

VISUAL CONTENT LAYER FOR SCALABLE OBJECT RECOGNITION IN URBAN IMAGE DATABASES

Xavier Baró, Sergio Escalera, Petia Radeva, and Jordi Vitrià

Computer Vision Center, Campus UAB, Edifici O, 08193, Bellaterra, Barcelona, Spain.
Dept. Matemàtica Aplicada i Anàlisi, UB, Gran Via 585, 08007, Barcelona, Spain.

ABSTRACT

Rich online map interaction represents a useful tool to get multimedia information related to physical places. With this type of systems, users can automatically compute the optimal route for a trip or to look for entertainment places or hotels near their actual position. Standard maps are defined as a fusion of layers, where each one contains specific data such height, streets, or a particular business location. In this paper we propose the construction of a visual content layer which describes the visual appearance of geographic locations in a city. We captured, by means of a Mobile Mapping system, a huge set of georeferenced images ($> 500K$) which cover the whole city of Barcelona. For each image, hundreds of region descriptions are computed off-line and described as a hash code. This allows an efficient and scalable way of accessing maps by visual content.

Index Terms— Visual content, object recognition, map layer, computer vision.

1. INTRODUCTION

With the exponential growth of multimedia content in Internet, powerful and robust data mining mechanisms and retrieval processes are required. Online map interaction is one of the emerging applications where these technologies can be deployed to access rich multimedia data associated to physical places. This source of knowledge can be useful for users in order to automatically compute the route of a trip or to look for entertainment places or hotels.

One of the most extended and powerful maps engine is Google Maps [1], where information is distributed in a set of layers, each one containing specific data, such as map pixel color, street name, or a particular business name. Recently, georeferenced street images have also been proposed as a new type of information linked to this map data [1], but up to now the automatic exploitation of this information has been rather limited.

This database can be potentially useful for several applications, such as looking for businesses by means of logos, urban furniture maintenance, or to retrieve any kind of visual object category. Recently, the authors of [2] have pro-

posed the use of georeferenced images to recognize a limited set of tourist sights via mobile vision services in urban scenarios. Other works have used GPS information to complement the visual information for event recognition in video sequences [3], or to define image indexing of spaces [4], which can be applied to guide image acquisition for spatial perception, to help camera setting for surveillance, or for image indexing in virtual tours.

In this article, we present our work on a database of about half million georeferenced images that cover the whole city of Barcelona. Images were obtained by means of a Mobile Mapping process. This database is processed for getting a visual content description. We have used fast affine Hessian-Laplace region detection and SURF description, which results in about 500.000.000 keypoints for the whole database. Next, a semantic indexing is performed in order to build a hash code for fast content access. With this information, we have built an object retrieval prototype that retrieves object categories demanded by the user. The new information is visualized in an interactive map application that uses the Google Maps API.

The paper is organized as follows: Section 2 describes the main modules of the system, corresponding to the data base acquisition and description, user queries, and the retrieval process. Section 3 validates the proposed system, and finally, section 5 concludes the paper.

2. BARCELONA MINING APPLICATION

We have developed a web-based application that works on a proprietary image data base of Barcelona, which has been acquired using a mobile mapping system [5]. All these images are processed so that a high amount of visual features are defined and codified by means of a hash code. These features represent the visual appearance of a specific geographical point and can be accessed by content.

A *javascript* program in the client web browser controls the interaction with the map, sending queries to a server that links our data and application to Google Maps. Given a query asking for topological information, street names, satellite imagery or a certain object category, the server retrieves the required information. Finally, the received data is shown

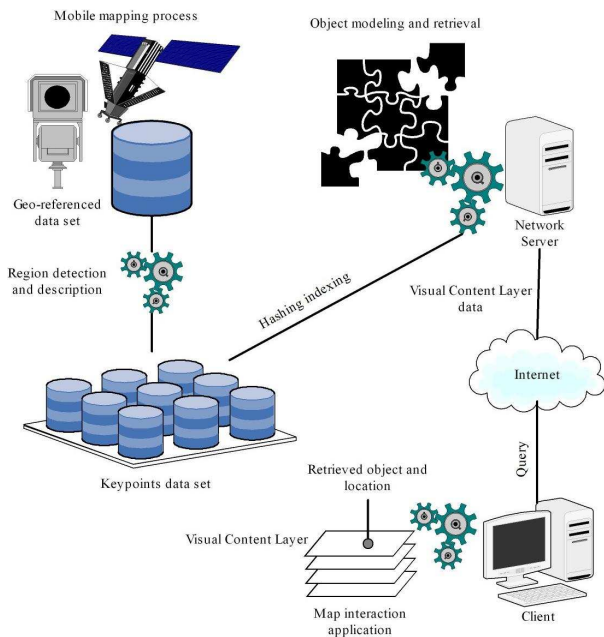


Fig. 1. Barcelona mining scheme.

in a Visual Content layer on the client, indicating their corresponding geographical location. The architecture of the Barcelona mining project is shown in Fig. 1. Next, each implemented module of this system is explained in detail.

2.1. Mobile Mapping Acquisition Process

The mobile mapping system [5] used in this work has been initially developed in order to create an inventory of the Barcelona streets, specially, to capture building facades. A total of 3.800 streets (corresponding to more than 1.290Km) were scanned using six color cameras with different orientations, all of them synchronized with a GPS/INS system, acquiring an image each five meters. The result is a set of about 500.000 georeferenced images. The mobile vehicle and examples of captured scenes of Barcelona are shown in Fig. 2.

2.2. Keypoint database generation

The captured images are stored using the *LizardTech MrSID* format, having a resolution of 1200×1600 pixels. At each image, we have applied the fast affine Hessian-Laplace region detector [6]. After normalizing the detected ellipses, the SURF descriptor [7] is computed for each detected region. We obtained more than 1000 regions per image, which corresponds to a total of 500.000.000 descriptions in the database. In order to manage this huge amount of data, we generate a compact binary representation for each descriptor, which is used as a hash code in order to rapidly find points in the



Fig. 2. Mobile Mapping vehicle and examples of captured urban scenes.

database similar to a given one.

Since we need to save millions of points in our database, the use of the Hessian based keypoints allows to use the sign of the Laplacian as a first indexing criterion. This allows to distinguish bright patterns on dark backgrounds from the reverse case. Nevertheless, the number of necessary comparisons to index the database recommends the use of hashing strategies. For this task, the *Candid covariance-free incremental principal component analysis* (CCPA) approach of [8] is applied in order to project the original 64-dimensions SURF feature space into a 31-dimensions feature space. This strategy allows a compact representation of the descriptors, computing the principal components incrementally without estimating the covariance matrix, and thus, being able to fastly process huge amount of data. This is performed keeping the scale of the observations and computing the mean of observations incrementally. Once the principal components are computed, the first 31 most significant components are selected, keeping to more than the 95% of the variance. Each of these values is set to one if it is greater than zero, or to zero otherwise. Finally, an additional bit corresponding to the sign of the Laplacian is introduced as the most-significant bit, obtaining a final representation of 32 bits. We also include for each keypoint its corresponding frame identifier, and its spatial coordinates (x, y) in the image.

2.3. Client application

The client application takes advantage of the recently opened Google APIs, that allows a fast and simple integration of our database with a complete map engine. Google Maps organizes geographical information in different layers, each one with a specific content, such as aerial or satellite images, street names, traffic information, businesses, etc. In this pa-

per, we propose to incorporate a new Visual Content layer, that allows content-based visual queries.

The client application allows the user to choose between a set of predefined visual objects represented as a set of point features. The selected object is sent to the server-side application, where it is processed. Once the retrieval process is ended, the client shows the points where candidate instances for the selected object are found, with the image of the instance and its geographical coordinates. In order to validate or complement the results, a Google Street View of the area is displayed when it is available.

2.4. Server application

The server application is developed as a Java servlet which performs the object retrieval task and generates the client code to show the results. The retrieval process consists of two basic operations, which are used in different steps of the retrieval process:

Point-to-point correspondence: This process consists of finding points in a given scene which are similar to each point of the model. In order to decide when two points are similar enough to link them we compute the distance between the descriptors of keypoints in the model and the descriptors of keypoints in the scene. If the distance between one point of the model with their nearest point is larger than a certain threshold from the second nearest point, the two points are linked.

Homography estimation: When a set of keypoints of the model and a set of keypoints of the scene have been linked, we use those correspondences in order to estimate the homography or collineation H which allows to transform the model points x to their correspondent scene points x' . The result is a transformation matrix H , where $x' = Hx$. The estimation of H is performed using the RANSAC algorithm, which allows to discard outliers, and therefore it is robust to errors on the point to point correspondence process.

When we are looking for some type of urban furniture which can be altered (vandalism, deformations, painting, etc.) or when it can be viewed from several points of view, it is useful to provide the system with more than one sample image, which allows a better generalization of the object, being able to find a wider set of instances. When multiple samples are used, it is necessary to normalize them, obtaining a virtual model which shares the information of the different instances.

The normalization process consists of selecting one of provided images as the reference model. Then, for each of the other samples, we find the corresponding points between this instance and the reference model, and estimate the homography between all the points. Finally, this homography is used in order to project the points of the given image over

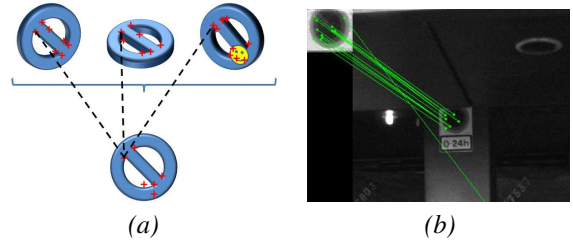


Fig. 3. (a) Models alignment. (b) Point-to-point correspondence sample.

the reference model. In Fig. 3a, an example of this process is shown, where four models are summarized in just one. Note that in the virtual model, we will have keypoints that do not correspond to an image artifact, as the ones generated by a simulated sticker, and different keypoints can be projected at the same point. This process only affects the spatial position of the keypoints, not their descriptors, which remain invariant.

Once we have a unique model, either because we use only a model or because a virtual model has been created from a set of models, next step consists of finding that model in the city images. From a naive point of view, this could be as simple as repeating the model alignment processes between the model and each scene image. That is, extracting and describing the keypoints of the model and the scene, finding the corresponding between pairs of points, estimating the homography between the model and the scene on those scenes where enough matching points are found, and finally taking a decision based on a matching measure between the projected model and the image. Nevertheless, from a practical point of view, this is the most complex stage because of the number of points and the computational burden associated to this process. In order to retrieve the keypoints from the database, the hash code described at the previous section is computed for each detected keypoint. Then, all the images with a large amount of corresponding points are selected, and the homography is estimated to verify that the points are coherently distributed.

3. VALIDATION

In order to test the image recovery system, we use the Google Search API, which allows to run a large set of queries (i.e. business name, address, etc.) and retrieve their position in a map. The experiment we have performed corresponds to the following kind of queries: Where could I find a shop with this logo? For example, we can ask for the offices corresponding to the most popular saving bank in Barcelona (with about 780 possible locations in the map). If we search the logo of this bank (see Fig. 4) in our image database, we can use the geographic distance to the nearest office, obtained using the Google Search API, to decide whether this instance was cor-

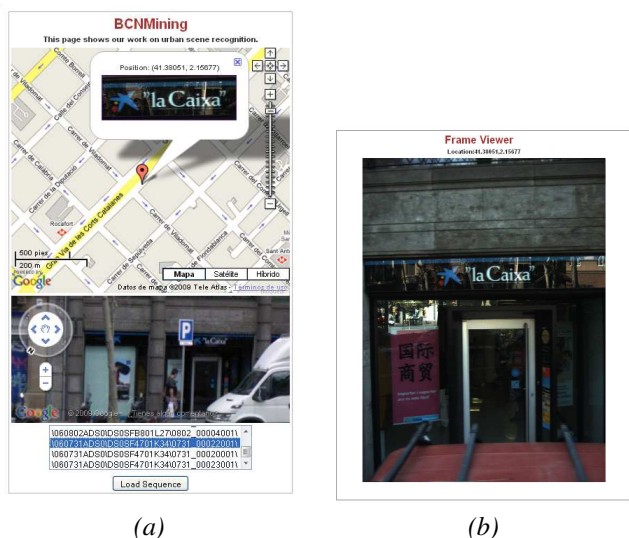


Fig. 4. (a) Barcelona mining interface. The detected object is shown with their position and the Google Street View is synchronized at object position. The object image contains a link to the whole frame, shown in (b).

rectly detected or not. Notice that although this validation method does not take into account some previsible fails due to occlusions or advertisements that contains the logo, it is a measure of the accuracy of the system respect to the real world. In this test, we obtain a hit ratio greater than the 60%.

In addition, we also validated our approach using the no park sign (see Fig. 3b). In this case, our methodology is applied over a set of 100 manually labeled images, obtaining similar results.

4. FUTURE WORK

In this paper, we have presented the first prototype of the Barcelona Mining Project. The two tests performed were carried out using two visual object categories of Barcelona: traffic sign and bank logo. The models of these symbols were previously aligned and saved in order to perform the indexing over the database and test the system. Next step of our research consists on integrating in the interface an option to generalize the process in order to retrieve any kind of image model given by the use. We also want to include more discriminative information into the descriptor, such as including color information. Also, we are studying alternatives in order to compare the robustness and speed of the hashing strategy. For example, a clear possibility is to learn off-line the keypoint descriptors so the indexing could be performed by means of a simple testing of a previously learnt classifier, as explained in [9].

5. CONCLUSION

In this paper, we presented a feasible way to retrieve image object models from a huge descriptor database. The most part of Barcelona city was captured by means of half million images using a Mobile Mapping system. More than a thousand of keypoints were detected, described, and indexed for each image in the database, using a hash table to encode the feature space. All this information is used in order to create a new Visual Content Layer for Google Maps, where the retrieved object candidates of a selected model are shown with their corresponding geo-referenced information, allowing an efficient and scalable way of accessing maps by its visual content.

6. REFERENCES

- [1] Google APIs <http://code.google.com/intl/en/apis/ajax/>
- [2] K. Amlacher and L. Paletta, "Geo-indexed object recognition for mobile vision tasks", *ACM International Conference Proceeding Series archive Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*, pp. 371-374, 2008.
- [3] J. Yuan, J. Luo, H. Kautz, Y. Wu, "Mining GPS traces and visual words for event classification", *Proceeding of the 1st ACM international conference on Multimedia information retrieval*, pp. 2-9, 2008.
- [4] H. Cai, J. Yu Zheng, "Locating key views for image indexing of spaces", *Proceeding of the 1st ACM international conference on Multimedia information retrieval*, pp. 31-38, 2008.
- [5] X. Baró, S. Escalera, J. Vitrià, O. Pujol, and P. Radeva, "Traffic Sign Recognition using Evolutionary Adaboost detection and Forest-ECOC classification", *IEEE Transactions on ITS*, to appear.
- [6] K Mikołajczyk, T Tuytelaars, C Schmid, A Zisserman, J Matas, F Schaffalitzky, T Kadir, and L Van Gool, "A comparison of affine region detectors", *IJCV*, vol. 65, 2005.
- [7] Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool, "SURF: Speeded Up Robust Features", *CVIU*, Vol. 110, No. 3, pp. 346–359, 2008
- [8] Juyang Weng, Yilu Zhang, Wey-Shiuan Hwang, "Candid covariance-free incremental principal component analysis". *IEEE Transactions on PAMI*, vol.25, no.8, pp. 1034-1040, Aug. 2003.
- [9] V. Lepetit, J. Pilet, P. Fua, "Point matching as a classification problem for fast and robust object pose estimation" *CVPR*, vol.2, pp. 244-250, 2004.