FAST INDEXING FOR IMAGE RETRIEVAL BASED ON LOCAL APPEARANCE WITH RE-RANKING

Hao Shao[1], *T. Svoboda*[1], *V. Ferrari*[1], *T. Tuytelaars*[2] *and L. Van Gool*[1 2]

[1] Computer Vision Lab, Swiss Federal Institute of Technology, Zurich, Switzerland
{haoshao,svoboda,ferrari,vangool}@vision.ee.ethz.ch
[2]ESAT, PSI, Katholieke Universiteit Leuven, Leuven, Belgium
{tinne.tuytelaars,luc.vangool}@esat.kuleuven.ac.be

ABSTRACT

This paper describes an approach to retrieve images containing specific objects, scenes or buildings. The image content is captured by a set of local features. More precisely, we use so-called invariant regions. These are features with shapes that self-adapt to the viewpoint. The physical parts on the object surface that they carve out is the same in all views, even though the extraction proceeds from a single view only. The surface patterns within the regions are then characterized by a feature vector of moment invariants. Invariance is under affine geometric deformations and scaled color bands with an offset added. This allows regions from different views to be matched efficiently. An indexing technique based on Vantage Point Tree organizes the feature vectors in such a way that a naive sequential search can be avoided. This results in sublinear computation times to retrieve images from a database. In order to get sufficient certainty about the correctness of the retrieved images, a method to increase the number of matched regions is introduced. This way, the system is both efficient and discriminant. It is demonstrated how scenes or buildings are recognized, even in case of partial visibility and under a large variety of viewing condition changes.

1. INTRODUCTION

Most content-based image retrieval systems focus on the overall, qualitative similarity of scenes. Hence, global aspects like colour gamuts and coarse spatial layout are among the features most often used. This paper deals with a somewhat different kind of retrieval, in the sense that images are sought that contain the same, prominent objects as the query image. One could e.g. look for images on the Internet with the same statue as contained in the query image, or the system could recognize one's location, by taking an image and looking for the most similar image in a large database of images taken all over a city. The latter type of application is given as an example further on in the paper.

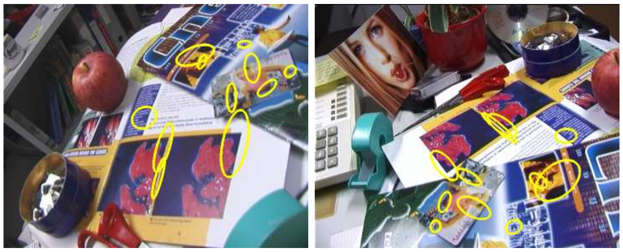

Fig. 1. Two different views of the same scene, with some invariant regions added. Note how they cover the same physical parts of the scene.

For the rather stringent type of similarity search propounded here, global and qualitative features no longer suffice. Hence, we propose the use of local colour patches as features, which are compared rather precisely. In particular, we propose to extract so-called invariant regions, which have become popular recently [1, 2, 3, 4]. Here, we use the intensity-based regions proposed by Tuytelaars and Van Gool [4]. Invariant regions correspond to small image patches, constructed around special points (here intensity extrema). These regions are special in that they automatically adapt their shapes in the image to the viewpoint of the camera. The crux of the matter is that this adaptation takes place without any knowledge about the viewpoint and without any comparison with other images in which the same regions are visible. Thus, in principle, the same physical parts are extracted from two images of the same scene, even if these have been taken from different viewpoints (and under different illumination). Fig. 1 shows an example. It shows two images and some of the invariant regions that have been extracted independently from both. Upon closer inspection, one sees that these regions enclose corresponding parts of the image indeed. Of course, in practice many non-corresponding regions are extracted as well. They are not shown here as the image would become to overloaded.

Once such regions have been extracted, the colour pattern that they enclose is described on the basis of invariant features. Invariance is both under affine geometric deformations and linear changes of intensity. The former renders the feature vectors describing the regions invariant under changes in viewpoint, the latter renders them invariant under changes in illumination. Both are necessary requirements for the intended applications. In our case, the feature vectors are composed of moment invariants [5]. The feature vectors make it possible to match invariant regions very efficiently, using hashing techniques. Such matching lies at the heart of our approach, since the similarity between images is quantified as the number of matching features. For more details on the extraction and description of the invariant regions, the reader is referred to the references given before.

The structure of this paper is as follows. Section 2 describes the way in which the efficient matching of invariant features between a query image and a database of reference images is achieved. Section 3 discusses how the matching can be improved, increasing the number of matches. This step increases the robustness of the retrieval and is only applied to images for which a sufficient number of matching regions have been found in the first step. Finally, section 4 describes an experiment, where query images taken in Zurich are compared against a database of over 1000 other images, taken all over the city. This work is part of a project on wearable computing, where vision provides one of the means to determine the location. In addition, it will allow to supply tourist information on the site to the user.

2. INDEXING AND RETRIEVAL

2.1. Vantage Point Tree

Yianilos introduced the Vantage Point Tree, or VP tree for short in [6]. A VP tree relies on pseudo-metrics. First a point **v** (vantage point) is selected. To construct the index, the database points are sorted according to their distance from **v** (i.e., in ascending order of $d(\bullet, \mathbf{v})$). The median distance is computed and all points having a distance smaller than the median are assigned to the left subspace of **v**, while the remaining ones are assigned to the right subspace. This procedure is recursively applied to the left and right subspace. The simplest VP trees rely on simple operations to select the appropriate vantage point among a random subset of the points associated with each node of the tree[6]. It is clear that the choice of vantage points at each level of the VP tree plays an important role in the performance of the indexing algorithm. In our implementation, we select the point which is furthest from the center of gravity as vantage point. Our experiments prove that this is a better choice than random selection. Because the vantage point tree is a binary tree structure, binary search algorithm can be used to search the tree. The searching complexity is then ($O \log_2 N$).

Indexing based on VP trees is faster than using K-d trees, since one does not need to project points onto each dimension during the querying stage. Moreover, similarity of regions is expressed as distances between feature vectors, which make VP trees the more natural choice (also using distances) and especially well suited for our task.

2.2. Off line database construction

When constructing the database, we first extract all invariant regions from each image in the database and compute the feature vector of moment invariants for each region. Then we construct the VP tree. As distance measure in feature space, we use the Mahalanobis distance, to correctly take into account the different variability and interdependence of the different components of the feature vectors. To this end, the covariance matrix was estimated based on all the regions extracted from the database images. At each leaf of the VP tree we store not only the feature vector, but also the invariant region and the database image from which it was extracted.

2.3. Retrieval implementation

When a new query image has to be processed, the first step is again to extract all invariant regions from the query image and to compute the feature vectors of moment invariants for them. Each region extracted from the query image is used as a query region to find similar regions in the database based on the binary tree searching algorithm. Here, two regions are considered similar if the Mahalanobis distance between them is below a predefined threshold. We say the query region is *matched* to a region in one of the database images if the latter is retrieved from the VP tree. Once all the regions in the query image have been processed, the images in the database can be ranked based on the number of matches found, i.e. based on the number of regions they contain that are similar to regions in the query image.

3. RE-RANKING THE TOP FIVE

Based on the indexing described in the previous section, images of the same scene are very likely among the top 5 retrieved. However, it is often not at the very first rank (see section 4). This is due to the fact that individual local features are not very discriminative and in a large database many images will have locally similar features. In this section, we propose a method to greatly improve the ranking of the first top 5 retrieved images.

Let us denote the query image as I_q and an image of the same scene in the database as I_d. The idea is to exploit the information supplied by a correct match in order to generate many other correct matches. Consider a region C_q which has been extracted in the query image, but not matched to

I_d, and another region S_q which has been matched to S_d in I_d. If C_q and S_q are spatially close and lie on the same physical surface, then they will probably be mapped from I_q to I_d by similar affine transformations. Hence projecting the candidate region C_q to I_d via the affine transformation that maps the support region S_q to S_d gives us a first approximation of the real C_d. This approximation is then refined by affinely transforming it so as to maximize the similarity between C_q and the deformed C_d. We have developed an efficient algorithm to search a bounded range of the space of affine transformations to realize this refinement step (for details, we refer to [7]). Finally, if the refined C_d has a similarity value to C_q larger than a threshold we consider it *propagated* to I_d. As similarity measure a linear combination of normalized cross-correlation and distance in normalized RGB space is used. This mixed measure proved to be particularly discriminant. If the support region S is a mismatch (e.g. to a wrong database image), or if C and S are not close and on the same physical surface, a high similarity is unlikely to arise and the region is not propagated.

We apply propagation to the 5 image pairs composed by the query image and each of the top 5 retrieved database images. For each pair, we first match regions of a higher resolution version of the images (640×480). A regular grid of overlapping circular regions covering the whole query image is generated. Propagation is then applied with these regions used as candidates and the matches provided by the 2-view matching as support regions.

For a database image that does indeed contain the same scene, many more new matches arise due to the propagation, as the parts of the images around the original matches probably correspond as well. For database images that do not contain the same scene, but erroneously got a high ranking by the indexing mechanism, on the other hand, the parts around the (mis)matches are probably different, and at most a few regions get propagated. The net effect is a permutation of the ranking that better reflects the relative similarity of the 5 database images to the query one (remember we rank according to the amount of matches).

Propagation is of course slower than indexing, but it is applied only to a few images. The power of the retrieval approach proposed in this paper relies in this combination: first a very efficient indexing stage which isolates only a few likely images, and then a more expensive, but also more detailed and accurate discrimination stage only on those. This way both efficiency *and* discriminancy are achieved.

4. EXPERIMENTS

To test our retrieval system, we used our ZuBuD database containing 1005 images captured in Zurich city and it contains image of over 200 different buildings [8]. The images have been acquired in different seasons, by using different cameras and they pose numerous image recognition challenges. The query is represented by 115 images which are *not* included in the database. Figure 2 shows the computation time needed to retrieve a top five ranking based on the indexing method described in section 2. Clearly, the retrieval speed is sublinear ($\mathcal{O}\log_2 N$) to the database size and linear in the number of regions extracted from the query image.

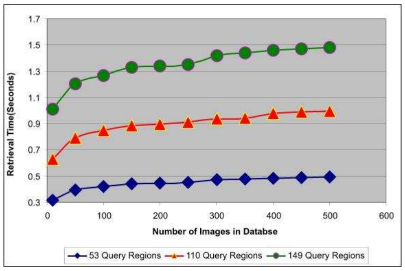

Fig. 2. Searching the database is sublinear in the number of images in the database and linear in the number of query regions per query image.

The indexing step ranked the correct building 97 times (out of 115) between the first five. However, only 47 times is the true match on the first place. The results are summarized in figure 3. In order to test the re-ranking algorithm, we selected randomly 10 cases where the indexing algorithm ranked the correct image between positions 2 to 5. The procedure moved the true match 9 times to the first place mostly with a very strong confidence and once to the second place. Figure 4 shows two examples of successful re-ranking.

5. CONCLUSIONS

In this contribution, a method to retrieve images from a database based on local features was proposed. It makes it possible to search in a database for images containing the same object or scene as shown in a query image, even in case of large changes in viewpoint, occlusions, partial visibility, changes in the illumination conditions, etc. The use of invariance combined with indexing techniques based on vantage point tree make the system extremely efficient and suitable for applications on a wearable system, where the human-machine interaction requires fast processing. On top of this, a slower but more accurate re-ranking mechanism is proposed, that allows to obtain improved retrieval results while keeping the computation times reasonable.

Acknowledgements

H. Shao and T. Svoboda acknowledge support of the Wearable computing polyproject of Swiss Federal Institute of Technology at Zurich. V. Ferrari has been supported by the EU project CIMWOS and T. Tuytelaars by the EU project

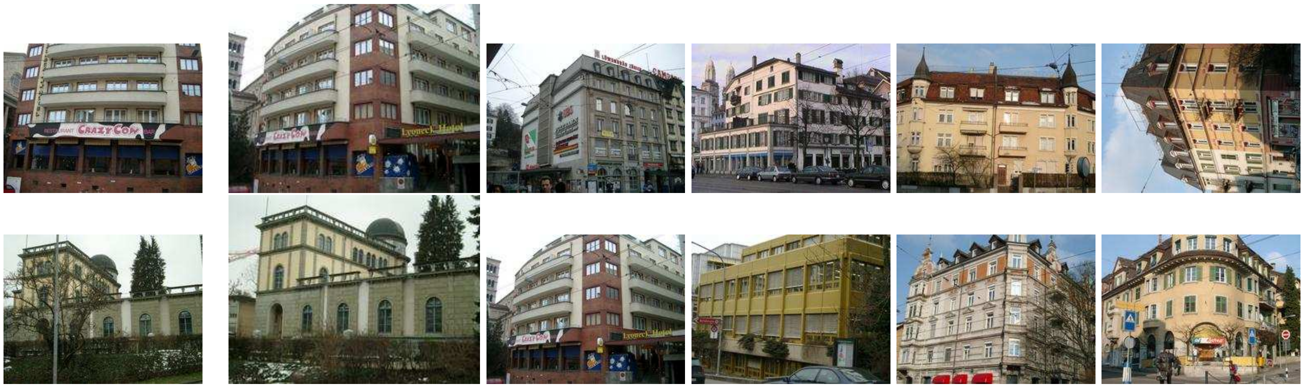

Fig. 4. Two examples of a successful re-ranking. The query image on left. From the second, numbered 1,2,3,4,5 images are returned by re-ranking. Top's order is 3,1,2,4,5 and bottom's order is 5,2,3,4,1 given by indexing before re-ranking

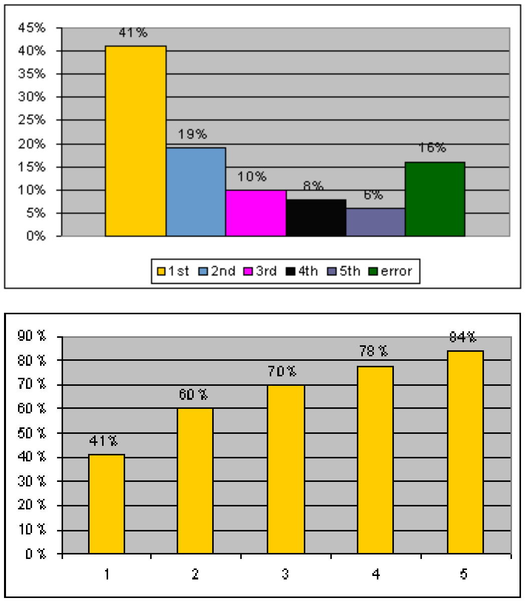

Fig. 3. Performance of our system on a ZuBuD database containing 1005 images: **Top:** percentage of correctly retrieved images at each of top 5 positions and **Bottom:** percentage of correctly retrieved image within top 5 positions.

VIBES and the National Fund for Scientific Research Flanders (Belgium).

6. REFERENCES

[1] A. Baumberg, "Reliable feature matching across widely separated views," in *ICCV*, 2000, pp. 774–781.

[2] J. Matas O. Chum, M. Urban and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *British Machine Vision Conference*, 2002.

[3] K. Mikolajczyk and C. Schmid, "Indexing based on scale invariant interest points," in *ICCV*, 2001, pp. 525–531.

[4] T. Tuytelaars and Van Gool., "Wide baseline stero based on local affinely invariant regions," in *British Machine Vision Conference*, 2000.

[5] F. Mindru, T. Moons, and L. Van Gool, "Color-based moment invariants for viewpoint and illumination independent recognition of planar color patterns," in *Proceedings of international conference on advances in pattern recognition*, 1998.

[6] P.N. Yianilos, "Data structure and algorithms for nearest neighbour search in general metric space," in *Proceedings of the 4th ACM-SIAM Symposium on Discrete Algorithms*, January 1993, pp. 311–321.

[7] V. Ferrari, T. Tuytelaars, and L. Van Gool, "Wide-baseline multiple-view correspondences," in *Computer Vision and Pattern Recognition*, July 2003, To appear.

[8] H. Shao, T. Svoboda, and L. Van Gool, "ZuBuD — zurich buildings database for image based recognition," Tech. Rep. 260, Computer Vision Laboratory, Swiss Federal Institute of Technology, March 2003, Database downloadable from http://www.vision.ee.ethz.ch/showroom/ZuBuD.