A semi-automatic tool to georeference historical landscape images

Smapshot is a web-based participatory virtual globe where users can georeference historical images of the landscape by clicking a minimum of six well identifiable correspondence points between the image and a 3D virtual globe. The images database is expected to grow exponentially. In a near future, the work of the web users will no longer be enough.

To tackle this issue, we developed a semi-automatic process to georeference images. The volunteers will be shown only images having a maximum number of neighbour images in the matching graph. These neighbour images are the ones with which they share some overlay. This overlap is detected using the SIFT algorithm in a pairewise matching process.

For an image pair made of a reference image with a known pose and a query image we want to georeference, we extracted the 3D world coordinates of the tie points from a digital elevation model.

Then, by running a perspective-n-point algorithm after having geometrically tested the resulting homography between the two images, we compute the 6 degree of freedom pose, i.e. the position (X,Y,Z) and orientation (azimuth, tilt and roll angles) of the query image. The query image then becomes a reference and the georeference computation can be propagated more deeply in the graph structure.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27204v1 | CC BY 4.0 Open Access | rec: 14 Sep 2018, publ: 14 Sep 2018

A semi-automatic tool to georeference historical landscape images

Nicolas Blanc[1], Timothée Produit[2], and Jens Ingensand[3]

[1,2,3]**University of Applied Sciences Western Switzerland, School of Management and Engineering Vaud, Rte de Cheseaux 1, CH-1400 Yverdon-les-Bains**

Corresponding author:
Nicolas Blanc[1]

Email address: nicolas.blanc1@heig-vd.ch

ABSTRACT

Smapshot is a web-based participatory virtual globe where users can georeference historical images of the landscape by clicking a minimum of six well identifiable correspondence points between the image and a 3D virtual globe. The images database is expected to grow exponentially. In a near future, the work of the web users will no longer be enough.
To tackle this issue, we developed a semi-automatic process to georeference images. The volunteers will be shown only images having a maximum number of neighbour images in the matching graph. These neighbour images are the ones with which they share some overlay. This overlap is detected using the SIFT algorithm in a pairewise matching process.
For an image pair made of a reference image with a known pose and a query image we want to georeference, we extracted the 3D world coordinates of the tie points from a digital elevation model.
Then, by running a perspective-n-point algorithm after having geometrically tested the resulting homography between the two images, we compute the 6 degree of freedom pose, i.e. the position (X,Y,Z) and orientation (azimuth, tilt and roll angles) of the query image. The query image then becomes a reference and the georeference computation can be propagated more deeply in the graph structure.

INTRODUCTION

Smapshot is a web-based participatory virtual globe. It uses crowdsourcing for the georeferencing of historical images. These images come from various sources, such as public archives or private libraries. Volunteers georeference images they know a prior coarse location by picking a minimum of six identifiable correspondence points both in the image and the textured 3D model of the landscape (Produit and Ingensand, 2018). A pose estimation algorithm computes the position and orientation of the camera and the image is then superimposed with the 3D model giving the impression that the observer is looking through a window of the past.

As every other online image databases, the smapshot one is expected to exponentially grow in a near future. The georeferencing of the images currently carried out by volunteer Internet users will no longer be enough. In order to considerably simplify their task, this work introduces a method to geolocate images in a semi-automatic way.

Some studies have shown it is possible to geolocate large quantities of images only based on their comparison with a database of millions of geotagged images (Hays and Efros, 2015; Weyand et al., 2016). Object retrieval technique can also be used for spatial matching between images (Philbin et al., 2007). Baatz et al. (2012) have used a sky segmentation method in mountaineous terrain to extract contour informations and matched them with DEM data using a bag-of-word approach (Fei-Fei and Perona, 2005). Produit et al. (2014) also makes use of a Kalman filter robust skyline matching approach with an initial user input to compute the pose of an image.

In recent years, deep learning methods such as convolutional neural networks (LeCun et al., 2015) have shown that they are also very well suited to classifying or searching images in very large databases, even for geolocation purposes (Weyand et al., 2016).

This work brings an optimization method in volunteers based image georeferencing. They will only

be shown central images in a pairewise matching graph. The geolocation they provide to these central images will then be automatically propagated to other images in the graph in a 4 stages iteration process involving: 1) a pairewise image matching to find images with overlay, 2) the extraction of highest valency images (i.e. central images) on the pairewise resulting graph, 3) the georeferencing of central images by volunteers, 4) the automatic georeferencing of images sharing overlay with the user georeferenced central ones.

METHOD

Building a sample dataset

All images come from the smapshot database and are already geolocated. This geolocation was manually verified prior to the work. Therefore their position and orientation can be used; 1) as a ground truth to validate and evaluate the geolocation computation and 2) to build sets of images having high chances to show some overlay, which is intended to drastically reduce the matching computation time. To build such subsets we have run the DBSCAN clustering algorithm (Ester et al., 1996) on the database.

Computation steps

Local features extraction and images matching

In a first stage, the method used consists by first computing local images descriptors on all images of a sample. Then, we used SIFT and its pairewise image matching algorithm based on the similarities of their descriptor (Lowe, 1999). Since descriptors are fixed dimension vectors, these similarities also characterize the distances between them within the descriptors space. By applying RANSAC (Fischler and Bolles, 1981) to compute the homography matrix for each image pair, we are able to remove some false positive in this first matching results.

Geometrical filter

Finally, in order to eliminate the last false positive results, we developed a geometrical filter. This filter works as follow: we first compute the projection of the frame of the first image onto the second one using the homography matrix computed with RANSAC and then we make some simple tests on this geometry. These tests only retains geometries with: 1) a non-intersecting shape 2) a convex shape 3) a shape surface representing at least 10% of the image.

Pairwise image matching and graph analysis

In a second stage, by looking at the matching results through the graph theory magnifier, i.e. in a graph, images are the nodes and positive matches are represented by edges, we can find the image(s) with the highest valency for each connected component of the graph. These will be the ones that will be subjected to georeferencing by users. As the 6 degree of freedom (DoF) of their pose (3 for their position + 3 for their orientation) is assumed to be validated by an operator, they are tagged as "reference" images. All other images are kept in the background for a geolocation automatic computation and are tagged "query" images. Hence, we keep the ground truth pose values for references images and hide them for the query images.

6DoF pose computation

In a last stage, we compute the 3D position and attitude angles defined by the azimuth, pitch and roll of each image (more precisely of the camera that has taken the picture) using a perspective-n-point (PnP) algorithm. Given an image pair consisting of a reference image and a query image showing some overlay, this algorithm makes use of the 2D query image coordinates of all the tie points and the 3D ground coordinates of these tie points to compute the pose of the query image. The 3D coordinates are prior extracted from a virtual image of a 3D digital elevation model, which is generated by placing and orienting a camera in a virtual globe with the exact same known pose of the reference image. The perspective-n-point algorithm we used makes the most of all the tie points by using a Levenberg–Marquardt approach to find a pose that minimizes reprojection errors of 3D points onto the image. It also uses RANSAC to filter out some outliers.

To verify the results, we compared the 6DoF poses values obtained for a subset of query images with the database ground truth. Once a query image has been successfully geolocalized, it becomes a reference image and can be used to geolocate another one deeper in the graph structure. This process goes on till the last image.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27204v1 | CC BY 4.0 Open Access | rec: 14 Sep 2018, publ: 14 Sep 2018

Evaluation

For each of these stages we evaluated the sensitivity of the results to individual parameter changes based on a confusion matrix (Stehman, 1997). We also checked how our outliers removal filter behaved. Finally, the entire process was run on 22 true pairs of a 27 images cluster. We extracted the 3D coordinates of tie points from the digital elevation model for the first image of the pair (reference) and computed the geolocation of the second (query) image.

RESULTS

Clusters

The DBSCAN algorithm distance parameter was adjusted to 800m to have compact and relevant enough clusters (fig. 1).

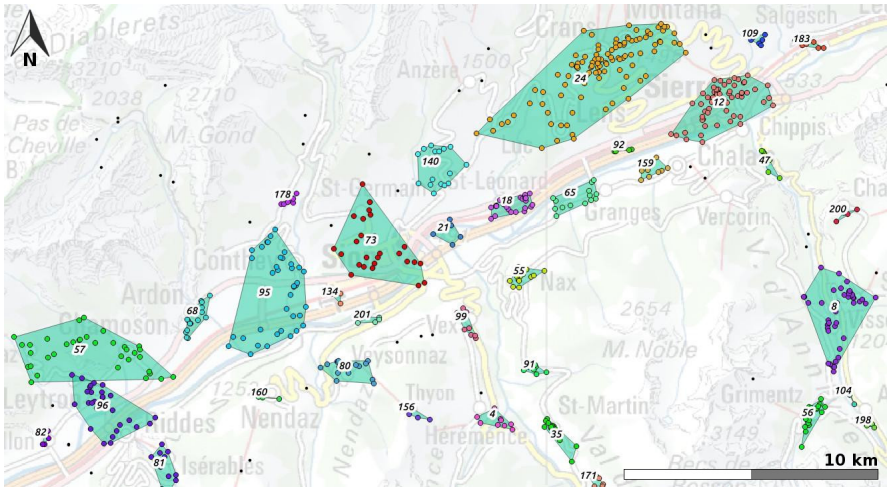
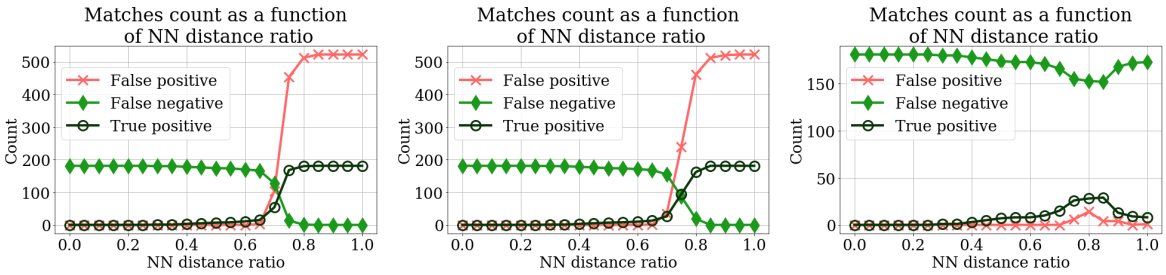


Figure 1. A view of some resulting images clusters.

Matching

Then, by making the nearest neighbour distance ratio of the SIFT matching procedure varying, we deduced its best value equals to 2/3 for our dataset. Applying RANSAC improves the results (fig. 2b), but our geometrical filter was much more better at this task (fig. 2c).



(a) After the simple matching. **(b)** After using RANSAC. **(c)** After applying our filter.

Figure 2. Results showing the improvement of the RANSAC outliers removal and our filter on the matches recall as a function of the nearest neighbor distance ratio used by the SIFT algorithm.

Others parameters of the SIFT and RANSAC algorithms were also tested (not shown here) and led to the values in table 1.

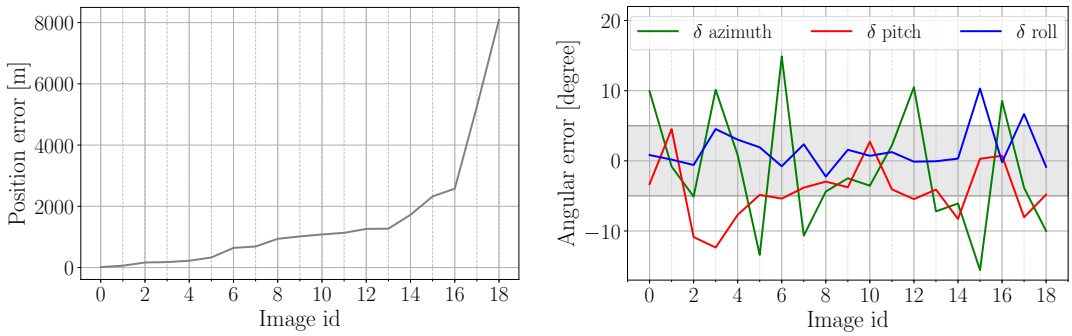
PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27204v1 | CC BY 4.0 Open Access | rec: 14 Sep 2018, publ: 14 Sep 2018

Algorithm	Parameter name	Value used in this study
SIFT	Number of octaves	2
	Contrast threshold	0.032
	Edge threshold	18
	Sigma	1.2
RANSAC	Reprojection error [px]	5
	Max. number of iterations	2000
	Confidence level	0.995

Table 1. Retained values for the SIFT and RANSAC algorithms.

Georeferencing

On the sample cluster, the 22 position results range from 15 m to 8 km (fig. 3a) and 3 were discarded because of a lack in DEM values (no data).



(a) Position errors. 70% of these values are under 2 km.

(b) Angular errors. The [-5, 5] range is greyed out.

Figure 3. Camera pose (ground truth minus computed position) results on 19 query images.

DISCUSSION

Results that are close to the user provided position (which is taken as ground truth) are often obtained when the camera was closed to the part of the landscape captured (fig. 4).

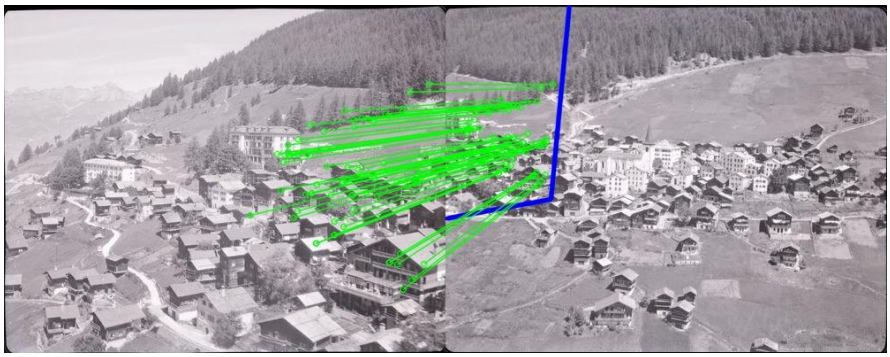


Figure 4. The pose result of the query image (right) is only 15 m close to the position in database. Images source: EPFL, Archives de la construction moderne.

In some cases, a large position or angular shifts may be explained by tie points not homogeneously distributed on the images. In addition to a small overlap, figure 5 shows that tie points on these two images are almost along a vertical line. This may explained why the azimuth is 16°away from the database value.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27204v1 | CC BY 4.0 Open Access | rec: 14 Sep 2018, publ: 14 Sep 2018

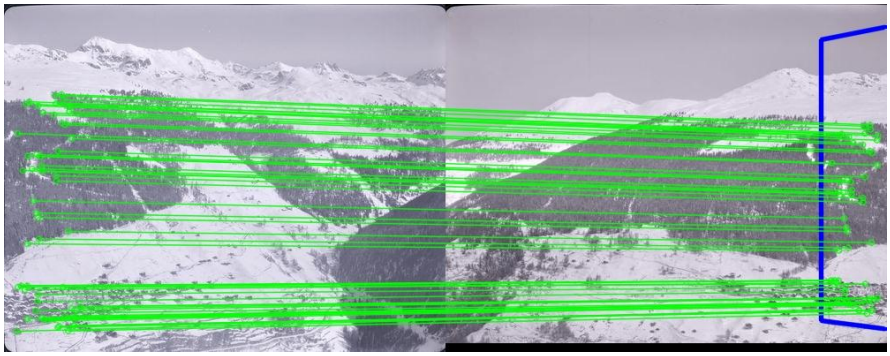
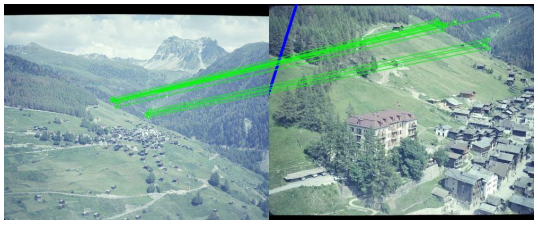
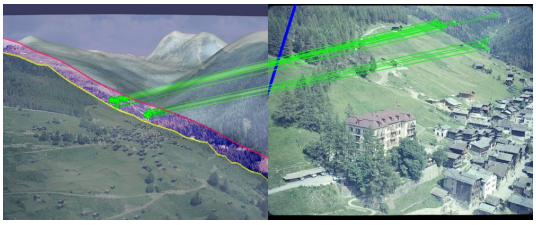


Figure 5. The pose result of the query image (right) is 2.3 km and an azimuth 16°away from the values in database.
Images source: EPFL, Archives de la construction moderne.

Other position shifts are explained by tie points beeing located close to a breakline in the image depth. Figure 6b shows such a case; tie points are in the background on the images, but their 3D coordinates are taken on the forground on the DEM, which is approximately 2 km away from their real location in the background.



(a) Tie points are located near a breakline in the image depth.



(b) When superimposed with the DEM, the breakline between the forground and the background is higher on the DEM than on the image.

Figure 6. The pose of the query image (right) is 1.7 km away from its position in database.
Images source: EPFL, Archives de la construction moderne.

CONCLUSION

Results showed that the pose computation of a query image is quite accurate when some conditions are respected; the accuracy of the reference image pose is crucial. It influences the DEM relative position and 3D coordinates of tie points. The query image pose accuracy cannot be better than the reference one.

Tie points distribution on images is also of great importance as shown by this study. A homogeneous distribution would undoubtedly improve the precision of the computed position.

This would also be the case if we can detect and use GCPs that are already in database to register a query image.

Furthermore, as the Brute-Force matching process is a $O(n^2)$ complex problem, it takes too much time to tackle any new large images dataset with no known a priori positions and where clusters cannot be built. This would need strong computational power. There are still some ways to explore prior to the matching stage, such as the reduction of features space or the avoidance of too much redundancy. In such way, machine learning or bag-of-features techniques may offer other valuable alternative to find overlapping images.

Finally, to integrate this process on the web platform, we will need a two-steps approach: 1) proposing overlapping images to the user and 2) once a picture has been successfully georeferenced by the user, giving him the possibility to see which other images were automatically registered and giving him the opportunity to refine the automatic results. Field of user engagement could for example help building a nice interface to improve and keep the users motivation in this great task.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27204v1 | CC BY 4.0 Open Access | rec: 14 Sep 2018, publ: 14 Sep 2018

REFERENCES

Baatz, G., Saurer, O., Köser, K., and Pollefeys, M. (2012). Large scale visual geo-localization of images in mountainous terrain. In Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., and Schmid, C., editors, *Computer Vision – ECCV 2012*, pages 517–530, Berlin, Heidelberg. Springer Berlin Heidelberg.

Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231.

Fei-Fei, L. and Perona, P. (2005). A bayesian hierarchical model for learning natural scene categories. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 524–531 vol. 2.

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395.

Hays, J. and Efros, A. A. (2015). *Large-Scale Image Geolocalization*, pages 41–62. Springer International Publishing, Cham.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157 vol.2.

Philbin, J., Chum, O., Isard, M., Sivic, J., and Zisserman, A. (2007). Object retrieval with large vocabularies and fast spatial matching. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.

Produit, T. and Ingensand, J. (2018). 3d georeferencing of historical photos by volunteers. In Mansourian, A., Pilesjö, P., Harrie, L., and van Lammeren, R., editors, *Geospatial Technologies for All*, pages 113–128, Cham. Springer International Publishing.

Produit, T., Tuia, D., Lepetit, V., and Golay, F. (2014). Pose estimation of web-shared landscape pictures. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, II-3:127–134.

Stehman, S. V. (1997). Selecting and interpreting measures of thematic classification accuracy. *Remote Sensing of Environment*, 62(1):77 – 89.

Weyand, T., Kostrikov, I., and Philbin, J. (2016). Planet - photo geolocation with convolutional neural networks. In Leibe, B., Matas, J., Sebe, N., and Welling, M., editors, *Computer Vision – ECCV 2016*, pages 37–55, Cham. Springer International Publishing.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27204v1 | CC BY 4.0 Open Access | rec: 14 Sep 2018, publ: 14 Sep 2018