5-pt Algorithm for Fast and Robust Essential Matrix Estimation," Proc. British Machine Vision Conference, pp.127.1–127.11, 2013.

[22] H. Stewénius, D. Nistér, F. Kahl, and F. Schaffalitzky, "A Minimal Solution for Relative Pose with Unknown Focal Length," Image and Vision Computing, vol.26, no.7, pp.871–877, 2008.

[23] C. Tomasi and T. Kanade, "Shape and Motion from Image Streams under Orthography: A Factorization Method," Int. J. of Computer Vision, vol.9, no.2, pp.137–154, 1992.

[24] C.J. Poelman and T. Kanade, "A Paraperspective Factorization Method for Shape and Motion Recovery," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.19, no.3, pp.206–218, 1997.

[25] P. Sturm and B. Triggs, "A Factorization Based Algorithm for Multi-Image Projective Structure and Motion," Proc. European Conf. on Computer Vision, pp.709–720, 1996.

[26] S. Christy and R. Horaud, "Euclidean Shape and Motion from Multiple Perspective Views by Affine Iterations," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.18, no.11, pp.1098–1104, 1996.

[27] T. Okatani, T. Yoshida, and K. Deguchi, "Efficient Algorithm for Low-Rank Matrix Factorization with Missing Components and Performance Comparison of Latest Algorithms," Proc. IEEE Int. Conf. on Computer Vision, pp.842–849, 2011.

[28] Y. Dai, H. Li, and M. He, "Projective Multiview Structure and Motion from Element-Wise Factorization," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.35, no.9, pp.2238–2251, 2013.

[29] J. Civera, O.G. Grasa, A.J. Davison, and J.M.M. Montiel, "1-Point RANSAC for Extended Kalman Filtering : Application to Real-Time Structure from Motion and Visual Odometry," J. of Field Robotics, vol.27, no.5, pp.609–631, 2010.

[30] E. Eade and T. Drummond, "Scalable Monocular SLAM," Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp.469–476, 2006.

[31] E. Eade and T. Drummond, "Monocular SLAM as a Graph of Coalesced Observations," Proc. IEEE Int. Conf. on Computer Vision, 8 pages, 2007.

[32] H. Strasdat, J.M.M. Montiel, and A.J. Davison, "Visual SLAM: Why Filter?," Image and Vision Computing, vol.30, no.2, pp.65–77, 2012.

[33] D. Nistér, O. Naroditsky, and J. Bergen, "Visual Odometry," Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp.652–659, 2004.

[34] Z. Zhang and Y. Shan, "Incremental Motion Estimation through Modified Bundle Adjustment," Proc. IEEE Int. Conf. on Image Processing, pp.343–346, 2003.

[35] C. Engels, H. Stewénius, and D. Nistér, "Bundle Adjustment Rules," Proc. Photogrammetric Computer Vision, pp.266–271, 2006.

[36] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd, "Generic and Real-Time Structure from Motion using Local Bundle Adjustment," Image and Vision Computing, vol.27, no.8, pp.1178–1193, 2009.

[37] S.A. Holmes and D.W. Murray, "Monocular SLAM with Conditionally Independent Split Mapping," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.35, no.6, pp.1451–1463, 2013.

[38] A. Kundu, K.M. Krishna, and C.V. Jawahar, "Realtime Multibody Visual SLAM with a Smoothly Moving Monocular Camera," Proc. IEEE Int. Conf. on Computer Vision, pp.2080–2087, 2011.

[39] W. Tan, H. Liu, Z. Dong, G. Zhang, and H. Bao, "Robust Monocular SLAM in Dynamic Environments," Proc. IEEE Int. Symp. on Mixed and Augmented Reality, pp.209–218, 2013.

[40] J. Hedborg, P.-E. Forssén, M. Felsberg, and E. Ringaby, "Rolling Shutter Bundle Adjustment," Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp.1434–1441, 2012.

[41] N. Snavely, S.M. Seitz, and R. Szeliski, "Modeling the World from Internet Photo Collections," Int. J. of Computer Vision, vol.80, no.2, pp.189–210, 2008.

[42] N. Snavely, S.M. Seitz, and R. Szeliski, "Skeletal Graphs for Efficient Structure from Motion," Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 8 pages, 2008.

[43] C. Wu, "VisualSFM: A Visual Structure from Motion System," `http://ccwu.me/vsfm/`, 2013.

[44] B. Triggs, P. Mclauchlan, R. Hartley, and A. Fitzgibbon, "Bundle Adjustment - A Modern Synthesis," Proc. Int. Workshop on Vision Algorithms, pp.298–372, 1999.

[45] Y. Jeong, D. Nistér, D. Steedly, R. Szeliski, and I.-S. Kweon, "Pushing the Envelope of Modern Methods for Bundle Adjustment," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.34, no.8, pp.1605–1617, 2012.

[46] Y.-D. Jian, D.C. Balcan, and F. Dellaert, "Generalized Subgraph Preconditioners for Large-Scale Bundle Adjustment," Proc. IEEE Int. Conf. on Computer Vision, pp.295–302, 2011.

[47] A. Kushal and S. Agarwal, "Visibility Based Preconditioning for Bundle Adjustment," Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp.1442–1449, 2012.

[48] Z. Dai, F. Zhang, and H. Wang, "Robust Maximum Likelihood Estimation by Sparse Bundle Adjustment using the L1 Norm," Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp.1672–1679, 2012.

[49] M.I.A. Lourakis and A.A. Argyros, "SBA: A Software Package for Generic Sparse Bundle Adjustment," ACM Trans. on Mathematical Software, vol.36, no.1, pp.2:1–2:30, 2009.

[50] M.I.A. Lourakis, "Sparse Non-Linear Least Squares Optimization for Geometric Vision," Proc. European Conf. on Computer Vision, pp.43–56, 2010.

[51] C. Wu, S. Agarwal, B. Curless, and S.M. Seitz, "Multicore Bundle Adjustment," Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp.3057–3064, 2011.

[52] S. Agarwal, K. Mierle, and Others, "Ceres Solver," `https://code.google.com/p/ceres-solver/`, 2013.

[53] A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer, "Fast and Incremental Method for Loop-Closure Detection using Bags of Visual Words," IEEE Trans. on Robotics, vol.24, no.5, pp.1027–1037, 2008.

[54] H. Strasdat, A.J. Davison, J.M.M. Montiel, and K. Konolige, "Double Window Optimisation for Constant Time Visual SLAM," Proc. IEEE Int. Conf. on Computer Vision, pp.2352–2359, 2011.

[55] B. Williams, M. Cummins, J. Neira, P. Newman, I. Reid, and J. Tardós, "A Comparison of Loop Closing Techniques in Monocular SLAM," Robotics and Autonomous Systems, vol.57, no.12, pp.1188–1197, 2009.

[56] A. Cohen, C. Zach, S.N. Sinha, and M. Pollefeys, "Discovering and Exploiting 3D Symmetries in Structure from Motion," Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp.1514–1521, 2012.

[57] A. Eudes, S. Naudet-Collette, M. Lhuillier, and M. Dhome, "Weighted Local Bundle Adjustment and Application to Odometry and Visual SLAM Fusion," Proc. British Machine Vision Conference, pp.25.1–25.10, 2010.

[58] J. Michot, A. Bartoli, and F. Gaspard, "Bi-Objective Bundle Adjustment with Application to Multi-Sensor SLAM," Proc. Int. Symp. on 3D Data Processing, Visualization and Transmission, 8 pages, 2010.

[59] R. Tenmoku, M. Kanbara, and N. Yokoya, "A Wearable Augmented Reality System using Positioning Infrastructures and a Pedometer," Proc. IEEE Int. Symp. on Wearable Computers, pp.110–117, 2003.

[60] W. Piekarski, R. Smith, and B.H. Thomas, "Designing Backpacks for High Fidelity Mobile Outdoor Augmented Reality," Proc. IEEE and ACM Int. Symp. on Mixed and Augmented Reality, pp.280–281, 2004.

[61] M. Kourogi, N. Sakata, T. Okuma, and T. Kurata, "Indoor/Outdoor Pedestrian Navigation with an Embedded GPS/RFID/Self-Contained Sensor System," Proc. Int. Conf. on Artificial Reality and Telexistence, pp.1310–1321, 2006.

[62] M. Ramachandran, A. Veeraraghavan, and R. Chellappa, "A Fast Bilinear Structure from Motion Algorithm using a Video Sequence and Inertial Sensors," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.33, no.1, pp.186–193, 2011.

[63] R. Carceroni, A. Kumar, and K. Daniilidis, "Structure from Motion with Known Camera Positions," Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp.477–484, 2006.

[64] M. Pollefeys, D. Nistér, J.-M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S.-J. Kim, P. Merrell, C. Salmi, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewénius, R. Yang, G. Welch, and H. Towles, "Detailed Real-Time Urban 3D Reconstruction from Video," Int. J. of Computer Vision, vol.78, no.2-3, pp.143–167, 2008.

[65] Y. Bok, Y. Jeong, D.-G. Choi, and I.S. Kweon, "Capturing Village-Level Heritages with a Hand-Held Camera-Laser Fusion Sensor," Int. J. of Computer Vision, vol.94, no.1, pp.36–53, 2011.

[66] L. Wei, C. Cappelle, Y. Ruichek, and F. Zann, "GPS and Stereovision-Based Visual Odometry: Application to Urban Scene Mapping and Intelligent Vehicle Localization," Int. J. of Vehicular Technology, vol.2011, 17 pages, 2011.

[67] M. Agrawal and K. Konolige, "Real-Time Localization in Outdoor Environments using Stereo Vision and Inexpensive GPS," Proc. IAPR Int. Conf. on Pattern Recognition, pp.1063–1068, 2006.

[68] D. Schleicher, L.M. Bergasa, M. Ocaña, R. Barea, and M.E. López, "Real-Time Hierarchical Outdoor SLAM Based on Stereovision and GPS Fusion," IEEE Trans. on Intelligent Transportation Systems, vol.10, no.3, pp.440–452, 2009.

[69] D. Dusha and L. Mejias, "Error Analysis and Attitude Observability of a Monocular GPS/Visual Odometry Integrated Navigation Filter," Int. J. of Robotics Research, vol.31, no.6, pp.714–737, 2012.

[70] Y. Yokochi, S. Ikeda, T. Sato, and N. Yokoya, "Extrinsic Camera Parameter Estimation Based-on Feature Tracking and GPS Data," Proc. Asian Conf. on Computer Vision, pp.369–378, 2006.

[71] S. Ikeda, T. Sato, K. Yamaguchi, and N. Yokoya, "Construction of Feature Landmark Database using Omnidirectional Videos and GPS Positions," Proc. Int. Conf. on 3-D Digital Imaging and Modeling, pp.249–256, 2007.

[72] M. Lhuillier, "Incremental Fusion of Structure-from-Motion and GPS using Constrained Bundle Adjustments," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.34, no.12, pp.2489–2495, 2012.

[73] D. Larnaout, V. Gay-Bellile, S. Bourgeois, and M. Dhome, "Vehicle 6-DoF Localization Based on SLAM Constrained by GPS and Digital Elevation Model Information," Proc. IEEE Int. Conf. on Image Processing, pp.2504–2508, 2013.

[74] H. Kato and M. Billinghurst, "Marker Tracking and HMD Calibration for a Video-Based Augmented Reality Conferencing System," Proc. IEEE and ACM Int. Workshop on Augmented Reality, pp.85–94, 1999.

[75] Y. Nakazato, M. Kanbara, and N. Yokoya, "Localization System for Large Indoor Environments using Invisible Markers," Proc. ACM Symp. on Virtual Reality Software and Technology, pp.295–296, 2008.

[76] S. Saito, A. Hiyama, T. Tanikawa, and M. Hirose, "Indoor Marker-Based Localization using Coded Seamless Pattern for Interior Decoration," Proc. IEEE Virtual Reality, pp.67–74, 2007.

[77] R. Tenmoku, A. Nishigami, F. Shibata, A. Kimura, and H. Tamura, "Balancing Design Freedom and Constraints in Wall Posters Masquerading as AR Tracking Markers," Proc. Human-Computer Interaction International, pp.263–272, 2009.

[78] J. Wang, H. Zha, and R. Cipolla, "Coarse-to-Fine Vision-Based Localization by Indexing Scale-Invariant Features," IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics, vol.36, no.2, pp.413–422, 2006.

[79] D. Nistér and H. Stewénius, "Scalable Recognition with a Vocabulary Tree," Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp.2161–2168, 2006.

[80] C. Valgren and A.J. Lilienthal, "SIFT, SURF & Seasons: Appearance-Based Long-Term Localization in Outdoor Environments," Robotics and Autonomous Systems, vol.58, no.2, pp.149–156, 2010.

[81] A.C. Murillo, C. Sagüés, J.J. Guerrero, T. Goedemé, T. Tuytelaars, and L. Van Gool, "From Omnidirectional Images to Hierarchical Localization," Robotics and Autonomous Systems, vol.55, no.5, pp.372–382, 2007.

[82] G. Schindler, M. Brown, and R. Szeliski, "City-Scale Location Recognition," Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 7 pages, 2007.

[83] D.M. Chen, G. Baatz, K. Köser, S.S. Tsai, R. Vedantham, T. Pylvänäinen, K. Roimela, X. Chen, J. Bach, M. Pollefeys, B. Girod, and R. Grzeszczuk, "City-Scale Landmark Identification on Mobile Devices," Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp.737–744, 2011.

[84] J. Hays and A.A. Efros, "IM2GPS : Estimating Geographic Information from a Single Image," Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 8 pages, 2008.

[85] Y. Kalantidis, G. Tolias, Y. Avrithis, M. Phinikettos, E. Spyrou, P. Mylonas, and S. Kollias, "VIRaL: Visual Image Retrieval and Localization," Multimedia Tools and Applications, vol.51, no.2, pp.555–592, 2010.

[86] Y. Yagi, K. Imai, K. Tsuji, and M. Yachida, "Iconic Memory-Based Omnidirectional Route Panorama Navigation," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.27, no.1, pp.78–87, 2005.

[87] J. Košecká, F. Li, and X. Yang, "Global Localization and Relative Positioning Based on Scale-Invariant Keypoints," Robotics and Autonomous Systems, vol.52, no.1, pp.27–38, 2005.

[88] H. Badino, D. Huber, and T. Kanade, "Real-Time Topometric Localization," Proc. IEEE Int. Conf. on Robotics and Automation, pp.1635–1642, 2012.

[89] G. Vaca-Castano, A.R. Zamir, and M. Shah, "City Scale Geo-Spatial Trajectory Estimation of a Moving Camera," Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp.1186–1193, 2012.

[90] A.I. Comport, E. Marchand, M. Pressigout, and F. Chaumette, "Real-Time Markerless Tracking for Augmented Reality: The Virtual Visual Servoing Framework," IEEE Trans. on Visualization and Computer Graphics, vol.12, no.4, pp.615–628, 2006.

[91] G. Reitmayr and T.W. Drummond, "Going Out: Robust Model-Based Tracking for Outdoor Augmented Reality," Proc. IEEE and ACM Int. Symp. on Mixed and Augmented Reality, pp.109–118, 2006.

[92] C. Cappelle, M.E. El Najjar, F. Charpillet, and D. Pomorski, "Virtual 3D City Model for Navigation in Urban Areas," J. of Intelligent & Robotic Systems, vol.66, no.3, pp.377–399, 2012.

[93] G. Baatz, K. Köser, D. Chen, R. Grzeszczuk, and M. Pollefeys, "Leveraging 3D City Models for Rotation Invariant Place-of-Interest Recognition," Int. J. of Computer Vision, vol.96, no.3, pp.315–334, 2012.

[94] M.A. Oikawa, T. Taketomi, G. Yamamoto, M. Fujisawa, T. Amano, J. Miyazaki, and H. Kato, "A Model-Based Tracking Framework for Textureless 3D Rigid Curved Objects," SBC J. on 3D Interactive Systems, vol.3, no.2, pp.2–15, 2012.

[95] G. Bleser, H. Wuest, and D. Stricker, "Online Camera Pose Estimation in Partially Known and Dynamic Scenes," Proc. IEEE and ACM Int. Symp. on Mixed and Augmented Reality, pp.56–65, 2006.

[96] P. Lothe, S. Bourgeois, E. Royer, M. Dhome, and S. Naudet-Collette, "Real-Time Vehicle Global Localisation with a Single Camera in Dense Urban Areas: Exploitation of Coarse 3D City Models," Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp.863–870, 2010.

[97] D. Larnaout, S. Bourgeois, V. Gay-Bellile, and M. Dhome, "Towards Bundle Adjustment with GIS Constraints for Online Geo-Localization of a Vehicle in Urban Center," Proc. Int. Conf. on 3D Imaging, Modeling, Processing, Visualization and Transmission, pp.348–355, 2012.

[98] M. Tamaazousti, V. Gay-Bellile, S. Naudet Collette, S. Bourgeois, and M. Dhome, "NonLinear Refinement of Structure from Motion Reconstruction by Taking Advantage of a Partial Knowledge of the Environment," Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp.3073–3080, 2011.

[99] I. Skrypnyk and D.G. Lowe, "Scene Modelling, Recognition and Tracking with Invariant Image Features," Proc. IEEE and ACM Int. Symp. on Mixed and Augmented Reality, pp.110–119, 2004.

[100] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof, "From Structure-from-Motion Point Clouds to Fast Location Recognition," Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp.2599–2606, 2009.

[101] Y. Li, N. Snavely, D. Huttenlocher, and P. Fua, "Worldwide Pose Estimation using 3D Point Clouds," Proc. European Conf. on Computer Vision, pp.15–29, 2012.

[102] T. Sattler, B. Leibe, and L. Kobbelt, "Improving Image-Based Localization by Active Correspondence Search," Proc. European Conf. on Computer Vision, pp.752–765, 2012.

[103] C. Arth, D. Wagner, M. Klopschitz, A. Irschara, and D. Schmalstieg, "Wide Area Localization on Mobile Phones," Proc. IEEE and ACM Int. Symp. on Mixed and Augmented Reality, pp.73–82, 2009.

[104] F. Wientapper, H. Wuest, and A. Kuijper, "Composing the Feature Map Retrieval Process for Robust and Ready-to-Use Monocular Tracking," Computers & Graphics, vol.35, no.4, pp.778–788, 2011.

[105] T. Taketomi, T. Sato, and N. Yokoya, "Real-Time and Accurate Extrinsic Camera Parameter Estimation using Feature Landmark Database for Augmented Reality," Computers & Graphics, vol.35, no.4, pp.768–777, 2011.

[106] H. Lim, S.N. Sinha, M.F. Cohen, and M. Uyttendaele, "Real-Time Image-Based 6-DOF Localization in Large-Scale environments," Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp.1043–1050, 2012.

[107] T. Sato, S. Ikeda, and N. Yokoya, "Extrinsic Camera Parameter Recovery from Multiple Image Sequences Captured by an Omni-Directional Multi-Camera System," Proc. European Conf. on Computer Vision, pp.326–340, 2004.

[108] N. Fioraio and L.D. Stefano, "Joint Detection, Tracking and Mapping by Semantic Bundle Adjustment," Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp.1538–1545, 2013.

[109] M. Bansal, K. Daniilidis, and H. Sawhney, "Ultra-Wide Baseline Facade Matching for Geo-Localization," Proc. European Conf. on Computer Vision, pp.175–186, 2012.

[110] T.-Y. Lin, S. Belongie, and J. Hays, "Cross-View Image Geolocalization," Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp.891–898, 2013.

[111] S. Kim, S. DiVerdi, J.S. Chang, T. Kang, R. Iltis, and T. Höllerer, "Implicit 3D Modeling and Tracking for Anywhere Augmentation," Proc. ACM Symp. on Virtual Reality Software and Technology, pp.19–28, 2007.

[112] K.Y.K. Leung, C.M. Clark, and J.P. Huissoon, "Localization in Urban Environments by Matching Ground Level Video Images with an Aerial Image," Proc. IEEE Int. Conf. on Robotics and Automation, pp.551–556, 2008.

[113] H. Toriya, I. Kitahara, and Y. Ohta, "A Mobile Camera Localization Method using Aerial-View Images," Proc. IAPR Asian Conf. on Pattern Recognition, pp.49–53, 2013.

[114] M. Noda, T. Takahashi, D. Deguchi, I. Ide, H. Murase, Y. Kojima, and T. Naito, "Vehicle Ego-Localization by Matching In-Vehicle Camera Images to an Aerial Image," Proc. Computer Vision in Vehicle Technology: From Earth to Mars, 10 pages, 2010.

[115] O. Pink, F. Moosmann, and A. Bachmann, "Visual Features for Vehicle Localization and Ego-Motion Estimation," Proc. IEEE Intelligent Vehicles Symposium, pp.254–260, 2009.

[116] T.-J. Cham, A. Ciptadi, W.-C. Tan, M.-T. Pham, and L.-T. Chia, "Estimating Camera Pose from a Single Urban Ground-View Omnidirectional Image and a 2D Building Outline Map," Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp.366–373, 2010.

[117] S. Mills, "Relative Orientation and Scale for Improved Feature Matching," Proc. IEEE Int. Conf. on Image Processing, pp.3484–3488, 2013.

[118] D.G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," Int. J. of Computer Vision, vol.60, no.2, pp.91–110, 2004.

[119] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-Up Robust Features (SURF)," Computer Vision and Image Understanding, vol.110, no.3, pp.346–359, 2008.

[120] T. Sekii, T. Sato, H. Kume, and N. Yokoya, "6-DOF Camera Pose Estimation using Reference Points on an Aerial Image without Altitude Information," IPSJ Trans. on Computer Vision and Applications, vol.5, pp.134–142, 2013.

[121] M.A. Brubaker, A. Geiger, and R. Urtasun, "Lost! Leveraging the Crowd for Probabilistic Visual Self-Localization," Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp.3057–3064, 2013.

[122] M.A. Fischler and R.C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," Communications of the ACM, vol.24, no.6, pp.381–395, 1981.

[123] J.-M. Morel and G. Yu, "ASIFT: A New Framework for Fully Affine Invariant Image Comparison," SIAM J. on Imaging Sciences, vol.2, no.2, pp.438–469, 2009.

[124] M. Özuysal, M. Calonder, V. Lepetit, and P. Fua, "Fast Keypoint Recognition using Random Ferns," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.32, no.3, pp.448–461, 2010.

[125] C. Wu, "SiftGPU: A GPU Implementation of Scale Invaraint Feature Transform (SIFT)," `http://cs.unc.edu/~ccwu/siftgpu`, 2007.

[126] "OpenCV," `http://opencv.org/`.

List of Publications

Journal Papers

1. T. Sekii, T. Sato, H. Kume, and N. Yokoya, "6-DOF Camera Pose Estimation using Reference Points on an Aerial Image without Altitude Information," IPSJ Trans. on Computer Vision and Applications, vol.5, pp.134–142, 2013.

2. H. Kume, T. Anai, T. Sato, T. Taketomi, N. Kochi, and N. Yokoya, "Extrinsic Camera Parameter Estimation from Video Images Considering GPS Positioning Confidence," J. of Institute of Image Electronics Engineers of Japan, vol.43, no.1, pp.35–43, 2014 (in Japanese). (Chapter 2).

International Conferences

1. H. Kume, T. Taketomi, T. Sato and N. Yokoya, "Extrinsic Camera Parameter Estimation using Video Images and GPS Considering GPS Positioning Accuracy," Proc. IAPR Int. Conf. on Pattern Recognition, pp.3923–3926, 2010.

2. H. Kume, A. Suppé, and T. Kanade, "Vehicle Localization along a Previously Driven Route using an Image Database," Proc. IAPR Conf. on Machine Vision Applications, pp.177–180, 2013. (Chapter 4).

3. T. Kanatani, H. Kume, T. Taketomi, T. Sato, and N. Yokoya, "Detection of 3D Points on Moving Objects from Point Cloud Data for 3D Modeling

of Outdoor Environments," Proc. IEEE Int. Conf. on Image Processing, pp.2163–2167, 2013.

4. H. Kume, T. Sato and N. Yokoya, "Sampling Based Bundle Adjustment using Feature Matches Between Ground-View and Aerial Images," Proc. Int. Conf. on Computer Vision Theory and Applications, pp.692–698, 2014. (Chapter 3).

Domestic Conferences

1. H. Kume, T. Taketomi, T. Sato and N. Yokoya, "Camera Pose Estimation from an Image Sequence and GPS Data Considering GPS Positioning Confidence," Proc. JSPRS Autum Conference, pp.127–128, 2009 (in Japanese).

2. H. Kume, T. Taketomi, T. Sato and N. Yokoya, "Consideration of GPS Positioning Accuracy for Camera Parameter Estimation from GPS and Video Images," Proc. Symp. on Pattern Measurement, pp.33–38, 2009 (in Japanese).

3. H. Kume, T. Taketomi, T. Sato and N. Yokoya, "Extrinsic Camera Parameter Estimation using Image Sequence and GPS Considering GPS Positioning Accuracy," Proc. Meeting on Image Recognition and Understanding, pp.1198–1205, 2010 (in Japanese).

4. H. Kume, T. Taketomi, T. Sato and N. Yokoya, "Extrinsic Camera Parameter Estimation using Video Images and GPS Considering GPS Positioning Confidence and Outlier," IPSJ SIG Technical Report, vol.2010–CVIM–173, no.36, pp.1–8, 2010 (in Japanese).

5. Y. Babaguchi, H. Kume, T. Sato and N. Yokoya, "Cyclic-Motion Reproduction for Moving Object in Telepresence System," Proc. ITE Winter Annual Convention, no.1–1, 1 page, 2010 (in Japanese).

6. T. Kanatani, H. Kume, T. Taketomi, M. Kanbara, and N. Yokoya, "Determination of Moving Objects for 3-D Modeling of Outdoor Environments," Proc. IEICE General Conference, no.D–12–31, p.134, 2011 (in Japanese).

7. Y. Babaguchi, H. Kume, T. Sato and N. Yokoya, "Viewpoint Varying Video Textures Preserving Temporal and Spatial Continuity of Moving Objects," IEICE Technical Report, PRMU2010-262, pp.139–144, 2011 (in Japanese).

8. H. Kume, T. Taketomi, T. Sato and N. Yokoya, "Camera Pose Estimation from an Image Sequence and GPS Data by Bundle Adjustment Considering Position Continuity," Proc. Meeting on Image Recognition and Understanding, pp.825–830, 2011 (in Japanese).

9. T. Sekii, H. Kume, T. Sato and N. Yokoya, "Camera Pose Estimation using Reference Points on an Aerial Image without Altitude Information by Solving PnP Problem," Proc. Kansai-Section Joint Convention of Institutes of Electrical Engineering, no.30P2–33, 2 pages, 2011 (in Japanese).

10. T. Sekii, H. Kume, T. Sato and N. Yokoya, "Camera Position and Posture Estimation for a Single Ground-View Image using Aerial Images," IPSJ SIG Technical Report, vol.2012–CVIM–180, no.5, pp.1–8, 2012 (in Japanese).

11. T. Kanatani, H. Kume, T. Taketomi, T. Sato, and N. Yokoya, "Detection of 3D Points on Moving Objects from Point Cloud Data Based on Luminance Variation for 3D Modeling of Outdoor Environments," IEICE Technical Report, ITS2011–42, pp.153–158, 2012 (in Japanese).

12. T. Sekii, H. Kume, T. Sato and N. Yokoya, "Camera Pose Estimation from a Ground-View Image using Reference Points on an Aerial Image without Altitude Information," Proc. Meeting on Image Recognition and Understanding, 8 pages, 2012 (in Japanese).

13. H. Kume, A. Suppé, and T. Kanade, "Vehicle Pose Estimation along a Previously Driven Route using an Image Database," IPSJ SIG Technical Report, vol.2013–CVIM–186, no.13, pp.1–7, 2013 (in Japanese). (Chapter 4).

14. A. Oko, H. Kume, T. Sato, T. Machida, N. Sano, and N. Yokoya, "Free-Viewpoint Image Rendering for Evaluating Image Processing Algorithms for In-Vehicle Camera System," Proc. Forum on Information Technology, vol.3, pp.141–142, 2013 (in Japanese).

15. A. Oko, T. Sato, H. Kume, T. Machida, and N. Yokoya, "Evaluation of Image Processing Algorithms on In-Vehicle Camera System using Free-Viewpoint Image Rendering," IEICE Technical Report, ITS2013–54, pp.141–146, 2014 (in Japanese).

Commentaries

1. H. Kume, A. Suppé, and T. Kanade, "Vehicle Localization using an Image Database," Image Laboratory, vol.24, no.12, pp.1–7, 2013 (in Japanese). (Chapter 4).

Awards

1. NAIST Top Scholarship Program, 2010.