A geograhic data gathering system for image geolocalization refining

Bernard Semaan, Myriam Servières, Guillaume Moreau, B. Chebaro

A GEOGRAPHIC DATA GATHERING SYSTEM FOR IMAGE GEOLOCALIZATION REFINING

B. Semaan[a,b], M. Servières[a], G. Moreau[a], B. Chebaro[b]

[a] UMR 1563 AAU-CRENAU - (bernard.semaan, myriam.servieres, guillaume.moreau)@crenau.archi.fr
[b] Lebanese University - (bchebaro)@ul.edu.lb

KEY WORDS: Image Geolocalization Refining, Building Detection, Semantic Data Extraction, Geographic Data Collection, Multi-Source Data Analysis

ABSTRACT:

Image geolocalization has become an important research field during the last decade. This field is divided into two main sections. The first is image geolocalization that is used to find out which country, region or city the image belongs to. The second one is refining image localization for uses that require more accuracy such as augmented reality and three dimensional environment reconstruction using images. In this paper we present a processing chain that gathers geographic data from several sources in order to deliver a better geolocalization than the GPS one of an image and precise camera pose parameters. In order to do so, we use multiple types of data. Among this information some are visible in the image and are extracted using image processing, other types of data can be extracted from image file headers or online image sharing platforms related information. Extracted information elements will not be expressive enough if they remain disconnected. We show that grouping these information elements helps finding the best geolocalization of the image.

1. INTRODUCTION

At present day, Internet is becoming fundamental in our daily life. People have been using network based services for navigation along with GPS sensor for the last decade. This need of navigation pushed researchers to develop methods to help users understand their surroundings using several navigation equipments and methods like augmented reality (Taketomi et al., 2011). These applications need to be fed with information about building shapes and other information like semantic ones as for example stores or building functionality. These details may be given by the cities administrators or mapping services. Thus, city managers and mapping services are creating digitalized duplicates of actual cities. These duplicates include mainly buildings, routes, addresses and other geographic information like infrastructure. A geographic information system (GIS) database can rarely be up to date in an active city. Buildings are demolished or constructed and not always updated in GIS. Constructions enclosing stores may change colors and store names often as well.

Technology widely available among users would help updating these geographic systems. In fact, collaborative approaches are becoming increasingly used following the Volunteered geographic information (VGI) movement (Goodchild, 2007). Websites such as OpenStreetMap.org and maps.google.com allow any user to add information into their database in order to make updates easier and more efficient. On the other side, many users publish their pictures on online image sharing websites like flickr, picasa or facebook. These photos may represent places in the city and can help, when well explored, updating geographic databases. All of the above encouraged us to propose and implement a system that gathers information from most of the existing sources. This system would allow us to precisely geolocalize images in an urban environment finding the camera pose (position and orientation) when shooting a photo. Our system considers a single image taken from ground level as if a tourist was taking a picture of the city. Actual automatic building detection method can only process images with a single building with non tilted roofs. The used image should show at least two facades of the building for better image geolocalization and orientation detection. In the following sections we first present a brief overview of existing geolocalization refining methods, we then introduce in Section 3 our proposed system and explain its different layers. In Section 4 we present the results obtained with a totally automated process and finally we propose perspectives and future work in Section 5 and conclude in Section 6.

2. RELATED WORK

We present in this section different geolocalization approaches using images visual part or embedded semantic data. In (Bioret et al., 2008), the authors compare a single 2D image to a 2D GIS map in order to find the camera pose. Each image is manually annotated to extract buildings corners, and roads are then pointed out. The corners information and the vanishing points are then used to reconstruct the shape of the buildings that are visible in the image. The results are queried in the GIS thanks to angles between buildings and roads relative positions to find out the pose information. A similar method implemented by (Chu et al., 2014), aims to automate even more the process. The authors start by correcting the tilt angle using the vertical vanishing point. They find out the horizontal vanishing points using segments extracted by the LSD detector (Grompone Von Gioi et al., 2010). The vertical edges are then detected and the following of the process is similar to (Bioret et al., 2008) by comparing the image to a two dimensional Map. Both of these methods use limited zone maps. In fact, searching angles between buildings at city scale gives too much answers and also leads to a lot of false positive (irrelevantly matched buildings). Thus those methods limit the search area around a base location detected by GPS sensors. Other methods like (Arth et al., 2015), compare detected segments with 2.5D maps including building elevations. This method allows simultaneous localization and mapping when the application is installed on a smartphone.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-IV-4-W3-63-2017

Some other techniques compare a query image with accurately geolocalized images or textures of three dimensional maps. We can find such methods in (Ventura et al., 2014) where authors created a mobile phone application that directly sends the first two images to reconstruct the 3D environment on a server. The result will then be compared with a 3D point cloud database including building textures. The point cloud is built using accurate GPS equipments and image acquisition technologies. (Zamir and Shah, 2010) and (Yazawa et al., 2009) uses image matching with street view images to find the best geolocalization. (Zamir and Shah, 2010) method downloads images from Google Street View and compares SIFT descriptors (Lowe, 2004) with the ones in the query image. SIFT descriptors are indexed using a tree that will be queried using the nearest-neighbor method. Zamir worked also on semantic data in (Zamir et al., 2011). The aim of this work is to identify commercial entities in the street view imagery. The method is based on textual information extraction which is challenging due to the variety and complexity of the images. (Yazawa et al., 2009) use SURF descriptors to find a match with already known panoramic images.

Urban furnitures may also help finding geolocalization or camera pose. (Soheilian and Brédif, 2014) works on three dimensional reconstruction of road signs using two dimensional images. They start with background knowledge about road signs form (e.g. square, triangle or circular), then re-projects the image to find out the pose of the camera with respect to the sign. We think other indexed urban furniture may be used too, specially when having metric information, we can estimate the distance separating the camera from the object we have detected, as was done for example by (Antigny et al., 2016). We thus aim in the following section to regroup some of the above methods in order to extract a better geolocalization than using each method alone.

3. DATA COLLECTION PROCESS

In the previous section we have presented some existing methods for urban images geolocalization. Unluckily none of those can work in a fully automatic and unsupervised way for any image. We thus present a system aiming at taking a 2D image with prior uncertain GPS position and refine its pose. This system combines visual information existing in the image with semantic information extracted from the image. Some other information may also be available such as camera intrinsic parameters saved in EXIF headers or users comments or tags assigned to the image retrieved from social networks. We compare these information elements to collaborative databases (as OpenStreetMap for example or in our case Nominatim (OpenStreetMap, 2017)) in order to get the most recent geographic information available.

Image geolocalization accuracy widely depends on the applications usage. GPS sensors available in most smartphones and common electronic devices have an average error of 4.9 m under open sky as mentioned in (National Coordination Office for Space-Based Positioning, Navigation, and Timing, 2017). Unfortunately this average error increases when used in urban environments as satellite signals may be reflected by buildings or blocked by underground sealing. Thereby we propose a system that uses information available in the image in order to find a better location beyond the GPS sensors precision. Our system, presented in Figure 1, is divided into three main layers: data retrieval and preprocessing layer, features extraction layer and decision making and validation layer.

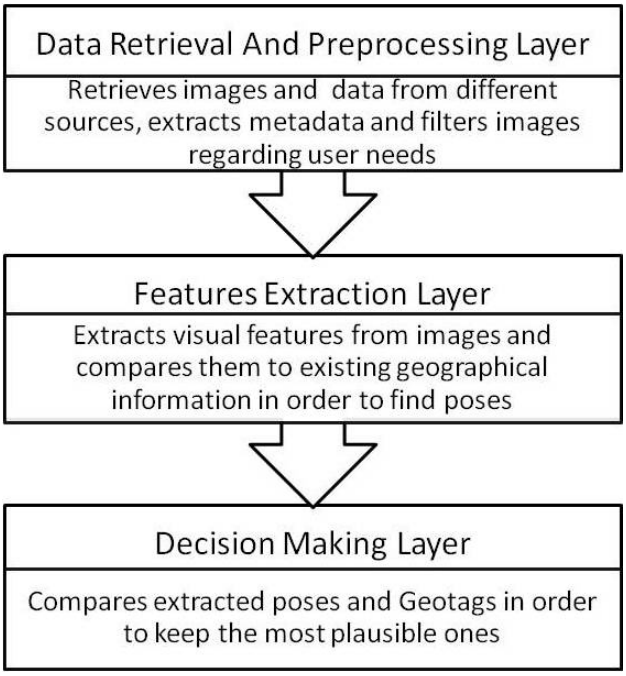


Figure 1. Proposed system layers.

Every layer in our system consists of several modules. Modules may be omitted, added or modified in order to get better results by implementing new methods. In this section's schemas, square represent modules, ovals represent data and the cylinder represents a database.

3.1 Data Retrieval And Preprocessing Layer

The first module goal in our data retrieval and preprocessing layer is to retrieve images. Images retrieved from online databases, as shown in Figure 2. The images should follow user requirements, ie. they may be related to a specific time (before or after a certain date) and/or a specific search zone (e.g. a specific city, or a delimited zone by a rectangular boundary box). Images retrieved from online sources should be also be retrieved with their metadata. Additional information like hashtags and comments may be retrieved too because it may evince geographical information. We included in our first layer an EXIF data extractor module that will provide information about camera parameters and some uncertain but helpful GPS data used to restrict the search zone for future image geolocalization. A filtering module using these EXIF information will then check that the image meets the user requirements. This results in a reduced map that encloses a smaller number of buildings. In fact the method we use in the following layer requires a map to compare the building shape with the one on the map. It would be impossible to find a building using only its two dimensional shape in the world map. Contrariwise, a reduced map (100×100m in our case) makes it possible to find a limited number of buildings and camera poses depending on the building shape. Thus the result of the first layer "data retrieval and preprocessing" is a list of images, each with a corresponding reduced map that includes the image prior location and its surrounding buildings.

3.2 Features Extraction Layer

The second layer of our system named "Features extraction" is presented in Figure 3. In this layer we extract several types of information that will be later combined for better pose estimation.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-IV-4-W3-63-2017 | © Authors 2017. CC BY 4.0 License.

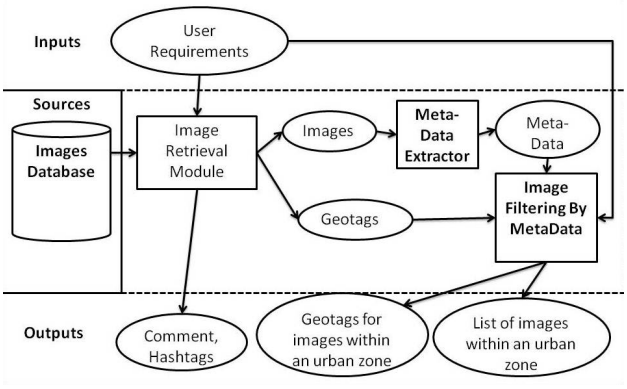


Figure 2. Data retrieval and preprocessing.

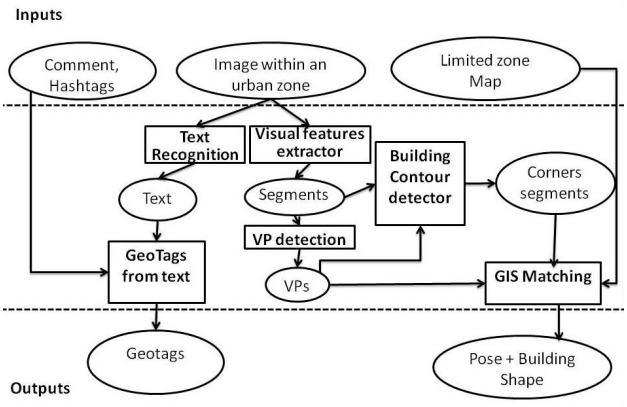


Figure 3. Features extraction layer.

We think that the more we have information, the chance of finding automatically the correct camera pose increases if they are concordant. In fact, getting data from different sources will help filling the lack of information of some of theses sources. Thus we include in our system several data type extractors. Some of these methods already existed and the challenge was to integrate them in our system in an efficient way. This layer deals with two types of data. The first type of data is the geometric one. It includes the information we can get out of the building shape or any data that can be compared to the two dimensional GIS, another photo or a three dimensional model of the city. The second type of data is the semantic information. Some of this information may be embedded in the visual part of the image. The remaining ones can be found in the image metadata.

Geometric Data

In the geometric part, we start by retrieving the segments in the image. We use for this purpose the Line Segment Detector (LSD) (Grompone Von Gioi et al., 2010) that is a segment detector giving subpixel accurate results. The extracted segments are then used to determine the vanishing points in the image. Man made scenes, especially those with buildings usually have at least one vanishing point.Vanishing points are detected using the approach in (Rother, 2002), which finds the average point where segments will converge and is used for horizontal and vertical vanishing points detection. The segments considered as vertical in a building are defined by a bundle of specific segment slopes under the assumtion that images were shot in a natural position. This bundle groups less segments than the horizontal segments one. These segments extracted from non-building objects such as cars, trees, pedestrians or other urban furniture are considered as spurious segments. These affect vanishing points detection in some cases.

The next module uses both segments and vanishing points to detect a building in a single two dimensional image. To this end, we first need to find long segments that may represent a building edge. As shown in Figure 4 the LSD segments detection method returns short segments from the image and only some of them are useful. We chain small segments when they have a sufficiently close slope and small separation distance to create bigger segments. Finally unchained segments are filtered by length. Small segments may represent far buildings in the background or other objects in an urban environment. Long segments represent buildings edges, buildings tops or balconies. An example result is presented in Figure 5. The initial image (with the final result) can be seen on Figure 6.

Figure 4. LSD segments detection.

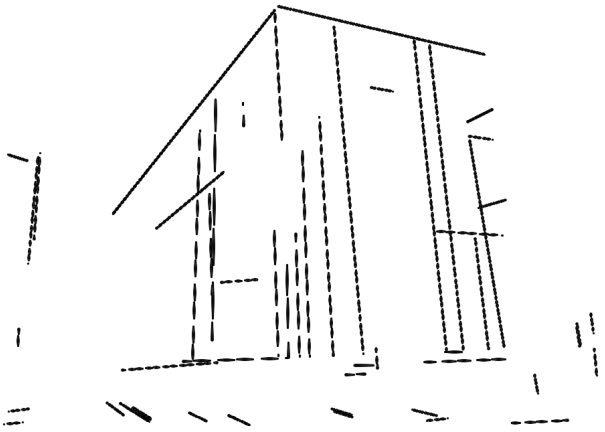


Figure 5. LSD segments after grouping.

Following to our semantic filtering presented above, remaining images in the urban zone always includes buildings. Some of these buildings may be too small to be detected by our method. Thus we perform an image filtering regarding the number of segments, ten segments at least in our case. The ten segments should have a minimal length that corresponds to 4% of the image height in our case (4% was fixed experimentally). Afterwards we filter

electric wires because they affect the building detection method. Electric wires and power poles are characterized being often surrounded by the same color or texture. We thus filter segments surrounded by the same color from both sides. This filtering may affect some unnecessary building segments that do not belong to the building envelope edges.

Another module finds the building contour polygon. We briefly present this module in this paper. We found that most of existing building detection techniques that uses two dimensional images are based on color distribution and clusters like (Saxena et al., 2009). Some other techniques use more than one image or a video sequence to make 3D reconstruction using stereoscopy (Bioret et al., 2010). In our building detection process we assume that the processed image only contains one large building that covers most of the picture. Segments belonging to small buildings have been filtered out in the grouping process described above. Remaining segments are then used to build the contour polygon of the building. We first consider the horizontal segments and start by only preserving the highest segments of the image (i.e. those that are in the upper part of the image). These segments represent the roof edges. The remaining segments are then chained together if they are collinear, and extended until they intersect an other segment when they belong to two different facades. Finally we add the vertical segments by joining the building corners found from horizontal edges with the floor. Thus the result of this method is the convex envelope of the building given by the vertical and horizontal extreme edges, see Figure 6.

Figure 6. Building convex envelope detection.

By using a method based only on segments, we found that it is robust against weather changes, textures variations in a building, and some occlusion problems existing in urban environments, as demonstrated in Figure 7. Unfortunately, the method can only detect one single building in an image. This method is not functional with complex architectures containing arcs and inclined walls. Finally buildings with non flat roofs may be problematic too. We will improve this building detection method while respecting its simplicity in order to find a solution for non flat roofs for example.

Having the building corners and the vanishing points, we can proceed to find the building shape and compare it with the 2D GIS data. The reduced map extracted using the prior geolocation will be used for this operation as well. We use the method implemented by (Bioret et al., 2008). The first part of this method takes

Figure 7. Tree occlusion overtaking.

the corners location on the picture and the vanishing points in order to find the angle between every two facades and the length ratio between them. Angles and ratio between facades will reveal the building two dimensional shape. The second part is then comparing the deduced building shape and the reduced map we have. The method will yet return the possible poses of the camera. Only one pose should be the correct one. An example is presented in Figure 8, only the yellow pose is the correct one.

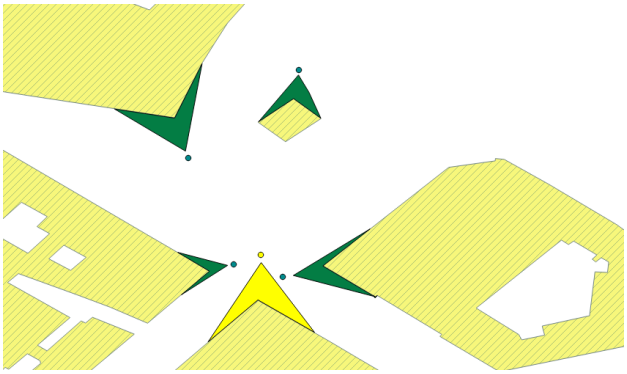


Figure 8. Map resulting from the GIS matching module.

Semantic Data

On the other hand, an image may also contain semantic information. Extracting semantic information like text can disclose geographic information. In fact, a restaurant logo text such as "McDonald's", "Burger King", etc., can be compared with the list of restaurants in the city. It is the same for any text in the picture that tells a store front name, road names, buildings number, etc. We use the Google Vision API (Google, 2017) in order to detect text in the image.The API returns the detected text in the image. We believe that comparing the detected text with a collaborative mapping project will provide up to date information about city places. In our system, we use the OpenStreetMap.org website and its Nominatim API (OpenStreetMap, 2017) that returns a list of geolocations and the building shapes when available.

3.3 Decision Making And Validation Layer

Finally, the "Decision making and validation" layer purpose is to reduce the number of candidate poses retrieved by the previous layer and try to make sure these are valid. Some images provide

semantic information only based geolocations, some others return geometric only based geolocations, the best results are found when an image returns both. When only geometric information is returned we take the surrounding poses of the rough geolocation extracted from the EXIF in a 50 meters radius zone. We experimentally choose 50m to be above the about 5m position uncertainty given by the GPS in good conditions. When only semantic information is returned we use this information. When both information elements are available we keep the poses among the 50m radius zone surrounding every semantic location. Finally, when orientation is available from the GIS matching module, we use the following methods to validate the poses: for testing, we do have the ground truth pose of our images, we thus compare the orientation and the geolocalisation with the available data. When using other image sources, we download views from Google Street View to match similitude with the image and thus validate the pose.

4. SYSTEM RESULTS

In this section we present our system results. Results have been manually evaluated and divided into 5 categories presented as (N) No detection, (W) Wrong Detection, (P) Partly Detected, (A) Detection with additional solutions, (S) Successful detection. Some of these categories may not exist for all methods. In the following we explain the categories used for each method then present results and their discussion. We performed our tests on 19 different images. Results are presented in Figure 12.

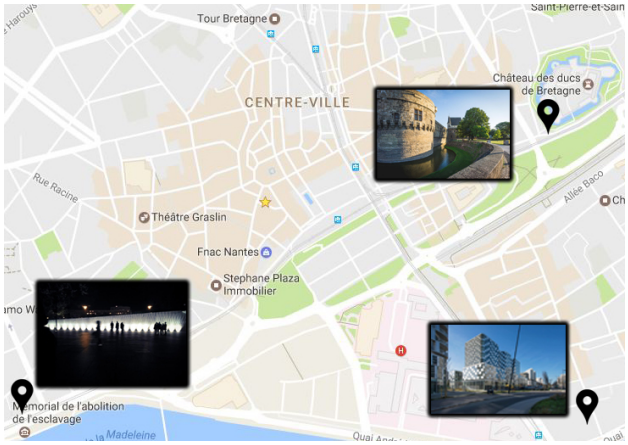


Figure 9. Map showing some images retrieved from flickr.

Image Type Detection: In the image type method we try to find the image content type to make sure we are dealing with an outdoor urban image with buildings content. We ask the Google API (Google, 2017) to return five tags describing the image content weighted with a score that fits each one's validity. If first 5 detected types are not among the list of keywords revealing a building, we consider that the image is not usable for the rest of our system. If one of the detected types is among the list of keywords revealing a building with a score higher than 75% we consider this image usable for the rest of our system. Some results are presented in Figure 10 for the images we have acquired ourselves and Figure 11 for images we have downloaded from flickr using our system. All of our 19 photos where successfully detected as buildings or store fronts.

Downloads from flickr in an urban zone contains both images containing buildings and some with none as shown in Figure 9, the API returns no keywords about buildings in this case.

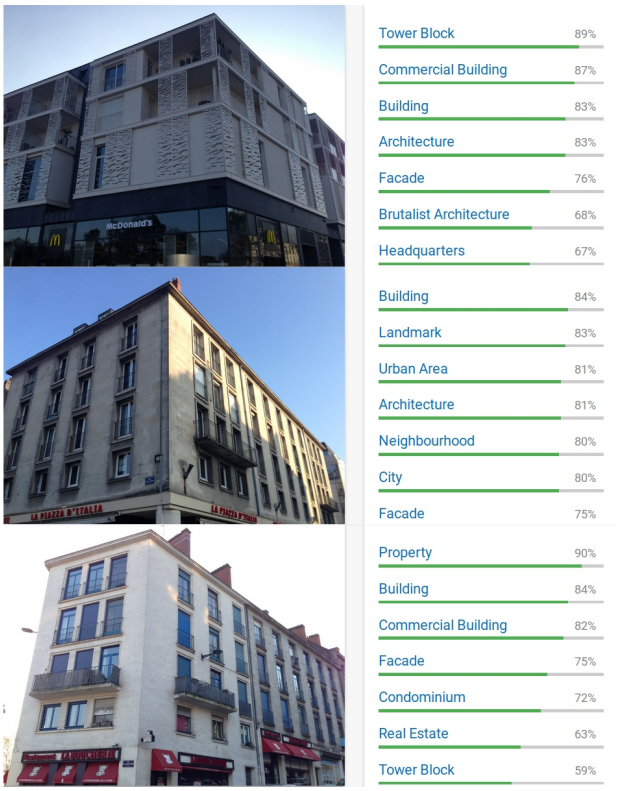


Figure 10. Image type detection samples (Our images).

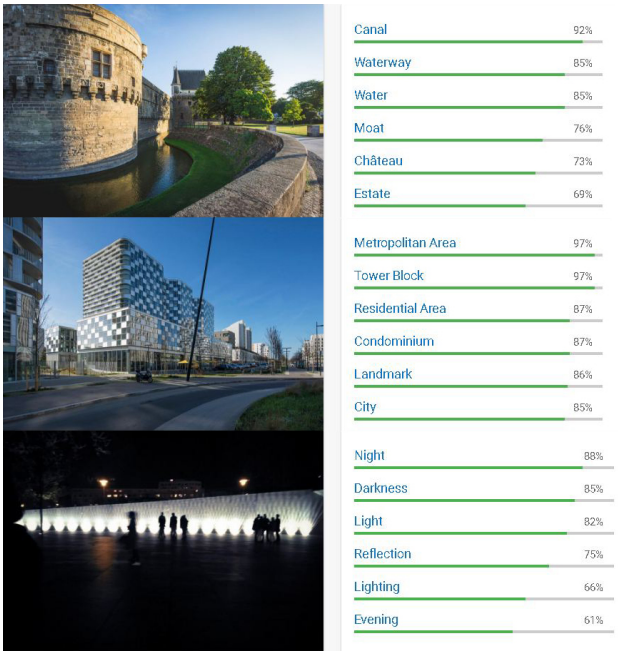


Figure 11. Image type detection samples (Images from flickr).

Text Detection: In the text detection method, (N) stands for no text was detected, (W) means that text was detected but not relevant to the image content, (P) means a text was detected with some missing letters, (A) means a relevant and non relevant text detected which may confuse the geolocation detection in some cases and (S) means only relevant text was successfully detected. Results are presented in Figure 12.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-IV-4-W3-63-2017 | © Authors 2017. CC BY 4.0 License.

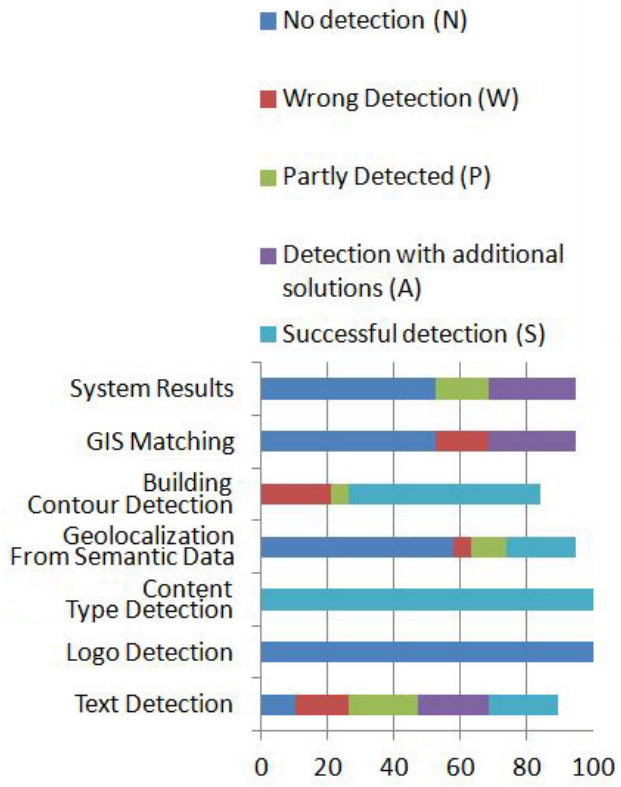


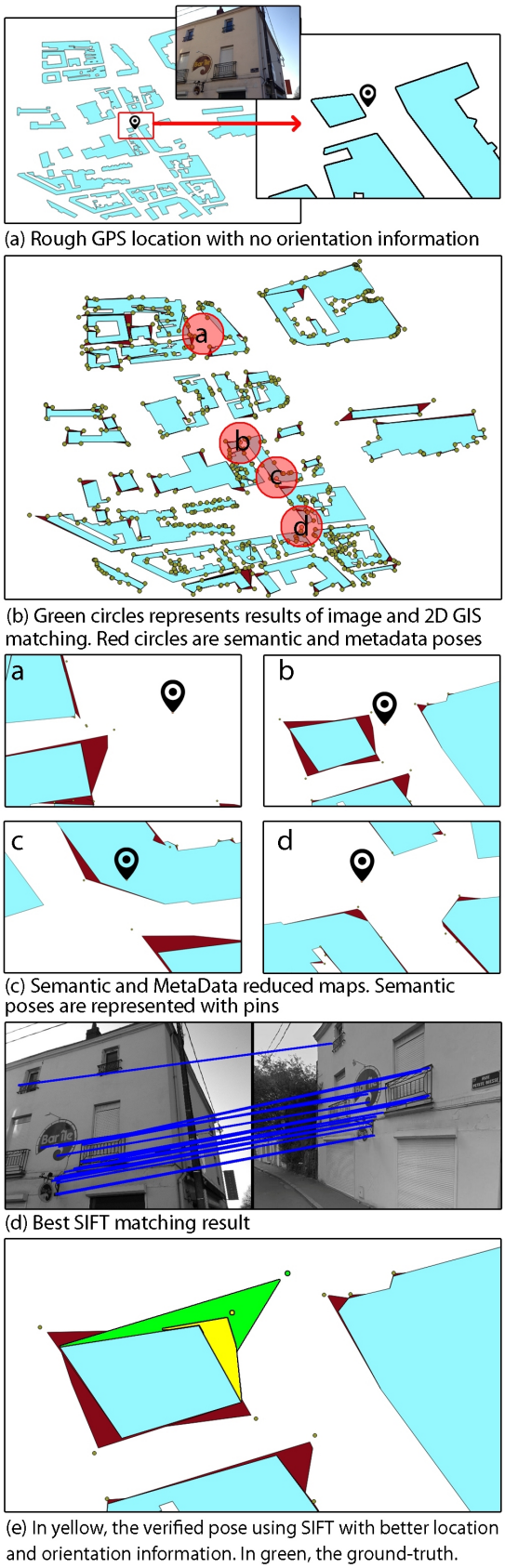
Figure 12. Detection results over our 19 test images.

Geolocalization from semantic information: Finally we regroup all the semantic information mentioned above and send them to the Nominatim API. (N) in this method means no GPS data found using the combination of semantic information, (W) means wrong locations found from semantic information, (P) means zone detected using semantic information but not building, (A) many buildings where detected and (S) means building detected with building polygon shape using some or all of the semantic information. Results are shown in Figure 12.

Building contour detection: In this method we visually compare the building detection and present the results in Figure 12 as following: (N) segments detected not sufficient for building detection, (W) wrong detection of building contour, (P) part of the building facades are detected and (S) visible building facades successfully detected.

Geolocalization using GIS matching (Bioret et al., 2008): This method shows the combination of building contour detection and geolocalization using (Bioret et al., 2008). Here, (N) means no poses found, (W) Wrong poses found, (A) many poses detected, (S) only the correct pose is detected. See results in Figure 12.

Geolocalizations for the complete system : This evaluation shows in Figure 12 the results from the entire system chain. (N) means no locations found with good orientation, (W) wrong locations and orientations found, (P) good location is among the found ones with wrong or no orientation information, (A) many poses detected and one of them is correct, (S) correct pose is detected using geometric method and semantic filtering. We can find that, at the end of the complete process chain, 50% of the results are not detected. Yet, the other 50% are either detected with some other false positive solutions or needs more precision. To find the correct pose, two solutions are possible. First one, is a manual inspection by an expert. Second one, is comparing the obtained solutions with Google Street View images downloaded using the list of poses we have. In Figure 13 we show the evolution of



(a) Rough GPS location with no orientation information

(b) Green circles represents results of image and 2D GIS matching. Red circles are semantic and metadata poses

(c) Semantic and MetaData reduced maps. Semantic poses are represented with pins

(d) Best SIFT matching result

(e) In yellow, the verified pose using SIFT with better location and orientation information. In green, the ground-truth.

Figure 13. Case study example

the location all along our processing chain. In (a) we can see pinned on the map the rough GPS location that is given by the smartphone that acquired the photo. This GPS location has no orientation information. In (b) we present the results of 2D GIS and image matching. 644 poses where returned in the chosen zone, they are represented as small green circles. Red circles represent the GPS location returned by the semantic and metadata information filtering surrounded by a 50m radius circle. Thus the next 4 mini-maps in Figure 13(c) are the ones kept after semantic filtering, only 18 poses remain among the previous 644. All these poses are sent to Google Street View API and we download images for the corresponding pose. The best SIFT matching result is then presented in (d). The last map shows the location found by the full automatic processing chain in yellow, ground-truth location and orientation are in green. Finally we compare the ground-truth with the initial rough pose and with the pose computed from our process. The rough GPS location was 7.3m away from the ground-truth location and no orientation pose was available. The processing chain result is only 3.98m from the ground-truth location and orientation is almost the same but with a wider viewing angle.

5. DISCUSSION AND FUTURE WORK

We have presented in the previous section our output results of several modules and the ones that combines all modules together. We discuss in this section some results and how they can be improved in future work.

We start by looking into the image type detection, we find that all images presented in this article are detected as property, architecture, building, house or facade. In fact, we use our own images for buildings in the city to present those results results in order to compare the results of every module with the known ground truth pose to assess the whole retrieval chain. Yet, we have tested this API with several images and we have found that it returns quite good results on image content.

When the text is clear, the text detection method detects it easily, but on the other side, if there is some fonts or colors mix in the same store facade, it makes text detection in urban environments much more complicated. If we would like to do a better text recognition, we should first perform a text detection and only send to the recognition process the part of the image that includes text. We then should introduce an autocomplete or a correction API for improving text quality. This would help to complete missing or occluded letters when detecting text. We then explore the geolocalization of images using only semantic data. Two causes affect the results in this module. The first cause is that some of the semantic data detected previously are wrong or not relevant for a good geolocation detection (e.g. an adds banner may present a text that is not relevant to the location, text detection may find text where none exists). The second cause is when the text sent to the API with is relevant but the place is not available on the Nominatim Places API, thus no results can be returned. In collaborative approaches, the only solution is to wait for someone to add the missing place or to insert it which beyons the scope of this work.

The building contour detection module will be improved to deal with several building on a same image and to handle special shaped buildings. We can likewise say that the method can be improved by detecting tilted roofs and improving vanishing points detection by filtering non building segments or zones. Concerning the (Bioret et al., 2008) method, it can be improved by adding routes information to the input information. In fact, the method already takes in consideration the roads and buildings relation, but, present automatic detection method does not return information about roads.

The final module aim is to combine results and return the available poses. This module will be completed with an automatic pose verification by automatically comparing Google street views to the base image. Some examples are presented in 14, our photos on the left side of the image and Google Street View photos on the right side of the image. We can find that many SIFT descriptors where matched in both photos. Thereby, the pose used to download the Google Street View image that has the biggest number of matching descriptors will be retained as the most accurate one.

Finally, we need to implement a weighted decision model to take into account the importance of every type of data, especially if the case of contradictory information.

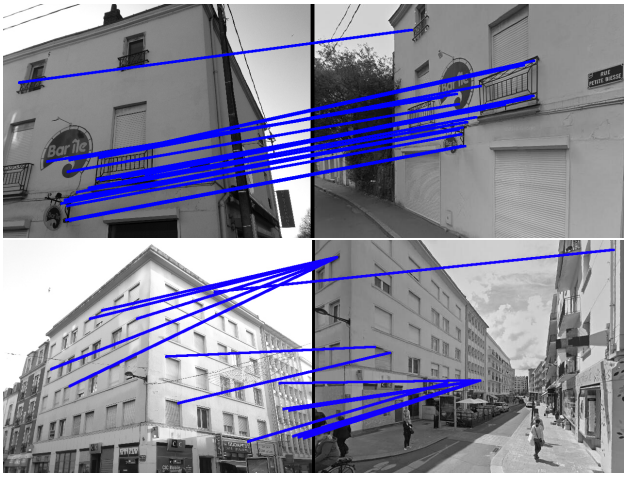


Figure 14. Geolocalization detection using the whole system

6. CONCLUSION

In this paper we have presented a system for image geolocalization refining. The system gathers and crosses different types of information from various sources using several modules. We believe results are promising for a totally automatic pose detection process. Even though some modules need improvement, we proved that gathering simple and accessible information can give quite good results. We have also show that exploring collaborative platforms downloading accurate data may reveal useful geolocalization information.

REFERENCES

Antigny, N., Servières, M. and Renaudin, V., 2016. Hybrid visual and inertial position and orientation estimation based on known urban 3D models. In: *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, IEEE, Madrid, pp. 1–8.

Arth, C., Pirchheim, C., Ventura, J., Schmalstieg, D. and Lepetit, V., 2015. Instant Outdoor Localization and SLAM Initialization from 2.5D Maps. *IEEE Transactions on Visualization and Computer Graphics* 21(11), pp. 1309–1318.

Bioret, N., Moreau, G. and Servières, M., 2010. Towards outdoor localization from gis data and 3d content extracted from videos. In: *Industrial Electronics (ISIE), 2010 IEEE International Symposium on*, IEEE, pp. 3613–3618.

Bioret, N., Servières, M. and Moreau, G., 2008. Outdoor localization based on image/gis correspondence using a simple 2d building layer. In: *2nd International Workshop on Mobile Geospatial Augmented Reality, LNGC, Québec, Canada*.

Chu, H., Gallagher, A. and Chen, T., 2014. GPS Refinement and Camera Orientation Estimation from a Single Image and a 2D Map. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, IEEE, pp. 171–178.

Goodchild, M. F., 2007. Citizens as sensors: The world of volunteered geography. *GeoJournal* 69(4), pp. 211–221.

Google, 2017. Google Vision API. `https://cloud.google.com/vision/`. Accessed: 2017-02-22.

Grompone Von Gioi, R., Jakubowicz, J., Morel, J. M. and Randall, G., 2010. LSD: A fast line segment detector with a false detection control. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32, pp. 722–732.

Lowe, D. G., 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60(2), pp. 91–110.

National Coordination Office for Space-Based Positioning, Navigation, and Timing, 2017. GPS Accuracy. `http://www.gps.gov/systems/gps/performance/accuracy/`. Accessed: 2017-02-22.

OpenStreetMap, 2017. Nominatim.OpenStreetMap.

Rother, C., 2002. A new approach to vanishing point detection in architectural environments. *Image and Vision Computing* 20(9), pp. 647–655.

Saxena, A., Sun, M. and Ng, A. Y., 2009. Make3D : Learning 3D Scene Structure from a Single Still Image. *Pattern Analysis and Machine Intelligence, IEEE Transactions* 31.5, pp. 824–840.

Soheilian, B. and Brédif, M., 2014. Multi-view 3D circular target reconstruction with uncertainty analysis. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* II-3(September), pp. 143–148.

Taketomi, T., Sato, T. and Yokoya, N., 2011. Real-time and accurate extrinsic camera parameter estimation using feature landmark database for augmented reality. *Computers & Graphics* 35(4), pp. 768–777.

Ventura, J., Arth, C., Reitmayr, G. and Schmalstieg, D., 2014. Global Localization from Monocular SLAM on a Mobile Phone. *Visualization and Computer Graphics, IEEE Transactions* 20.4, pp. 531–539.

Yazawa, N., Uchiyama, H., Saito, H., Servières, M. and Moreau, G., 2009. Image based view localization system retrieving from a panorama database by surf. In: *Proc. IAPR Conf. on Machine Vision Applications (MVA)*, pp. 118–121.

Zamir, A. R. and Shah, M., 2010. Accurate image localization based on google maps street view. In: *European Conference on Computer Vision*, Springer, pp. 255–268.

Zamir, A. R., Darino, A. and Shah, M., 2011. Street view challenge: Identification of commercial entities in street view imagery. In: *Machine Learning and Applications and Workshops (ICMLA), 2011 10th International Conference on*, Vol. 2, IEEE, pp. 380–383.