Published in *Proceedings of the JURSE 2023*, Heraklion, Greece, 17-19 may 2023, which should be cited to refer to this work. URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10144171&tag=1

Geolocation of a panoramic camera by reference pairing

ZEDE Chahine-Nicolas,
School of Management and Engineering Vaud, HES-SO
University of Applied Sciences Western Switzerland
Yverdon-les-Bains, Suisse
chahine-nicolas@outlook.fr

GRESSIN Adrien,
School of Management and Engineering Vaud, HES-SO
University of Applied Sciences Western Switzerland
Yverdon-les-Bains, Suisse
adrien.gressin@heig-vd.ch

***Abstract*— Panoramic cameras are now available to a large audience. They provide good results on photogrammetry application, but they are still limited by their positioning. This project aims to geolocate a commercial 360° camera in an urban environment, by extracting points in fisheye images and match them with reference from a LiDAR (Light Detection and Ranging) dataset. Such reference points are located on the horizon line, visible from the camera point of view. Matching points are then introduced as Ground Control Points to improve the camera positioning accuracy. A fully automatic solution for position refinement, based on LiDAR data is proposed in this paper.**

***Keywords*— fisheye camera, geolocation, matching, horizon line contour, LiDAR**

I. INTRODUCTION

It is possible for photogrammetric survey to produce a high-quality measurement of a scene with mainstream cameras (e.g GoPro). Their biggest dilemma is to achieve a high precision geolocation, especially in dense urban environments where GNSS (Global Navigation Satellite System) data give poor accuracy.

Terrestrial photogrammetry can reproduce a highly detailed scene in a short amount of time. An example is the update of underground networks maps because pipes can be intertwined. [1] choose to combine photogrammetry with deep learning segmentation framework on the images to automate the detection of each piece. After a scene reconstruction, the model is referenced thanks to a Real Time Kinematics (RTK) GPS associated with the camera. A limitation of using RTK GNSS [2] is the masking and multipath effects on the GPS signals. Surrounding buildings and construction mask part of the sky, which is not favorable to obtain precise GNSS measurements [3].

In this article, the main goal is to refine the camera geolocation with enough "clues" visible in the images captured that can be found on an already georeferenced dataset. Such "clues" can be introduced as Ground Control Points (GCPs) in a bundle adjustment method. For this purpose, panoramic cameras have an advantage for their capacity to record at 360°. Several existing datasets may be useful for our project, such as official surveys, OpenStreetMap or LiDAR data. The target accuracy for the camera positioning after refinement is about 10 to 20 cm. We chose to use the high-density LiDAR data from swissTopo [4] (> 20 pts/m²) as reference.

Our method, detailed in this paper, is the following. To find GCPs, similar points of view are made inside the points cloud, by reading EXIF data and intrinsic lens properties, creating "simulated" images. The first object of interest is the Horizon Line Contour (HLC) in both real and simulated. It can be extracted by doing a semantic segmentation of the sky pixels. Then, two HLCs can be matched with the Dynamic Time Warping (DTW) [5] pairing the corresponding indices of both lists and thus, create GCPs needed for a new position estimation. It highlights a way to compare a 2D image data and a 3D DEM or LiDAR data.

II. STATE OF THE ART

Many works on the geolocation of an image exist, as shown by [6], for each type of spatial information and degree of automation.

The first method is bundle adjustment, with GCPs or already referenced cameras [7]. One of the limitations is that the existing set of photographs is heterogeneously distributed over the territory. To make GCPs, the pixel and terrain coordinates must be known. For pixel coordinates, manual selection, or automatic detection thanks to targets can be used. They must be well distributed in the area of interest and be measured by a topographic method (GNSS or total station for example). This solution is already applicable, but it is difficult to set up and time consuming. Meanwhile, GCPs may be automatically extracted from other references.

The desired solution must be autonomous to be usable in practice. [8], inspired by the game Geoguessr, try to estimate the localization of an image with the help of machine learning. The training was done all over the United States, but the neural networks had difficulties to find the correct location, due to images similarity and repartition (mostly on road). Google VPS (Visual Positioning System) [9] combines Google Street View with the camera's observations, which is promising in dense urban areas. Its principle is to detect stable features overtime (such as building, contrary to vegetation) as GCPs. This work shows the benefit of "classified" GCPs.

Other similar studies [10], [11], [12] use the Horizon Line Contour as a location describer. In vast zones, such as Switzerland [10], [11], they benefit from characteristic mountain ranges. Here, the idea is to extract the HLC of urban objects and compare it to a HLC extracted from a Digital Elevation Model (DEM). With the help of DTW [5], 2D and 3D coordinates can be linked to create a reference point.

Regarding the previous methods, the chosen strategy is to build robust GCPs in fisheye images, from visible time-robust features. It can be done by associating points easily recognizable in images and in a *virtual* camera, such as corners and edges of visible objects. To help corner detection, a sky segmentation creates a binary mask, facilitating the skyline (and so building edges) extraction. As a result, a new camera's pose estimation is computed.

III. DATASETS

To test this strategy, the GoPro MAX 360 was chosen for its couple of wide-angle lenses and for being accessible. 30 videos were taken between March and May 2022, around the main building of HEIG-VD. The video file ".360" produced by the GoPro, is regularly cut into two panoramic images, which are themselves transformed into a couple of front/back fisheye images. Figure 1 shows an example of such fisheye images obtained during the first month. The images dataset tested is composed of two videos from May 2022 of 10s, from which we extract about 30 images (see Figure 3).

In addition, selected references were LiDAR data from LiDAR HD [13] and SwissSURFACE3D [4]. SwissSURFACE3D is organized in 1 km² tiles, classified in 6 classes (ground in light green, building in red, vegetation in dark green, etc..), with an average density of 15-20 points per m². The average precision is 20 cm in planimetry and 10 cm in altimetry. The tile used, containing the HEIG-VD

a) Front b) Back

Fig. 1: An example of a pair of fisheye images extracted from the GoPro

building, is depicted in Figure 2.

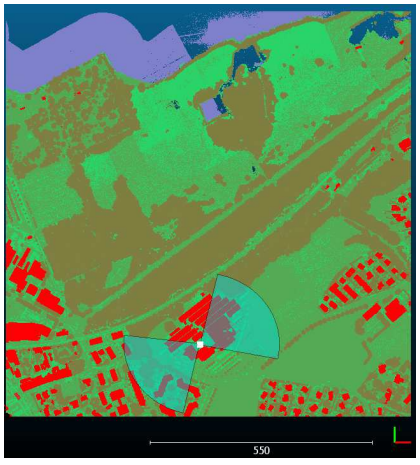


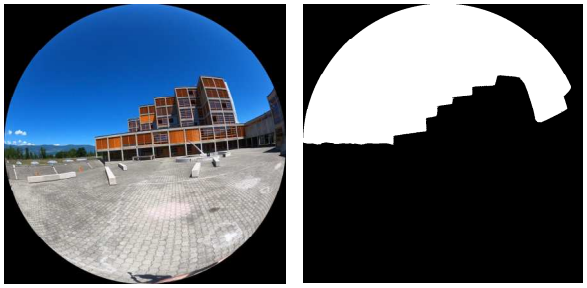
Fig. 2: The aerial view of the LiDAR tile used for tests. The bottom left corner coordinates are 2540000 East, 1181000 North in the MN95 Swiss system coordinate. In white, the camera position from Fig. 1. In blue, the FOV (Field of view).

Now that datasets are defined, our method can be tested, with a first approach consisting in HLC analysis.

IV. PROPOSED METHOD

The first strategy consists in comparing the observation of the camera with a second *virtual* camera, placed and oriented in a 3D model according to sensor measurements. Then a pairing can be done with recognizable objects detected in both pictures, resulting in a series of Ground Control Points. RANSAC and Least Squares applied to collinearity equations on those GCPs enable a new location calculation, enriched by LiDAR data.

A. Real camera image processing



a) Before b) After

Fig. 3: Result of the sky detection, the sky is masked by white pixels.

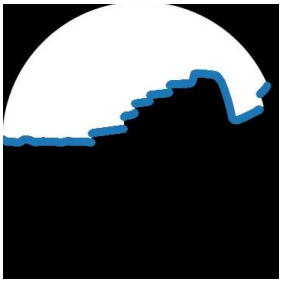


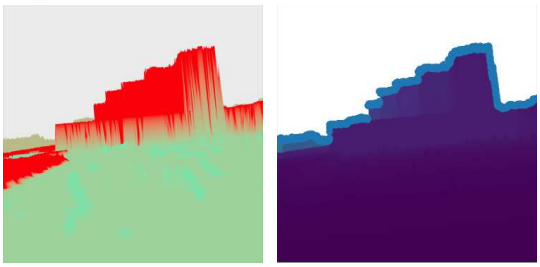
Fig. 4: Horizon line contour extracted from the image, in blue.

With the neural network proposed by [14], a sky segmentation of every fisheye picture can be done, see Figure 3. Having only two classes makes the horizon easy to find by checking the color's variation position, column by column. The result is a list of pixel coordinates [I', J'], see Figure 4.

The next step is to find the corresponding coordinates [X, Y, Z] of those points. This sky segmentation algorithm, while very performant, is limited by the presence of long thin vertical objects, such as street lamps, antennas, or flag poles because they highly affect the horizon lines on the image while being almost invisible in the LiDAR data.

B. Simulated camera image processing

By applying the same extrinsic (location, orientation) and intrinsic (focal length, principal point) parameters to a virtual camera OpenCV can be used to render a view and a



a) Mesh build from LiDAR b) Depth map

Fig. 5: View captured from a virtual camera, placed in the mesh

depth map, see Figure 5.a. and 5.b.

The depth map can be used for the HLC detection since there is a clear separation between ground points, with finite distance D < 1 km and *infinite* for the sky. By applying the

same method used in Figure 4, a second list of coordinates [I, J] is extracted, see Figure 5.b.

with the depth map, it is possible to associate a 3D point with a 2D pixel [I, J] if extrinsic and intrinsic parameters are known. From the equation of collinearity (1), where F is the focal length, R the rotation matrix of the camera, S the camera's position, m a pixel and M the 3D points corresponding to the pixel.

$$m = (i,j) = F - \frac{k^t.F.R.(M-S)}{k^t.R.(M-S)} \text{ with k = [0,0,1]} \quad (1)$$

$$M = [X,Y,Z] = \frac{D}{\|F\|} \times R(m-F) + S \quad (2)$$

The result is a list of LiDAR points coordinates [I, J, X, Y, Z], to create GCPs in the fisheye view, the two preceding results need to be paired to get [I', J', X, Y, Z].

C. Horizon lines matching

The two horizon lines may be affected by many parameters, such as measurement inaccuracies, fisheye distortions, or the difference of FOV (Field of view), which can lead to deformations and deviations between the two HLC. To find the correct pairing, the Dynamic Time Warping or DTW [5], [15] can be used here to associate indices of [I, J] points with corresponding [I', J'] points. DTW takes two temporal signals and associates pairs of indices according to cost minimization of a cumulative distance matrix. It is used in word recognition in sentences for its robustness [5], which are subject to many deformation (stretching, compression, breaks). Because our horizon lines are continuous (from a pixel perspective), the 'X' axis can be used as a "Time" axis for the matching. To improve the result, DTW can perform open-ended alignments [16], [17] to minimize hFOV difference. In Figure 6, the pairings are

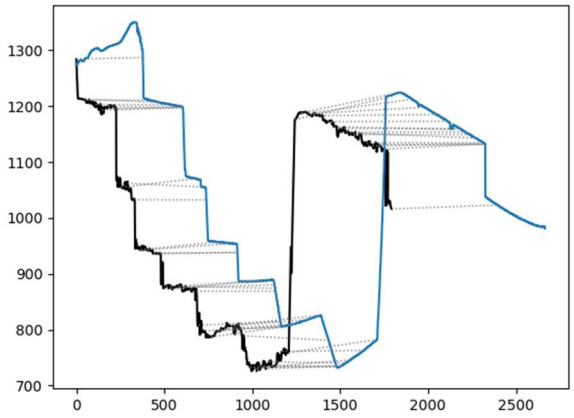


Fig. 6: DTW applied on the two HLC

visible.

After the DTW, between 1500 to 2000 GCPs are made, as shown in Figure 7.

To utilize them as GCPs in our lists of images, filtering is necessary to find for each extracted image, corresponding homologous points.

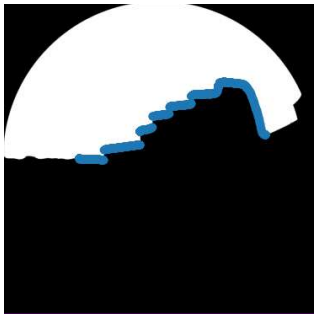


Fig. 7: Ground control points constructed.

D. GCPs extraction

Localizing one image fisheye with distortion was difficult for the Least Square algorithm applied on the collinearity equation to converge to a reasonable solution. Because of that, it was decided to take advantage of the whole video, and by extension, all fisheyes capture, to get a Structure-from-Motion (SfM). For this, we need GCPs that can be found and automatically associated in each image for the calculation of the SfM. It was decided to use points on building corners. The extraction is done with the Ramer-Douglas-Peucker [18], consisting of keeping the farthest point C to a segment [AB], then by iteration, find the same point for segments [AC] and [CB]. To limit the number of iterations, there must be a minimum distance of 20 pixels between the segment and the point (empirically found). It extracts points on the building corner, see Figure 8.

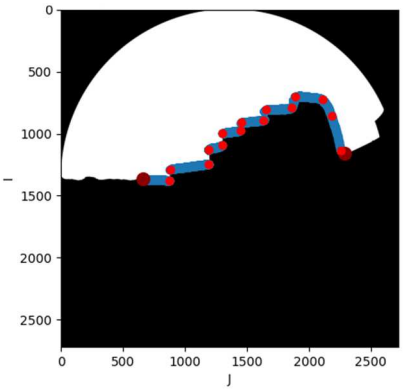


Fig. 8: GCPs extracted from the blue skyline, in red

Once the GCPs are extracted, they can be exported in an SfM software like Metashape [19], as GCPs, on every fisheye's view to recalculate the positions of the cameras.

E. Conclusion on the first method

This first approach, based on Horizon Lines Contour, shows that it is possible to improve a panoramic camera localization automatically with visual information. While the result is better than a simple camera's GNSS measure, it must be improved to be usable on actual surveying missions. A way to improve our result is to find homologue GCPs in each photo. GCPs showing the same element with close terrain coordinates could be assimilated as one GCP, helping the camera alignment.

V. Results

The previous method was applied to a set of 120 fisheye images. The calculation time to extract the points from an image is 2 minutes on average, the most time consuming being to obtain the 3D coordinates. The quality of the position refinement of the camera's GPS is estimated with a distance from the measurement of a GNSS antenna, considered as "True" value. A solid point was measured by a series of GNSS antenna observations, and a "control" video was taken, in making sure it ends on the same point. The difference between the ground truth and the GoPro's GNSS in planimetry is 2m, before the introduction of GCPs, whereas the newly estimated pose had a deviation of 1.2 meters from the antenna. As this result has been obtained

under controlled environments, (clear sky, little sky obstruction, easily recognizable shape and away from walls), the current approach needs to be improved for more challenging scenes. Currently, our GCPs built from sky lines are not robust enough for precise photogrammetric survey but it can be improved by more options to experiment with, as presented in perspectives.

A. Perspectives and current works

Other cameras must be tested to evaluate the robustness of this method. To change the image geometry, an Insta 360 Pro 2 was chosen. Having 6 fisheye lenses and the possibility to work on spherical images (equirectangular projection).

Moreover, the neural networks used detect only the sky, but much information can be obtained with a finer classification. Panoptic segmentations of urban landscape are promising for our project since it allows us to use more references (such as OpenStreetMap, Information System of the Territory in Geneva, etc..) and to benefit from the Swiss LiDAR classification (Ground, Vegetation, building etc..). This should also be a homogeneous distribution of points on the images.

In fact, knowing the class of the LiDAR allows the introduction of weighted measures in the compensation system. For example, a ground control point made from vegetation classified points is more affected by annual variations than a building GCP.

Finally, the horizon visible from the photos may be farther than 1km, (e.g., mountains) because they have an impact on the ground / sky separation. To resolve this, LiDAR data can be visualized and used with a varying density, depending on the distance of observations. This is a Potree architecture [20] that can be tested here, to benefit more information.

REFERENCES

[1] Carreaud, A., Mariani, F., and Gressin, A. (2022). Automating the underground cadastral survey: a processing chain proposal. The International Archives of Photogrammetry, Remote Sensing and Spatial Information Science, 43, 565-570.

[2] Wang, L., Groves, P., & Ziebart, M. (2012). Multi-Constellation GNSS Performance Evaluation for Urban Canyons Using Large Virtual Reality City Models. Journal of Navigation, 65(3), 459-476.

[3] Ng, K. M., Johari, J., Abdullah, S. A. C., Ahmad, A., & Laja, B. N. (2018). Performance evaluation of the RTK-GNSS navigating under different landscape. In 2018 18th International Conference on Control, Automation and Systems (ICCAS) (pp. 1424-1428). IEEE.

[4] SwissTopo (2022) swissSURFACE3D, Office fédéral de topographie swisstopo. Available at: https://www.swisstopo.admin.ch/fr/geodata/height/surface3d.html (Accessed: 6 November 2022).

[5] Rabiner, L., & Juang, B. H. (1993). Fundamentals of speech recognition. Prentice-Hall, Inc.. M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[6] Blettery, E., Fernandes, N., & Gouet-Brunet, V. (2021). How to Spatialize Geographical Iconographic Heritage. In Proceedings of the 3rd Workshop on Structuring and Understanding of Multimedia heritAge Contents (pp. 31-40).

[7] Luhmann, T., Robson, S., Kyle, S., and Boehm, J. (2019). Close-range photogrammetry and 3D imaging. de Gruyter.

[8] Theethira, N.S.P. (2022) 'GeoguessrLSTM'. Available at: https://github.com/Nirvan66/geoguessrLSTM/blob/23ec9d433a5db6a50690faa21bd8039743827803/documentation/CSCI5922_ProjectReport.pdf (Accessed: 6 November 2022).

[9] Tilman, R. (2019) Using Global Localization to Improve Navigation. Available at: https://ai.googleblog.com/2019/02/using-global-localization-to-improve.html (Accessed: 7 November 2022).

[10] Baatz, G., Saurer, O., Köser, K., and Pollefeys, M. (2012). Large scale visual geolocalization of images in mountainous terrain. In European conference on computer vision, pages 517–530.

[11] Baboud, L., Čadík, M., Eisemann, E., and Seidel, H.-P. (2011). Automatic phototo-terrain alignment for the annotation of mountain pictures. In CVPR 2011, pages 41

[12] Produit, T., Tuia, D., Lepetit, V., and Golay, F. (2014). Pose estimation of webshared landscape pictures. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2(3) :127.

[13] IGN (2022) LiDAR HD | Géoservices. Available at: https://geoservices.ign.fr/lidarhd (Accessed: 21 August 2022).

[14] Zou, Z. (2020). Castle in the sky: Dynamic sky replacement and harmonization in videos. arXiv preprint arXiv:2010.11800. I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[15] Berndt, D. J., & Clifford, J. (1994). Using dynamic time warping to find patterns in time series. In KDD workshop (Vol. 10, No. 16, pp. 359-370). R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[16] Giorgino, T. (2009). Computing and visualizing dynamic time warping alignments in R: the dtw package. Journal of statistical Software, 31, 1-24.

[17] Tormene, P., Giorgino, T., Quaglini, S., & Stefanelli, M. (2009). Matching incomplete time series with dynamic time warping: an algorithm and an application to post-stroke rehabilitation. Artificial intelligence in medicine, 45(1), 11-34.

[18] Douglas, D. H. and Peucker, T. K. (1973). Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. Cartographica : the international journal for geographic information and geovisualization, 10(2) :112–122.

[19] Agisoft (2022) 'Agisoft Metashape User Manual - Professional Edition, Version 1.8'.

[20] Schütz, M. (2016). Potree: Rendering large point clouds in web browsers. Technische Universität Wien, Wiedeń.