STUDY OF SIFT DESCRIPTORS FOR IMAGE MATCHING BASED LOCALIZATION IN URBAN STREET VIEW CONTEXT

David Picard[1], Matthieu Cord[1] and Eduardo Valle[2]
[1]LIP6 UPMC
Paris 6
104 avenue du Président Kennedy
75016 Paris FRANCE
{david.picard, matthieu.cord}@lip6.fr
[2]ETIS, CNRS, ENSEA, Univ Cergy-Pontoise,
F-95000 Cergy-Pontoise
mail@eduardovalle.com

KEY WORDS: Image, Databases, Matching, Retrieval, Urban, High resolution

ABSTRACT

In this paper we evaluate the quality of vote-based retrieval using SIFT descriptors in a database of street view photography, a challeging context where the fraction of mismatched descriptors tends to be very high. This work is part of the iTowns project, for which high resolution street views of Paris have been taken. The goal is to retrieve the views of a urban scene given a query picture. We have carried out experiments for several techniques of image matching, including a post-processing step to check the geometric consistency of the results. We have shown that the efficiency of SIFT based matching depends largely on the image database content, and that the post-processing step is essential to the retrieval performances.

1 INTRODUCTION

In this paper, we evaluate the effectiveness of a voting strategy using SIFT descriptors for near-duplicate retrieval of urban scenes. We have observed that, compared to previously repported applications of SIFT (object recognition, stereoscopy, etc.) (Lowe, 2003) this context presents the challenge of a very high rate of descriptor mismatches, due to the complexity of both the scene and the transformations it might suffer. We have thus, evaluated how different strategies to filter out the false matches can improve the effectiveness of retrieval.

This study is part of the iTowns project, which is about defining a new generation of multimedia web tools that mixes a broadband 3D geographic image-based browser with an image-based search engine [1]. Fig. 1 shows an example of pictures taken for the project.

The first goal of the new type of search engine, is to retrieve, in the high-resolution database, the scene corresponding to a given query image. Let us imagine the following scenario: a user is looking for information about a restaurant in front of him (feedback from patrons, for instance). He takes a picture of the restaurant with his phone and send it to the iTowns web server. The image is matched on the database and the desired information is retrieved and sent back to the user.

In order to accomplish this goal, there is basically three steps to perform :

1. Match the query image with the corresponding scene in the database.
2. Find information associated with the scene and related to the query.
3. Retrieve only relevant information regarding the user interests.

[1]See http://itowns.ign.fr

In this paper, we focus on the first part, and consider the use of state of the art techniques for near-duplicate image matching. Recently, techniques have been developed for the detection of copies where transformations between images are well known (rotation, scaling, global illumination change etc). Those techniques involve the extraction of points of interest in the images, then the matching of the points in the query with the points in the database, and the aggregation of the matches for images of the database using a voting strategy. We try to extend these techniques to the matching of images with less constrainted, and thus more realistic transformations (change of viewpoint, local illumination, etc).

The paper is organized as follows: the next section introduces keypoint-based image matching. We explain in section 3 the strategy used to perform an efficient approximate *k-NN* search in the database in order to associate query points with points in the database. Then, we detail in section 4 the geometrical consistency used to filter irrelevant matches. Experiments are done on two representative subsets of the iTowns collection, and results are shows in section 5, before we conclude.

2 KEYPOINTS BASED IMAGE MATCHING

The essential elements of keypoint-based image matching appeared in (Schmid and Mohr, 1997): the use of points of interest, local descriptors computed around those points, a dissimilarity criterion based on a vote-counting algorithm, and a step of consistency checking on the matches before the final vote count and ranking of the results. We use the SIFT points of interest (Lowe, 2003) to describe the

Figure 1: Panoramic view of the *Place de la Nation* from the project iTowns.

image (Fig. 2). The SIFT descriptor consists in a 128-dimensional vector containing a set of gradient orientation histograms.

Figure 2: SIFT points of interest with respecting scales.

The classic method to use keypoints for image matching is pair-wise image comparison. For all points of a query image A, find the best matching point in a target image B. If the resulting match has good contrast (*i.e.* the distance of the query point to the best point in B is far less than the distance to the second best, meaning that the query point has only one corresponding target point), add a vote to B. An example of matching points is shown on Fig. 3. The best image in the database corresponding to the query image is the one with higher votes.

Figure 3: SIFT points matching between a query taken with a mobile phone and an image from the iTowns database.

One of the problems of pair-wise image comparison is that it induces a sequential, linear-time, processing, which is unfeasible for large databases. Hence, instead of finding best matches between keypoints of query and target images, the best matches are found between the query and the keypoints in the entire collection. The retrieval scheme is as follows :

1. For each points in the query, find the *k-nearest neighbours* (*k-NN*) in the database.
2. For each neighbour found, add one vote to the corresponding image.
3. Rank image by descending number of votes.

The main difference with pair-wise comparison is that each keypoint of the query has k associated matches. Thus, points of the query with no corresponding points in the database (points of objects that are not in the database for instance) will still vote. Those votes are randomly distributed among images and thus contribute to increase the ranking of irrelevant images.

In order to remove the influence of those irrelevant points, a geometrical constraint is applied to the matches, removing points in the target that are not coherent with the spatial distribution of points in the query.

3 APPROXIMATE K-NN SEARCH

There are several techniques for efficient kNN search on large databases, like the KD-tree (Friedman et al., 1976), the LSH (Indyk and Motwani, 1998) or projective methods (Kleinberg, 1997). A comprehensive study can be found in (Valle, 2008). Those methods are all approximate because, in order to obtain more efficiency they sacrifice exactness in the name of speed. That means that they find the correct answers with good probability, but not certitude.

We have chosen Multicurves (Valle et al., 2008), a method based on space-filling curves, which are fractal curves able to map the dimensions of the input space into an one-dimensional space, while locally preserving the order (*i.e.*, putting near in the curve point which are near in the space). The one-dimensional data can then be indexed using traditional efficient techniques.

The particularity of Multicurves is using several of those curves at once: first, it projects the input space into a few moderate-dimensional subspaces, then it uses one space filling curve to index each one of those subspaces. This allows the method to better manage the problems associated to high-dimensional indexing. In our experiments, we have used Multicurves with 4 curves, each of them indexing 32 of the 128 dimensions that compose the SIFT input

space. Details of the method as well as its evualuation for copy detection can be found in (Valle et al., 2008).

Each keypoints of the database within the *k-NN* is added to the list of matches of the image it belongs. A basic method to retrieve images corresponding to the scene is to rank the images by descending order of the size of the lists of matches.

4 GEOMETRICAL CONSISTENCY

Since every point of the query is associated with many points in the database, irrelevant points of the query will still influence the ranking. However, we can make the assumption that for those matches, the relative positions of the query and target points within their respective images are not coherent.Thus, a geometric constraint over the ensemble of matches between two images shall be able to remove the irrelevant matches.

We test two criteria of geometrical consistency. The first criteria is to estimate the 2D affine transform between the two images, and then to remove the points not coherent with it. Although the transformation between the images is indeed 3D, we assume that under small perspective changes, a 2D affine transform is enough to catch the transformation of a single plane (in our case, the front of the building). The algorithm used to estimate the affine transform is RANSAC, a model estimation technique which can deal with a large fraction of outliers (Fischler and Bolles, 1981). An example of matches after the removal of non-coherent points is shown on Fig. 4.

Figure 4: Matching points after the non-coherent to the estimate 2D affine transform matches have been remove.

The second criterion is to keep only the matches which correspond to the most frequent angle difference between matched points (Jegou et al., 2008). This is done by computing an histogram of the difference between the principal direction of the query and the target point of a match. We then keep the matches corresponding to the most frequent value in the histogram.

5 EXPERIMENTS

5.1 Protocol

We have tested four methods for comparison on two subset of the *iTowns* images, namely:

- A pairwise matching using a distance contrast criterion (named *Image Matching* there after).
- A *k-NN* search plus a vote (named *Brute vote*).
- A *k-NN* search plus the angle differences consistency criterion (named *Angle differences*).
- A *k-NN* search plus the 2D affine transform consistency criterion (name *Ransac*).

We set $k = 10$ for the *k-NN* Search. The parameters for the *RANSAC* algorithm were empiricaly set to 15 pixels maximum distance to fit the model and minimum 3 inliers for the affine transformation.

The first dataset consisted of 82 images of a single street (about 350 000 keypoints). The query set contained images taken by a mobile phone in front of some of the shops in the street. As the images (both in the query set and in the database) are direct views of the buildings, we considered this test as easy, since the transformation between query and its corresponding target images is simple. The second dataset contained 300 images of a large boulevard (about 3.5 millions of keypoints). The queries where taken with a mobile phone from the sidewalk. As the vehicle taking the pictures was in the middle of the street, the targeted regions of the images (a shop, for instance) are very small. Thus, few keypoints of each image are describing something we might be looking for. As there are many severe transformations (scaling, viewpoint changes), we consider this test difficult. For both sets, we have manually built the groundtruth by annotating which images correspond to each query.

We have used three criteria for the evaluation. The first consisted in measuring the rank of the first relevant image retrieved (average of the query set). The second measure was the evolution of the number of relevant image in the retrieved set, as the size of this set increased. The third criterion was the precision, the number of relevant images retrieved over the number of images retrieved.

5.2 Results on Dataset 1

An example of the first images retrieved using the *Brute vote* is shown on Fig. 5. The first images retrieved with this technique have about 2000 matching keypoints (images in this set contain about 5000 keypoints). There are several occlusions between the query image and the images of iTowns (for instance the car in front of the shop). However, a relevant image is found within the first images.

Fig. 6 presents the same result, but with the *Angle differences* refinement. The first images retrieved have about 200 matching keypoints. As we can see, the refinement introduced a re-ranking of the first images profitable to

Figure 5: First images retrieved using *Brute vote*. The query has a dark red border, while relevant images have a bright green border.

the relevant image. The same query but with the *Ransac* method is shown on Fig. 7. Images retrieved have less than 10 matching keypoints. The removal of non-coherent points increases the ranks of relevant images. The improvement is thus better than the one of the *Angle differences* refinement.

Figure 6: First images retrieved using the *Angle differences* refinement.

Figure 7: First images retrieved using the *Ransac* refinement.

We have computed the mean best rank among relevant images for a set of ten queries. We also compared the multicurves approach to a linear processing of the database for the *k-NN* search, in order to see the influence of the approximate search. The ranks and times are shown in Table 1.

Method	mean best rank	time
Image matching	27.09	11514s
Linear search	5.45	22967s
Brute vote	14	447s
Ransac	**1.09**	-
Angle Differences	7.91	-

Table 1: Mean best rank for the first dataset. '-' denotes a time not computed.

As we can see, the time used for the pair-wise comparison or for the linear *k-NN* search are prohibitive. Since *Brute vote* uses Multicurves, which is an approximate *k-NN* method, we should expect some degradation when compared to *Linear search*, which uses the costly exact *k-NN* search. We note, however, that by using *Ransac*, the precision lost is more than compensated. The *Ransac* refinement has the best results, and is totally satisfactory from the users point of view.

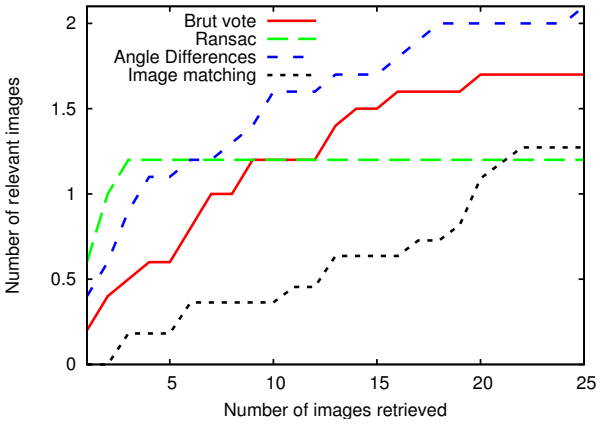

Figure 8: Evolution of the number of relevant images against the number of images retrieved.

We measure the evolution of the number of relevant images as the percentage of the database retrieved increases on Fig. 8. The *Ransac* method outperforms the other in the beginning of the retrieval, but then stops to retrieve images (if no coherent affine transform is found, then the image has a null vote). The *Angle Differences* and the brute voting are less efficient, but still manage to retrieve relevant images within the top 10 images. The pair-wise comparison fails to showing relevant images within the top 10.

The precision (ratio between number of relevant images retrieved and total images retrieved) is shown on Fig. 9. The precision within the first five images retrieved (which is the most relevant metric to the user) is better for the *Ransac* refinement. Past this point, all three *k-NN* based methods are almost equivalent. The pair-wise comparison is surprisingly worse than the other methods.

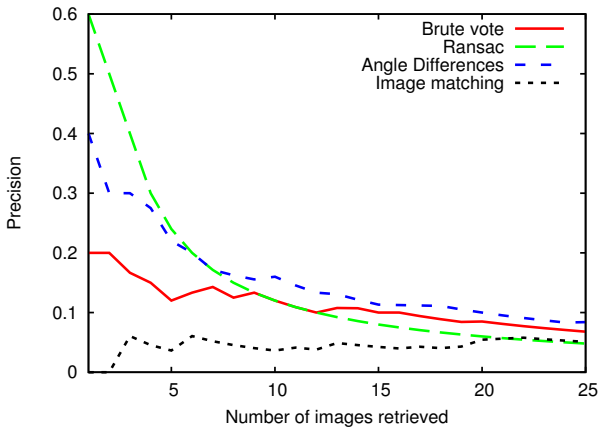

Figure 9: Evolution of the precision against the number of images retrieved.

5.3 Results on dataset 2

An example of results using the brute voting is shown on fig Fig. 10. As we can see, none of the top images are relevant. The same occurs with the angle differences refinement.

Figure 10: Example of first images retrieved using the *k-NN* voting for the second subset.

The RANSAC refinement (Fig. 11) is able to retrieve two relevant images within the first five images, which means that irrelevant matches have been well filtered out.

Like we did for the first subset, we compute the mean best rank shown in table 2. We were not able to compare with linear *k-NN* search due to the time taken by this method.

The first observation is that none of the methods is able to retrieve even one relevant image within the top ten, which means that the methods are not able to give satisfying results from the users point of view. Nevertheless, the geometric consistency step (either *Ransac* or the *Angle Differences*) provides a nice improvement.

Figure 11: Example of first images retrieved using the *Ransac* refinement for the second subset.

Method	mean best rank
Image matching	80.67
Brute vote	98.80
Ransac	**34.40**
Angle Differences	59.10

Table 2: Mean best rank for the second dataset.

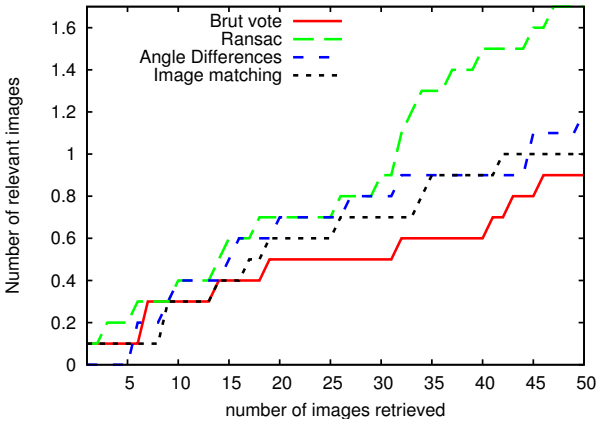

Figure 12: Evolution of the number of relevant images against the number of images retrieved.

The evolution of the number of relevant images is shown on Fig. 12. As we can see, all methods are almost equivalent, with the *Ransac* strategy being a little better for the last 20 images of the top 50.

The precision is shown on Fig. 13, and is very low for all methods. The best result is obtained for the *Ransac*

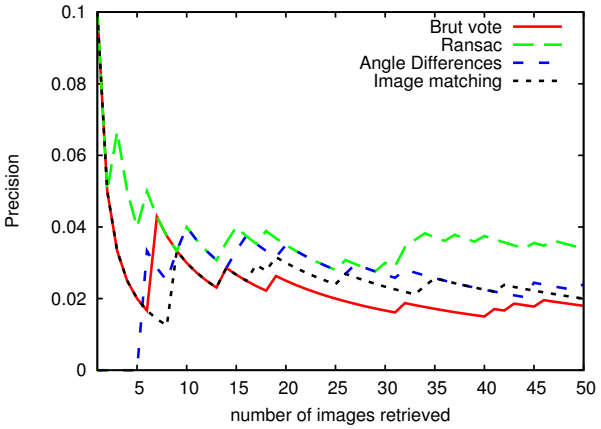

Figure 13: Evolution of the precision against the number of images retrieved.

strategy, but it is still under 5% most of the retrieval. In overall, all methods failed at finding the relevant scene in the database.

6 CONCLUSION

In this paper, we have reviewed the use of keypoints based voting strategy for image matching in the context of the iTowns project. We have tested different strategies (pair-wise comparison, *k-NN* search with brute voting, angle differences refinement, and 2D affine transform estimation) on two subset of urban scene database.

We have first found that there is no penalty in using an approximate *k-NN* search, which is a huge improvement on the retrieval speed. Even for small datasets like the first we used, a pair-wise comparison or a linear *k-NN* search is not feasible for interactive application.

The second point we have found is that the post-processing of the voting strategies is essential to the success of the retrieval. The *Ransac* refinement is the only one able to retrieve at least one relevant image within the first five images, which is the main criterion for a user in this kind of task. A further improvement could be the estimation of more complexe transformation that are more robust to perspective changes.

However, overall results largely depend on the database content. In the case of a small database (which can be obtained through geolocalization) with well taken pictures like the first we used, the results are good enough to be used in the intended application.For the second database, the quality of the results is very low, making them inadequate for our applications. This lack of quality might be an intrinsic characteristic of SIFT when confronted to images like ours, that contain many problematic features (complex shadows, trees, branches, etc), which spawn a huge amount of descriptors with low discriminant power. Those points increase dramatically the number of false matches, inflating the rank of of non relevant images (such as on Fig. 14, which has more matches than the relevant images). As improvement, we suggest a filtering of the database in order to remove points that are not informative.

To conclude, we consider the extension of keypoints based method from copy detection to the matching of scene in difficult context as not successful. We think there is more work to do both on the descriptors and on the matching process. We intend to share our databases and groundtruth with the community in order to allow the benchmarking of those tasks on difficult images.

Figure 14: False matching between two images after geometric consistency check.

REFERENCES

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), pp. 381–395.

Friedman, J., Bentley, J. L. and Finkel, R. A., 1976. An algroithm for finding best matches in logarithmic expected time. Technical report, Stanford, CA, USA.

Indyk, P. and Motwani, R., 1998. Approximate nearest neighbors: towards removing the curse of dimensionality. In: STOC '98: Proceedings of the thirtieth annual ACM symposium on Theory of computing, ACM, New York, NY, USA, pp. 604–613.

Jegou, H., Douze, M. and Schmid, C., 2008. Hamming embedding and weak geometric consistency for large scale image search. In: A. Z. David Forsyth, Philip Torr (ed.), European Conference on Computer Vision, LNCS, Vol. I, Springer, pp. 304–317.

Kleinberg, J. M., 1997. Two algorithms for nearest-neighbor search in high dimensions. In: STOC '97: Proceedings of the twenty-ninth annual ACM symposium on Theory of computing, ACM, New York, NY, USA, pp. 599–608.

Lowe, D., 2003. Distinctive image features from scale-invariant keypoints. In: International Journal of Computer Vision, Vol. 20, pp. 91–110.

Schmid, C. and Mohr, R., 1997. Local grayvalue invariants for image retrieval. IEEE Trans. Pattern Anal. Mach. Intell. 19(5), pp. 530–535.

Valle, E., 2008. Local-Descriptor Matching for Image Identification Systems. PhD thesis, Univ. Cergy-Pontoise, ETIS, UMR CNRS 8051. Direction : S. Philipp-Foliguet, M. Cord.

Valle, E., Cord, M. and Philipp-Foliguet, S., 2008. High-dimensional descriptor indexing for large multimedia databases. In: CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management, ACM, New York, NY, USA, pp. 739–748.