Image Based-Localization on Mobile Devices Using Geometric Features of Buildings

Hasinarivo Ramanana
Dept. of Mathematics and Computer Science
University of Antananarivo
Madagascar
e-mail: hasram006@gmail.com

Andriamasinoro Rahajaniaina
Department of Mathematics, Computing and Applications,
University of Toamasina,
Madagascar
e-mail: hajatoam@gmail.com

Jean-Pierre Jessel
IRIT, VORTEX
Paul Sabatier University
Toulouse
France
e-mail: jessel@irit.fr

***Abstract*—Outdoor localization is problem that many people are facing in everyday life. One way to determine the location of a user is to use an image-based localization method. In this paper, we propose an approach based on geometric features of buildings to address image-based localization in an outdoor environment. Our proposed method can be described as follows: first, we compute descriptors of buildings façades at the scene by using cross-ratios, then, we match them to images in a database, we get the length of the façade retrieved and we estimate the location of the user's camera. We use cross-ratios to compute the descriptors of buildings because it is a projective invariant. Our method is tested with buildings within a campus and a Geographic Information System (GIS) created from OpenStreetMap.**

Keywords-Image-based localization; cross-ratios; GIS; building recognition.

I. INTRODUCTION

Image-based localization consists in determining the position of a user's camera, i.e., the user 's position, by computing the distance from known objects in an image. We use it in many domains such as robot localization and landmark recognition. In urban environment, tall buildings may interfere with the Global Positioning System (GPS) signal which results in an inaccurate localization. In such a situation, an image-based localization may replace the GPS-based solution [16]. Generally, image-based localization is composed of three main steps: performing image matching of the scene from all geo-referenced images stored in database and retaining the best candidate to find the approximated area of the user, computing the length of a known object in the scene and estimating the position of the user.

For the first step, special features of the image are extracted to differentiate it from the other images in the database. In [1], [7] and [8] authors used Scale Invariant Feature Transform (SIFT) descriptor [2] for image matching. In [3], Roberto Cipolla utilized Harris-Stephens detector [4]. In [5], C. Card and W. Hoff used Oriented FAST and Rotated BRIEF (ORB) [6] descriptor for image matching.

This article proposes an image-based localization approach in an urban environment which extracts geometric features of building's façade as keypoints. We use cross-ratios to describe buildings images. The remainder of this paper is organized as follows. Section 2 lists the related works. Section 3 describes our proposed method in detail. Section 4 explains our experiences and our results. Section 5 is the conclusion.

II. RELATED WORKS

There are many researchers who proposed techniques for outdoor localization in a city using mobile devices. In this section, we will list some works relating to image-based localization and building recognition.

Johansson and Cipolla [13] proposed a technique that uses the parallel planes in buildings to find the homography of images in a city scene and hence predict the location of the camera. This approach reduces the amount of memory allocated for images in the database but it is lacking in precision.

N. Haala and J. Bohm [14] presented a system for locating a building in a city using a database of 3D models of buildings. They convert a 3D model into a single 2D view per orientation for each image and apply a 2D to 2D matching. To recognize the building, the system extracts the edges and corners by the Generalized Hough Transform. The telepointing device used for their approach is composed of a camera, a GPS receiver, an electronic compass, a tilt sensor and a laptop. All of these materials are hard to bear.

In 2004, D. Robertson and R. Cipolla [15] worked on a localization technique in urban environment using a smartphone. The user takes a picture of his surroundings and sends this image as a query to a server which searches in a database of façades. The system computes the vanishing points of query image in horizontal and vertical directions by extracting lines in theses principal directions. These vanishing points will be used for camera pose estimation. After this process, the system computes descriptors in query image based on Harris corner detection. The descriptor is defined by a vector of 8x8 matrix of Red Green Blue (RGB) pixel values centered in each interest point. The detection of interest points is repeated with different image scales using a pyramid of scaled images. After that, the matching is executed at each level of scale in the pyramid to achieve robust matching. This approach requires additional memory

 ISBN: 978-1-61208-469-5

space and high computation time, which decreases the speed of the process.

In 2009, N. Yazawa and H. Uchiyama [17] developed a system for estimating user position by matching a captured image from a camera equipped with a compass and GPS into a database of 104 panoramas. The method Speeded Up Robust Features [20] (SURF) is used for image comparison and triangulation for estimating the camera pose. The matching of SURF features with panoramas in the database and the captured image took 400 seconds. In our system, we want to improve this time computation.

In 2013, M. Donoser and D. Schmalstieg [18] introduced a discriminative classification problem for matching interest points detected in the query image and the 3D point in the known world. They compared their method with the standard Nearest Neighbor, Random Fern and Random Forest. The result proves that their proposed method gives the highest value of mean classification accuracies and standard deviations.

In 2015, B. Zeisl and T. Sattler [19] presented a voting-based pose estimation strategy for matching images in the database and query image. They wanted to compare spatial verification and appearance-based filtering.

III. PROPOSED METHOD

A. System overview

Our proposed method is divided in two parts: server side and client side. The client side is composed of a smartphone

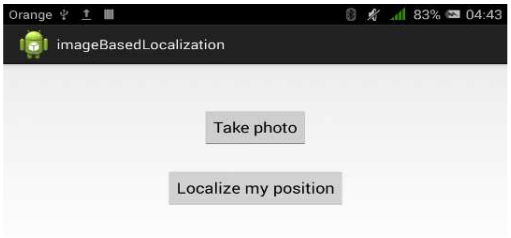


Figure 1. Home interface of the application

with which the user takes pictures of his surroundings with the smartphone's camera. The photo of buildings will serve to estimate his position. The user interface of our application is shown in Figure 1.

The information related to a specific building is shown on the smartphone's screen after identification. After that, the system estimates the user position. The server stores all images of buildings that people have already taken, computes descriptors of images according to the general structure of buildings and saves them in a YAML file (acronym of "YAML Ain't Markup Language") which will be uploaded to the smartphone. Figure 2 shows this process.

This YAML file is coupled with 2D GIS which contains the spatial disposition of buildings within the campus that we are interested in our experience. We use OpenStreetMap and Quantum Geographic Information System [21] (QGIS) software to create these data. A screenshot of the spatial data of the campus, exploited within QGIS, is shown in Figure 3.

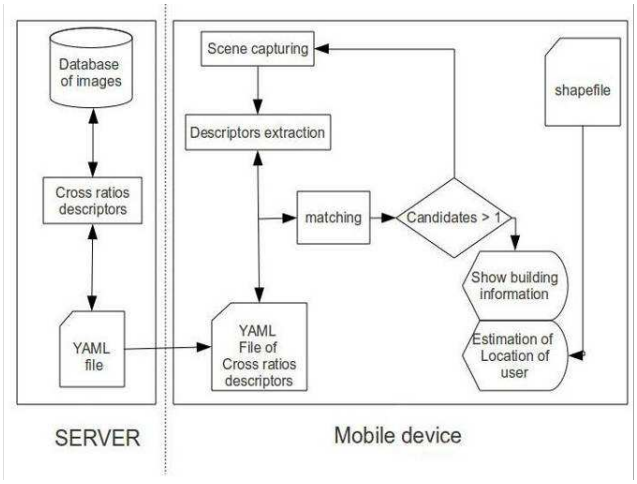


Figure 2. Flow chart diagram describing the system

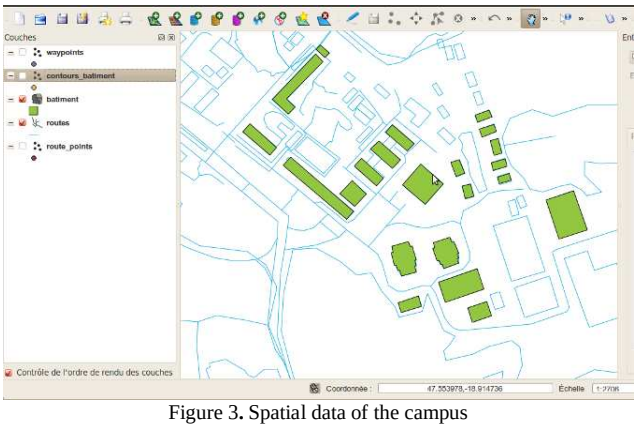


Figure 3. Spatial data of the campus

B. Descriptors extraction and images matching

Most of the time, buildings present a lot of linear structures resulted from windows, doors and facades' arrangement (Figure 4-a). We will use these linear structures to detect keypoints of buildings. To extract these keypoints, we follow this algorithm:

Step 1: Convert the image into greyscale and apply the Canny Edge detector [23] on that image (Figure 4-b).

Step 2: Apply Standard Hough line Transform [24] and keep only the lines in the vanishing directions (horizontal and vertical). Figure 4-c shows vertical (green) and horizontal (red) lines after Hough lines Transform.

Step 3: Extract the intersections of lines (blue dots in Figure 4-d) in different vanishing directions and keep them as relevant keypoints for that image.

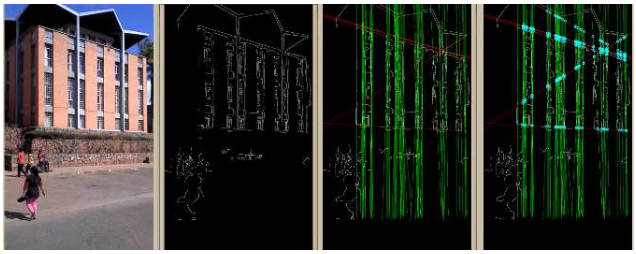
From these keypoints, we compute the descriptor of each keypoint as the cross-ratio of four collinear points and store them in a matrix. We choose cross-ratio because it is a projective invariant. The cross ratios CR of four collinear points P1, P2, P3 and P4 is defined as follows:

 ISBN: 978-1-61208-469-5

$$CR = \frac{(x3 - x1)(x4 - x2)}{(x3 - x2)(x4 - x1)} \quad (1)$$

where x1, x2, x3 and x4 are respectively the values of x-coordinates of points P 1, P2, P3 and P4.

To match cross ratios descriptors of two images, we use Fast Library for Approximate Nearest Neighbour (FLANN)



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

Figure 4. Extraction of intersections of lines.

[9] based matching to find the best matches of descriptors in the scene. FLANN is enhanced by Random Sample Concensus [22] (RANSAC) algorithm to remove outliers. We build indexes using Locality Sensitive Hashing (LSH) [10] [11] which is robust in high dimension of data. With LSH, we can achieve faster matching.

C. First pose estimation from GIS query

The list of intersections from the descriptor extraction step will serve as input to track the edges of the building. The coordinates of intersections are stored in a matrix called intersections matrix. This matrix is shown in Figure 5. We keep the external coordinates of intersections to mark the building location in the image. We query the name of the building into the GIS to find its position on the map. At this step, the approximated user position is estimated as near the facade of this building.

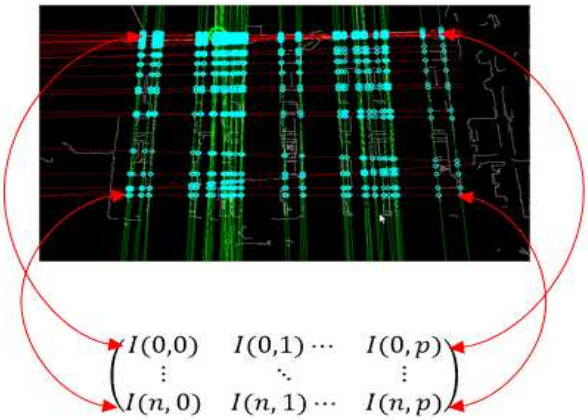


Figure 5. Matrix of coordinates of intersection points. Blue dots represent the extracted intersections of line in step 3.

D. Camera pose estimation from camera parameters and GIS

After retrieving the approximated user position, we would like to know his exact position by camera pose estimation. For that, we compute the homography of the facade extracted in the camera's screen and the facade in the GIS. We can compute the camera pose estimation by computing the camera intrinsic by this matrix as in [12]:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = A[RT] \begin{pmatrix} x_{wc} \\ y_{wc} \\ z_{wc} \\ 1 \end{pmatrix} \quad (2)$$

where A is the camera intrinsic matrix,

R and T are the extrinsic parameters (Rotation and Translation matrix) of the camera, x_{wc}, y_{wc} and z_{wc} are world coordinates of one point and $[u\ v]^T$ are the coordinate of one point in pixel coordinates.

$$A = \begin{pmatrix} \alpha_x & \gamma & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3)$$

where γ is the skew (often we take 0 as its value), $[u_0,v_0]^T$ is called the principal point, usually the coordinates of the image center, α_x and α_y are the scale factor in the x and y coordinate directions, and are proportional to the focal length f of the camera:

$$\begin{cases} \alpha_x = k_x f \\ \alpha_y = k_y f \end{cases} \quad (4)$$

k_x and k_y are the number of pixels per unit distance in x and y directions.

The coordinates of camera in the world coordinates are given by:

$$C = -R^{-1}T \quad (5)$$

IV. EXPERIENCE AND RESULTS

We perform our experiment with an Alcatel One Touch Pixi 3, a low cost smartphone, which has the following specifications: processor: MediaTek MT6572M - 1 GHz Dual Core, OS: Android 4.4.2 KitKat, RAM 512 Mbyte. Our database is composed of 100 images of buildings taken around a campus.

Here, we show some matching results of images in the scene and from the database: the images on the left are query images taken from smartphone, and the images on the right

 ISBN: 978-1-61208-469-5

are those stored in the database. Here we prove that our cross-ratio descriptors are perspective invariant. Thus, the result image from the database can be the image of the same building but in different view point as we see in the first line of result images (Figure 6-b).

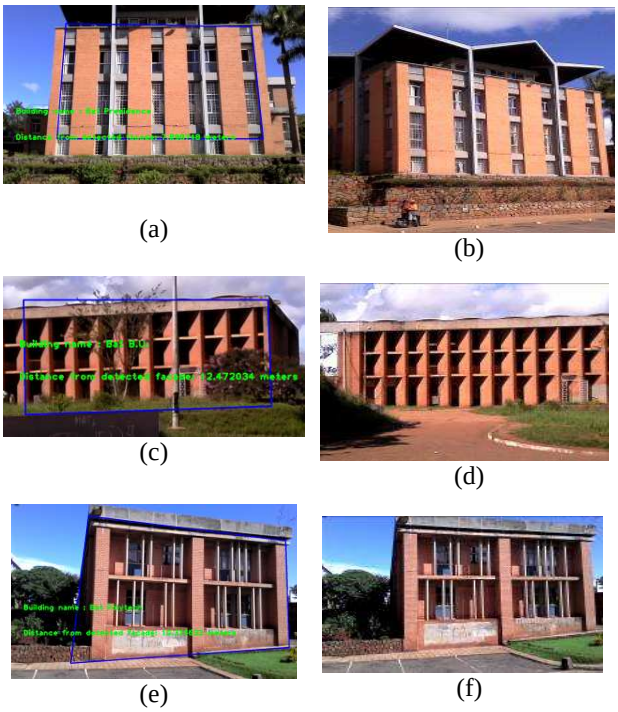


Figure 6. Some results of building localization

In addition, the building's façade is marked within a blue rectangle and the information about its position is printed on the screen. The time computation for matching 100 images in the database took about 120 milliseconds with this low cost smartphone. This demonstrates the accuracy of our approach for real-time application.

V. CONCLUSION AND FUTURE WORK

In this paper, we present an urban image-based localization method in mobile devices using cross-ratios of feature points detected on the building's facades. These feature points are intersections of horizontal and vertical lines in the vanishing directions after applying Hough transform. All coordinates of intersections are kept in a matrix which will be used to detect the edges of building's facade. We compute the pose estimation of user in the world coordinates in keeping with length of facade in the GIS and length of the same facade in the smartphone's screen.

Our approach can be used for other images with linear structure such as trains or buses, in order to classify them. We can improve the method presented in this paper by segmenting the building in the image. In addition, we can use parallel computing and middleware to perform our technique in order to reach a better performance.

REFERENCES

[1] W. Zhang and J. Kosecka, "Image Based Localization in Urban Environments," Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission, June 2006, pp. 33-40.

[2] D. G. Lowe, "Object recognition from local scale-invariant features". Proc. 7th International Conference on Computer Vision (ICCV'99), Corfu, Greece, 1999, pp. 1150-1157.

[3] D. Robertson and R.Cipolla, "An image-based System for urban Navigation", In Proc. British Machine Vision Conference, Kingston, UK, 2004, pp. 819-828.

[4] C. G. Harris and M. Stephens. "A combined corner and edge detector". In Proc 4th Alvey Vision Conf, Manchester, 1988, pp. 147–151.

[5] C. Card and W. Hoff, "Qualitative Image-Based Localization in a Large Building", IPCV, 2015, pp. 338-344.

[6] E. Rublee, V. Rabaud, K. Konolige, and Gary Bradski, "ORB: an efficient alternative to SIFT or SURF", Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, 2011, pp. 2564-2571, doi: 10.1109/ICCV.2011.6126544.

[7] L. Carozza, F. Bosché, and M. Abdel-Wahab, "Image-based localization for an indoor vr/ar construction training system". CONVR, 2013.

[8] Y. Huang et al.,"Image-based Localization for Indoor Environment Using Mobile Phone",The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2015, pp. 211-215.

[9] M. Muja and D. G. Lowe, "Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration", In Proc. International Conference on Computer Vision Theory and Applications (VISAPP'09) (Lisbon, Portugal), 2009, pp. 331—340.

[10] P. Indyk and R. Motwani, "Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality", In Proceedings of 30th Symposium on Theory of Computing, 1998

[11] A. Andoni and P. Indyk, "Near-optimal hashing algorithm for approximate nearest neighbour in high dimensions", Communications of the ACM, Vol. 51, 2008

[12] P. R. S. Mendonca and R. Cipolla, "A Simple Technique for Self-Calibration", Computer Vision and Pattern Recognition (CVPR). IEEE Computer Society Conference on. (Volume:1), June 1999., pp. 500-505

[13] B. Johansson and R. Cipolla, "A system for automatic pose-estimation from a single image in a city scene", In IASTED Int. Conf. Signal Processing Pattern Recognition and Applications, Greece., 2002

[14] Haala, N., Böhm, J."A multi-sensor system for positioning in urban environments". ISPRS J PHOTOGRAMM, 58 (1-2), 2003 pp. 31-42. doi:10.1016/S0924-2716(03)00015-7.

[15] D. Robertson and R.Cipolla, "An image-based System for urban Navigation", In Proc. British Machine Vision Conference, Kingston, UK., pp. 819-828, 2004

[16] N. Bioret, G. Moreau and M. Servières , "Urban Localization based on Correspondences between Street Photographs and 2D Building GIS Layer",CORESA, Toulouse, France, 2009.

[17] N. Yazawa and H. Uchiyama, "Image based view localization system retrieving from a panorama database by surf", in Proc. of the IAPR Conference on Machine Vision. Applications, 2009, pp. 118-121.

[18] M. Donoser and D. Schmalstieg, "Discriminative Feature-to-Point Matching in Image-Based Localization", Conference CVPR , IEEE, 2014, pp. 516-523.

[19] B. Zeisl, T/ Sattler, and Marc Pollefeys, "Camera Pose Voting for Large-Scale Image-Based Localization",Conference ICCV, IEEE, 2015, pp. 2704-2712.

 ISBN: 978-1-61208-469-5

[20] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: Speeded up robust features", In Proc. ECCV, 2006, pp. 404–417.

[21] Quantum GIS, http://qgis.org/.

[22] 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, 1981, pp.381–395.

[23] J. Canny, "A Computational Approach to Edge Detection", IEEE Trans. Pattern Analysis and Machine Intelligence, 1986, pp.679–698, 1986.

[24] D. H. Ballard, "Generalizing the Hough Transform to Detect Arbitrary Shapes". Pattern Recognition, 1981, pp. 111–122, doi:10.1016/0031-3203(81)90009-1.

 ISBN: 978-1-61208-469-5