Indoor Image Geocoding using Synthetic Views

Nuno Mourão de Amorim   Jorge Gustavo Rocha
Departmento de Informática
Universidade do Minho
4710-057 Braga, Portugal
ntma90@gmail.com, jgr@di.uminho.pt

Abstract

The Global Positioning System is well known for not reaching indoor environments. Several Indoor Positioning System's have been proposed, but most of these solutions either have high accuracy errors or use expensive material to attenuate positioning errors. In this paper we propose a Computer Vision routine which is able to compute the location and orientation on indoor environments. This routine is based on Structure from Motion, an incremental algorithm which recovers the 3D structure from related photographs. The 3D structures generated are geocoded, stored in a database, and new photographs can be added at any time. By combining these 3D structures with the already existing Synthetic Views method for fast location recognition, we are able to compute the indoor GPS coordinates and orientation of new photographs in less than a second.

Keywords

Indoor Positioning System, Geocoding, Structure from Motion, Synthetic Views.

1. INTRODUCTION

Used for civil, commercial and military purposes, the *Global Positioning System (GPS)* has proven to be a resourceful and useful service. By using 4 or more satellites, this system use a trilateration process for world wide location recognition. Since the process of trilateration requires the communication between the subject to be located and the satellites, occluded zones (by bad weather or buildings) often hinder this communication which renders the *GPS* ineffective.

To complement the *GPS* in these zones, several *Indoor Positioning System (IPS)* solutions were researched. These solutions were evaluated in several performance metrics were the most relevant are: accuracy, precision, complexity, robustness, scalability and equipment cost. Presently there is not any official *IPS* because the existing solutions does not balance these metrics.

Further tackling the indoor localization problem, the *Computer Vision* community has been supporting the use of *Structure from Motion*, a *Simultaneous Location and Mapping (SLAM)* technique to geo-registrate photographs without prior information of their location. The advantage of this system related to other research's is that it does not require expensive hardware. A simple photograph taken from cellphone is all it is needed. But most of the *SFM* solutions proposed were only tested on outdoor environments, where the *GPS* signal is strong the majority of time. Besides, the implementation of their complete pipeline is not publicly available.

So, to offer continuity to this research, we were motivated to develop a prototype which uses one of the existing *Computer Vision* fast localization methods to perform image geocoding on occluded zones, without any prior information of where photographs were taken.

In this paper we want to prove that already existing methods can be applied into indoor with just few modifications. By developing a prototype, we will offer the necessary tools for the *Computer Vision* community to experiment and improve this research. We also want to show that both scalability and performance may be achieved without expensive hardware.

2. RELATED WORK

The task of indoor localization is related with several areas of research. Taking advantage of *Inertial Measurement Unit's (IMU)* device, [Woodman 08, Patrick 09] proposed an indoor location system using this measurement device. The *IMU* are known for high drifting errors which is accumulated over time. To compensate errors, other research's propose to use a combined *IMU* system with *RFID's* [Ruiz 12] or *Ultra-Wideband* measurements [Hol 09].

By using Ultrasonic waves, [Priyantha 05, Minami 04, Hazas 06] propose solutions which use these high accuracy waves for indoor localization. Although these solutions compute locations with reduced positioning errors, the hardware needed for ultra-sonic waves is expensive.

Due to high range coverage and low cost hardware usage, [J. 00, NI 03, Zhang 10, Das 14] proposed *IPS* solutions which use radio frequencies for location recognition. Subjects to be located are required to carry a small *RFID* device which acts as a receiver of tracking information. Although *RFID* solutions are able to compute positions on indoor environments, radio frequencies are affected by signal interference caused by infrastructures.

In [Mao 13, Jung 14], a solution for indoor location system using a infrared system was proposed. Subjects to be located are required to use an infrared device which periodically sends information to infrared sensors positioned along the building structure. The nearby sensors which are able to capture the messages, consequently compute the current position of users.

From the presented state of art it is noticeable that either the solution uses expensive hardware to attain precise localization or cheap and inaccurate hardware which is complemented with additional components to attenuate localization errors.

Tackling the localization problem with optical techniques, research such as [Schindler 07, Irschara 09, Li 10, Li 12] use Structure from Motion models for location recognition. Structure from Motion allows the reconstruction of 3D models seen by several related 2D photographs. Since these models store information of photographs which gave them origin, new related photographs may be added by comparing their visual information with the 3D structure. Because querying new photographs to a large database is highly inefficient, [Schindler 07, Irschara 09, Li 12] devised routines which accelerate the query phase by retrieving a set of potential useful information from the database to locate new photographs. If a new photograph validates a pose verification, then it is correctly placed within the model and the *GPS* position is returned. The advantage of these research's is allowing location recognition while only requiring cheap hardware such mobile-phone cameras.

3. IMAGE GEOCODING PROTOTYPE

From the state of art *Computer Vision* solutions, we decided to use the Synthetic Views method [Irschara 09] on our location recognition prototype. This choice was motivated by the result of their work which proves that image based localization in large environments can be made in real time.

Since this method was only experimented on outdoor environments, the following sub-sections offer a brief overview of the inherent problems of indoor *SFM* models. Afterwards we explain how the *SFM* model can be used to retrieve *GPS* coordinates when querying photographs to be geocoded. And the last subsection will provide the explanation of the implementation of our prototype and the available software used to complete the Synthetic Views pipeline.

3.1. Understanding Indoor Models

Although the theory of image geocoding on outdoor and indoor places should be the same, we questioned ourselves why the state of art research do not address experiments into indoor environments. Driven by this curiosity, we started by exploring the *SFM* process to understand to output of *3D* models. To do so, we chose *VisualSFM* [Wu b], one of the available free to use *SFM* software's to reconstruct indoor models. Data sets for both indoor and outdoor environments were gathered and we found that:

- Indoor environments are less descriptive than outdoor. These environments often contain areas with few decoration like blank walls. Since features are defined in areas where a sudden discrepancy on intensity occurs (edges, corners, ridges), indoor images often contain less number of features;
- Indoor narrow areas prevent good baseline photographs. The *SFM* reconstruction strongly depends on a chosen pair of photographs to start an incremental reconstruction. A good baseline pair is often a pair which contains a wide area of common features to support the further positioning of photographs. Due to the existence of narrow rooms, corridors, entrances, it is often difficult to take good baseline photographs.

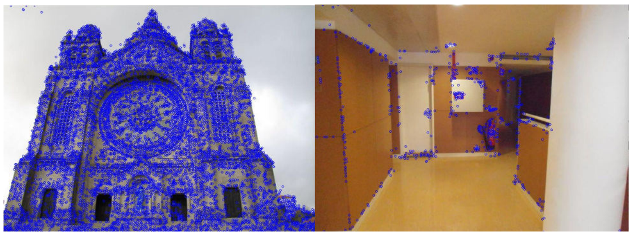

Figure 1. Example of the reduction of features extracted on indoor environments, where each blue point is a feature extracted. At the left, about 10100 features were extracted by how descriptive the church facade is. At the left, only 1621 were extracted.

Due to these facts *3D* models were often partitioned into 3 or more sub-models to avoid degeneracy. Although theoretically, these models are still usable to perform image geocoding, practically each sub-model may contain different levels of drifting, which may create pose estimations with different levels of errors even when taken in the same environment. As a workaround to this problem, we increased the number of photographs and decreased the distance between photographs positions, which slightly increased the quality of *3D* models. Since more photographs are required, each with less features than outdoor models, we had to adjust some thresholds used in *Synthetic Views*.

3.2. Geocoding with Structure from Motion

So, to geocode a photograph means to retrieve the associated *GPS* information through other geographic information. Using the *SFM* algorithm we are able to recover the structure seen by several photographs and place the

Figure 2. Example of an indoor reconstruction using VisualSFM (at the left). This structure represents an amphitheater (at the right). Analyzing the 3D structure recovered, few 3D points are visible and the structure semantic is hard to notice.

inputted photographs relatively to the generated structure. Although we may add new photographs to the reconstruction by continuing the incremental *SFM* process, none of the outputted data contains geographic information. Therefore, to relate the *3D* referential (where models are reconstructed) to the *GPS* coordinate system (where models are located in the world), the respective *GPS* coordinates of each photograph used on the model reconstruction must be injected in the system. With both coordinate systems (*3D* and *GPS*) available, it is possible to approximate a 4x4 transformation matrix *G* which maps *3D* positions *X* into *GPS* coordinates.

$$gps_{coords} = X.G \tag{1}$$

The *G* matrix can be obtained by computing an affine transformation using a set of coordinates from both *3D* and *GPS* systems. This matrix can also be used to transform the direction vector into *GPS*, since vectors can be represented as two points. In this case, the *GPS* direction can be represented in cardinal directions.

3.3. Synthetic Views Implementation

So, to develop the image geocoding prototype, we started by exploring an existing, uncompleted *SFM* code at [SFM]. Although we did not need the entire *SFM* routine (as we already had *VisualSFM* for reconstructions, we recycled functions which allow pose verification and pose estimations. For feature processing we decided to use *SIFT* [Lowe 04] as a feature descriptor, since it is also used in [Irschara 09].

The Synthetic Views method is divided into 2 phases:

- An offline phase where 3D models are refined into a compressed yet representative structure, and stored into an image database;
- An online phase where the image database is used to geocode incoming images.

As stated in [Irschara 09, p. 2], 3D models outputted by *SFM* often contain several redundant descriptors. In order to reduce the feature repeatability, [Irschara 09, p. 2] propose the use of *Mean Shift Clustering* [Comaniciu 02]. This method applies a global threshold to cluster features which have an high level of similarity, without losing the pose estimation capability. In our implementation we used the mean of the descriptors associated to each *3D* point. Although this may lead to a much more aggressive compression, after performing few tests we noticed that the information kept is still capable of allowing a good feature matching rate. As both orientation and scale will be needed afterwards, we also compute their mean for each *3D* point and extrapolate them into the *3D* coordinate system. While the *3D* orientation is given by the mean of the direction of each associated 2D point to the respective camera view points, the *3D* scale is computed by the following equation.

$$scale_{3D} = \frac{scale * distance_{viewTo3DPoint}}{focallength} \tag{2}$$

After compressing relevant information, artificial cameras are placed uniformly around the *3D* model and each one takes a snapshot. To place each camera, we compute the model ground plane by approximating a plane to the position of each original view. This plane is then divided into a grid and 12 views (each with 30°viewing direction difference) are placed into each grid position. Since indoor models are narrower than outdoor models, we do not tilt the viewing direction of each synthetic view by 10°as in [Irschara 09, p. 4]. For each placed synthetic view, 3D visible points are re-projected into these cameras and artificial photographs are created, each containing the compressed 2D and 3D point information. A 3D point is only visible by a given synthetic view if:

- Its position lies within the camera view frustrum culling;
- The scale of 3D point is higher than 1 in terms of *Difference of Gaussian (DoG)*;
- The difference of its 3D orientation with the viewing direction of the current synthetic view is lower than 30°(face culling).

This process is done iteratively until we fill the ground plane with synthetic views. With all the views placed, the best views to represent the model are selected. To do this selection, each synthetic view is evaluated by their view coverage. Both original views and synthetic views are used to build a square binary matrix, where 1 means that a view *A* covers *B* and 0 the opposite. For outdoor models we consider the proposed values for view coverage, so, a view *A* covers *B* if *A* contains at least 150 points seen by *B*. For indoor, to adapt to the reduction of points per view explained in 3.1 this threshold is set to 30 points. Beside this condition, it is also defined that each view covers itself. The best views to represent the 3D model are retrieved by applying the greedy algorithm described in [Irschara 09, p. 4]).

The information of the best synthetic views are then stored in *3D* documents containing *3D* visible points, their associated *2D* points, compressed descriptors and a *GPS* matrix

G described in 3.2 which is computed before compressing the models. Also, their descriptors are injected into a vocabulary tree to support the geocoding process.

Vocabulary Tree is a structure which allows the creation of an image database, where the descriptors of images are transcribed into 'words' and propagated through a n-ary balanced tree. The main idea behind the vocabulary tree is that similar descriptors will be propagated to the same leafs. When querying an image against the vocabulary tree, the more similarity the propagation is, the higher the probability of 2 images positively match when performing feature matching. Furthermore, this image database structure is scalable and adaptable to higher databases as it allows the creation of trees with more levels (which refines the result of queries) and more branches (which allows a better descriptor distinction). Although [Irschara 09, p. 5] used a vocabulary tree which benefits the use of the *GPU* to speed up the top document retrieval and their probabilistic scoring function which allows a more precise retrieval of the top best documents to match, we are using the implementation available in [Snavely] which corresponds to the vocabulary tree researched in [Nist 06]. This decision was based on the fact that we needed the full compression process available as soon as possible and the vocabulary tree document retrieval available at [Snavely] was outputting the desired results in our experiments.

After compressing enough models, in the online phase new photographs are queried against the image database and a pose estimation is delivered whenever a positive match is returned.

The online phase of our prototype is defined by the following operations:

- Feature Extraction on the query photograph;
- Query vocabulary tree for top matches;
- Feature match between the query image and the synthetic views retrieved;
- Solve *Perspective-N-Point Ransac* to pose estimate the query image;
- Compute the *GPS* position and orientation.

To perform feature extraction and matching we used *SiftGPU* [Wu a] a fast and accurate implementation of *SIFT* [Lowe 04] which benefits the computers *GPU* and the *CUDA* toolkit to fasten heavy matrix operations. For feature extraction we extract around 1280 features for the query image, as it is an high enough number to contain relevant features for the pose estimation phase. The descriptors extracted are then queried in the vocabulary tree and the top 10 best documents are retrieved. Feature matching is then performed between the query and 3D documents features. Once a positive match is found, we try to compute the projection matrix by applying the *Perspective-N-Point Ransac* supplied by the *OpenCV* library [Ope] to evaluate the coherency of the 2D query points with the 3D document points. If the computed projection matrix validates at least 10 inlier points, we define the query image as pose estimated. Using the *G* matrix stored within the *3D* document which matched the current image to geocode, the *3D* position and orientation returned by the pose estimation are extrapolated into *GPS* coordinates.

The source code developed for our prototype can be found at [Amorim 14].

4. DATASET AND RESULTS

To evaluate the performance of *Synthetic Views* for geocoding photographs we gathered a data set of 443 indoor and 802 outdoor of geocoded photographs from locations in Braga, Montalegre and Viana do Castelo in Portugal.

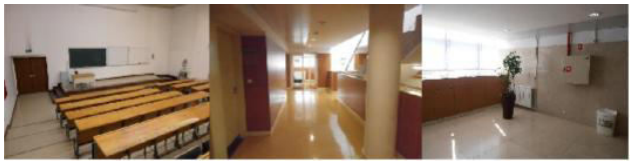

Figure 3. Some of the indoor photographs used to evaluate the performance of our prototype.

From these photographs, we built 12 *3D* models with *VisualSFM*, which resulted in 215403 *3D* points projected by 808293 features extracted. We then applied the *Synthetic Views* compression on the reconstructed models to remove the excess of information stored in the vocabulary tree. Table 1 shows the compression value obtained with our prototype.

Original Number of Descriptors	808293
Compressed Number of Descriptors	231942

Table 1. Number of descriptors before and after applying the compression.

From the selected locations, another set of 25 indoor and 25 outdoor geocoded photographs were collected and sequentially geocoded to the compressed database. Their associated *GPS* coordinates were only used to compute precision errors between the estimated coordinates and their ground truth. Table 2 provides the estimation rate and time spent on geocoding each photograph and Table 3 presents the average time spent on each operation when geocoding a single photograph. Each query photograph used has 1000 pixels width and 750 height.

These results were attained on a computer with a CPU Intel i5-4200U 1.6 GHz with a GPU nVidia GeForce 820M, while using the CUDA version of SiftGPU and a vocabulary tree of 5 levels and 10 branches to store documents.

So, analysing the provided tables, with Synthetic Views we were able to compress redundant descriptors on our

	Indoor	Outdoor
Geocode Rate	16 (64%)	22 (88%)
Precision Recall	[0.307, 4.050] m	[0.252, 7.620] m
Mean Precision	1.160 m	2.560 m
Mean Time	0.451 s	0.520 s

Table 2. Overall statistics from pose estimating 25 indoor and 25 outdoor 1000x750 photographs with Synthetic Views.

	Indoor	Outdoor
Keypoint Extraction (1280)	0.216	0.230
Vocabulary Tree (top 10)	0.178	0.181
F. Matcher (1280x2500)	0.015 * N	0.015 * N
PnP Ransac	0.001 * N	0.001 * N
Compute GPS Coordinates	0.0001	0.0001

Table 3. Computational time expressed in seconds for each operation on Synthetic Views. The value N indicates the number of top documents to retrieve. On feature matching, the computational time is based on a comparison of 1280 query features with 2500 document features. Both number of features are based on the mean of features extracted and features stored within 3D documents.

database by more than 71% of its initial value. As stated before, we increased the number of photographs taken per environment and decreased the distance between the photographs positions. This resulted in more stable indoor models, but with an high level of redundancy, which justifies the advantage of the compression phase made by Synthetic Views.

Although the compression rate is high, we were able to geocode 16 indoor and 22 outdoor photographs from our input data set of 50 photographs. Due to the discrepancy of indoor successful geocoded photographs to outdoor, we believe that this method is still not adapted to indoor environments. When compressing our database, we noticed that indoor models required more synthetic views placed compared to outdoor. Since each indoor view shares few features with other views due to reasons explained in 3.1, for some indoor models the 30 points coverage threshold was not allowing a good coverage, which justifies the need of an higher number of views to completely cover indoor models. Lowering this threshold degenerated all indoor models and hindered the geocoding process. As the source of this problem comes from the consistency of indoor models, we propose the use of special wide angle cameras to augment the environment coverage (more features per photograph) and increase the overlap between photographs. Reconstructing indoor environments with these photographs should deliver refined indoor models.

The accuracy of our prototype should also be improved. Successfully geocoded indoor photographs were located with a mean precision error of 1.160 meters. Outdoor photographs reached the 2.560 meters errors. While, for oudoor models, 2 meters errors may not imply inaccurate localization, for indoor the same errors may place photograph behind walls or bellow floors more often. Here, we proposed the use of Multicore Bundle Adjustment [Wu 11], to locally optimize positioning errors. This algorithm benefits the GPU and CPU to fasten the optimization, which allows more accurate precision while maintaining the geocoding performance.

As for the geocoding speed of this method, with our database we are able to geocode new indoor photographs with a computational time ranging from 0.410 seconds to 0.554 seconds and outdoor photographs from 0.427 to 0.571 depending on which top document query images match. It is natural that outdoor photographs spend more time to be geocoded compared to indoor, since we are processing an higher amount of extracted features.

Although we did not evaluate our prototype with larger data sets, the only operation that should need adjusting is the vocabulary tree querying. As the database increases, more descriptor distinction is required in order to not confuse the top documents retrieved. As stated before, the number of levels and branches of the tree may be adjusted to the stored information.

5. CONCLUSION AND FUTURE WORK

We presented a method which leverages speed with scalability to solve the indoor localization problem. Our contribution was the development of a running prototype. Our indoor geocode system is able to receive images to be geocoded and delivers their *GPS* position and orientation. By making available our code to the *Computer Vision* community we are contributing to the discussion and refinement of state of the art mechanisms used in indoor localization problem.

As future work, we aim to improve our *Synthetic Views* based prototype, by integrating the vocabulary described in [Irschara 09, p. 5] and apply the Multicore Bundle Adjustment on the final geocoding position. We believe that this will greatly boost the performance and refine the accuracy of our system in larger scales.

In the short future, we will prepare and share some data sets, for indoor spaces, to use for benchmarking different tools and approaches. These data sets will improve and speed up the evaluation of new methods.

To further prove the utility of our image based geocoding system, we will develop a client-server service where users can send photographs from their mobile phones, to be geocoded. Our server returns the estimated *GPS* position of the photograph sent.

Image geocoding based on feature prioritization [Li 10] is also being explored in the recent literature. We are very interested in the development of another prototype, based on *Prioritized Features*, to see how it compares in terms of accuracy, speed and scalability, with the existent synthetic views based prototype.

References

[Amorim 14] Nuno Mourão Amorim. Image Geocoding Prototype. `https://bitbucket.org/nam81/qt-sfm.git`, 2014. Accessed 20 October 2014.

[Comaniciu 02] Dorin Comaniciu, Peter Meer, and Senior Member. Mean Shift: A Robust Approach Toward Feature Space Analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):603–619, 2002.

[Das 14] Piyali Das and Dharma. P. Argawal. RFID for Indoor Position Determination. *Instrumentation and Measurement, IEEE Transactions*, pages 561–567, 2014.

[Hazas 06] M. Hazas and A. Hopper. Broadband ultrasonic location systems for improved indoor positioning. *Mobile Computing, IEEE Transactions*, pages 536–547, 2006.

[Hol 09] Jeroen D. Hol, Fred. Dijkstra, and Schon Thomas B. Luinge, Henk. Tightly Coupled UWB/IMU Pose Estimation. *Proceedings of the IEEE International Conference on Ultra-Wideband (ICUWB)*, 2009.

[Irschara 09] Arnold Irschara, Christopher Zach, Jan-Michael Frahm, and Horst Bischof. From structure-from-motion point clouds to fast location recognition. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2599–2606, June 2009.

[J. 00] Hightower J., Want R., and Borriello G. SpotON: An indoor 3D location sensing techology based on RF signal strength. *UW CSE00-02-02*, 2000.

[Jung 14] H. Oh. Jung, Kim Doojin, and H. Lee Beom. An Indoor Localization System for Mobile Robots Using an Active Infrared Positioning Sensor . *Journal of Industrial and Intelligent Information*, pages 35–38, 2014.

[Li 10] Yunpeng Li, Noah Snavely, and Daniel P Huttenlocher. Location Recognition using Prioritized Feature Matching. *ECCV'10 Proceedings of the 11th European conference on Computer vision*, pages 791–804, 2010.

[Li 12] Yunpeng Li, Noah Snavely, Dan Huttenlocher, and Pascal Fua. Worldwide Pose Estimation using 3D Point Clouds. *ECCV'12 Proceedings of the 12th European conference on Computer Vision*, pages 15–29, 2012.

[Lowe 04] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, pages 91–110, 2004.

[Mao 13] Ling Mao, Jiapin Chen, Zhenbo Li, and Dawei Zhang. Relative Localization Method of Multiple Micro Robots Based on Simple Sensors. *International Journal of Advanced Robotic Systems*, 2013.

[Minami 04] M. Minami, Y. Fukuju, K. Hirasawa, S. Yokoyama, M. Mizumachi, H. Morikawa, and T. Aoyama. Dolphin: A pratical approach for implementing a fully distributed indoor ultrasonic positioning system. *Ubicomp*, pages 347–365, 2004.

[NI 03] Lionel M. NI, Liu Yunhao, Cho Lau Yiu, and Abhishek P. Patil. LANDMARC: Indoor Location Sensing Using Active RFID. *Pervasive Computing and Communications, 2003. (PerCom 2003). Proceedings of the First IEEE International Conference*, pages 407–415, 2003.

[Nist 06] David Nist and Henrik Stew. Scalable Recognition with a Vocabulary Tree. *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2161–2168, 2006.

[Ope] OpenCV (Open Source Computer Vision). `http://opencv.org/`. Accessed 7 May 2014.

[Patrick 09] Robertson Patrick, Michael Angermann, and Bernhard Krach. Simultaneous localization and mapping for pedestrians using only foot-mounted inertial sensors. *Proceedings of the 11th international conference on Ubiquitous computing*, pages 93–96, 2009.

[Priyantha 05] N. B. Priyantha. The cricket indoor location system. *PhD Thesis, Massachusetts Institute of Technology*, page 199, 2005.

[Ruiz 12] A.R.J. Ruiz, F.S. Granja, J.C. Priesto Honorato, and J.I.G. Rosas. Accurate Pedestrian Indoor Navigation by Tightly Coupling Foot-Mounted IMU and RFID Measurements. *Instrumentation and Measurement, IEEE Transactions*, pages 178–189, 2012.

[Schindler 07] G. Schindler, M. Brown, and R. Szeliski. City-Scale Location Recognition. *Computer Vision and Pattern Recognition, CVPR '07. IEEE Conference*, pages 1–7, 2007.

[SFM] Structure from Motion ToyLib. `https://github.com/royshil/SfM-Toy-Library`. Accessed 7 May 2014.

[Snavely] Noah Snavely. VocabTree2. `https://github.com/snavely/VocabTree2`. Accessed 7 May 2014.

[Woodman 08] Oliver Woodman and Robert Harle. Pedestrian localisation for indoor environments. *UbiComp '08 Proceedings of the 10th international conference on Ubiquitous computing*, pages 114–123, 2008.

[Wu a] Changchang Wu. SiftGPU. `http://cs.unc.edu/~ccwu/siftgpu/`. Accessed 8 May 2014.

[Wu b] Changchang Wu. VisualSFM - A Visual Structure from Motion System. `http://ccwu.me/vsfm/`. Accessed 7 May 2014.

[Wu 11] Changchang Wu, Sameer Agarwal, Brian Curless, and Steven M. Seitz. Multicore Bundle Adjustment. *CVPR 2011*, pages 3057–3064, 2011.

[Zhang 10] Dian Zhang, Yanyan Yang, Dachao Cheng, Shiyuan Liu, and Lionel NI. COCKTAIL: An RF-based Hybrid Approach for Indoor Localization. *IEEE International Conference on Communications (ICC)*, pages 1–5, 2010.