GRAPH-BASED APPROACH FOR 3D OBJECT DUPLICATE DETECTION

Peter Vajda, Frederic Dufaux, Thien Ha Minh and Touradj Ebrahimi

Multimedia Signal Processing Group – MMSPG
Institute of Electrical Engineering - IEL
Ecole Polytechnique Fédérale de Lausanne - EPFL
CH-1015 Lausanne, Switzerland

ABSTRACT

In this paper, we consider the challenging problem of object duplicate detection and localization. Several applications require efficient object duplicate detection methods, such as automatic video and image tagging, video surveillance, and high level image or video search. In this paper, a novel graph-based approach for 3D object duplicate detection in still images is proposed. A graph model is used to represent the spatial information of the object in order to avoid making an explicit 3D object model. Therefore, better performance is achieved in terms of robustness and computational complexity.

1. INTRODUCTION

Image and video retrieval is an important task in visual computation. Significant efforts have been put in this area. Most existing image search and retrieval methods are based on 2D regions and features. However, these methods often fail to deal with changes in view points.

A new object duplicate detection approach is proposed in this paper. The training set is composed of images from a target object, captured from different directions. The aim then is to determine whether the target object is present in a set of images in other scenes, and to determine the locations and sizes of each occurrence. Such an object duplicate detection approach can be useful in a number of applications. For instance, tags can be propagated to new images based on the detection of the same object in previously annotated images; image and video search can be performed on semantic objects, and finally the occurrence of a precise object, such as a suspect car, can be detected in a large video surveillance database.

Several research works have successfully addressed the problem of identification of specific regions in an image or video database. However, 3D object detection has not received the same interest. Therefore, in this paper, we make a step toward 3D object detection, while keeping the efficiency of 2D.

In most prior work for object duplicate detection, two specific problems can be identified. The first aims at defining a similarity measure between image regions. The second problem consists in locating the position of the target object, based on the previously defined measure.

Related to the first problem, two state-of-the-art techniques should be mentioned. The “Bag of Words Model”, which is based on the histogram of local features. Zhang *et al.* in [1] presented a comparative study on different local features on texture and object recognition tasks based on this technique. The “Bag of Words Model” does not include spatial information from the objects. However, it gives a robust, simple, and efficient solution for recognition.

Conversely, the “Part Based Model” considers spatial information of the local features as well. A promising method in [2] shows that the “Part based Model” performs well even in difficult situations such as in PASCAL VOC 2007 dataset. More precisely, Star Model is used to represent the objects based on Histogram of Oriented Gradient (HOG) features.

The problem of multi-view object detection is still largely unresolved. However, some interesting solutions have been proposed for retrieving different visual views from the set of images or video. An approach described in [3] uses tracking to retrieve several different views of a same object in order to generate its representative model. The model in then used to recognize objects in a simple and accurate manner. In [4] the same task is performed using a 3D model of the object, where affine covariant regions are used for object modeling from video sequences.

For the second problem, namely, the localization of the position of the target object, affine covariant regions provide a set of points invariant to scale, rotation and translation, as well as robust to illumination changes, and changes of viewpoints [5]. On these regions, local descriptors, such as Scale Invariant Feature Transform (SIFT) [6] are extracted. The Generalized Hough Transform or a probabilistic model [8] can then be applied in order to localize the position of the object in the query image.

Our approach combines the advantages of being as efficient as in “Bag of Words Model”, and as accurate as in “Part Based Model”. Another advantage of the proposed approach is that it requires a small number of training images in order to build a good model for the target object. A training phase aims at constructing the spatial relations between features in the target object, which is then represented by a graph. In other words, an attempt toward 3D modeling is made, while keeping the efficiency of 2D processing.

The paper is organized as follows; the proposed method is presented in Section 2. Experiments and results are discussed in Section 3. Finally, Section 4 concludes with a summary and some perspectives for future study.

2. PROPOSED METHOD

In this section, we present an efficient solution for 3D object duplicate detection in static images. The goal is to detect the presence of a target object and to predict its bounding box, based on a set of images containing that object. A small number of training images, typically one to four, containing different views of the target object, are sufficient enough to achieve good performance.

The system architecture proposed in this paper is illustrated in Figure 1.

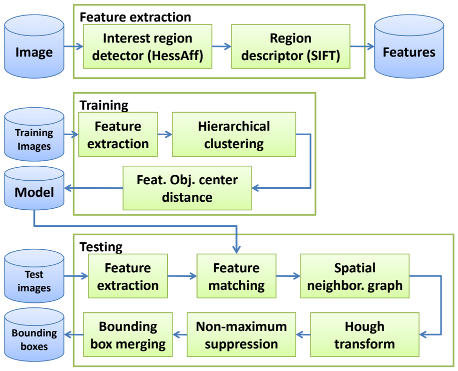

Figure 1: System architecture of the proposed object duplicate detection system

2.1. Feature extraction

To resolve the localization problem efficiently, we use sparse features. Interest regions are extracted making use of a Hessian affine detector, as it has been shown to outperform other detectors [5]. Position, scale and orientation for each interest region are computed. Scale invariant image descriptors (SIFT) are then extracted from interest regions [6], as they remain robust to arbitrary changes in viewpoints.

2.2. Training

During the training phase, a set of images of the target object (from different views, and with eventual deformations) is processed. The training images correspond to a single object filling up the whole field of view. Therefore we assume that the center of the image can be used as a good approximation of the object's center.

First, *features* are *extracted* from the training images, as described in subsection 2.1.

Hierarchical clustering is then applied in the feature space, because of its efficiency. We use Nister and Stewenius "vocabulary tree" algorithm [9], based on fast hierarchical k-means clustering. The number of the clusters, k, defines the branch factor (number of children of each node) of the tree. First, an initial k-means process is run on the training data, defining k clusters. Features are represented by the center of their cluster. The same process is then recursively applied for each cluster to create hierarchical clusters. Computational complexity of hierarchical clustering described above, can be significantly less when compared to a conventional nearest neighbor search.

Finally, the *interest regions vectors* in a coordinate system at the center of the target object are stored, to be used for calculation of a bounding box in testing phase.

2.3. Testing

To retrieve images according to a query, "one-to-one" nearest neighbor *feature matching* is applied. Thanks to the hierarchical clustering described in the previous subsection, this operation requires minimal computational resources. In a first phase, for each feature extracted from the query image, the corresponding nearest neighbor feature in the model is identified based on a Euclidean distance. If the squared value of the distance between these features is larger than a threshold (T_d), the feature in the query image will be disregarded. In a second phase, the previously identified feature points in the model image corresponding to nearest neighbors of features in the query are identified. If their squared distance is larger than T_d, or if they are not the same features as the original features in the phase one, then they are also disregarded. This procedure ensures a better selection of matching features.

To build a *spatial neighborhood graph* for an object, the k_{nn}-nearest neighbors in the query image are calculated (Figure 2). This graph contains spatial information from the potentially matching objects, hence making our algorithm more robust. More precisely, for each feature k_{nn}-nearest matched spatial neighbors are selected, both in the image and in the model, respectively. Connections between the neighbors of a feature in the query image, and those of its corresponding feature's neighbors in the model are created. To avoid wrong connections, only those neighbors at a distance similar to the object size will be connected. The scale values obtained in the feature extraction step as described in subsection 2.1. If the ratio between the normalized scales of the original and its neighbors is between R_u and R_d, then the feature and its neighbors will be connected. The normalized scale is the scale of the feature divided by the scale of the matched feature. These connections are the edges of the graph and the corresponding features will be the nodes. Nodes with at least T_{outdeg} out degree are accepted. The above process will be repeated once more on the remaining features corresponding to the nodes of the graph. This produces a rather robust graph of objects. An expected property of this graph is that ideally nodes of a given edge should not be part of two different objects (i.e. each edge is only part of one object). However, it is possible to have more than one graph per object. With this method, several mismatched features can be disregarded, thanks to spatial position in the object and the size of the features.

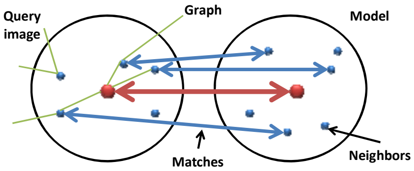

Figure 2: Spatial neighborhood graph construction.

The position is also considered by applying *General Hough Transform* on the nodes of the graph [6]. Each node votes for the center and size of the bounding box in the query image, using the orientation and scale of the extracted feature as described in subsection 2.1. The number of edges on each node is the weight of the vote. Local maxima are found in the obtained histogram. A threshold value is set heuristically and applied. This method results in several bounding boxes with large overlapping and duplicated bounding boxes.

Duplicated bounding boxes can be rejected by applying *Non-maximum suppression* algorithm [7]. In the case where the target object in the training image is slightly different from the predicted object, the resulting bounding boxes are satisfactory.

But if we consider different views of the object as training images, then more separated bounding box can be obtained from the previous method for the same object. To avoid this problem, we *merge the bounding boxes,* considering the number of edges of the graphs intersecting two bounding boxes. The ratio between the number of intersected edges and the number of edges in both bounding boxes is thresholded by T_{bb}. If enough edges are in the intersection, then the two bounding boxes will be merged. The problem of multi-view images, with this simple graph model, is then solved. If more than one object is present in the test image, our algorithm still gives good results, as the graphs are separated. Separated groups of bounding boxes are calculated using Warshall algorithm [10] on the obtained bounding box connection matrix. The bounding boxes in each group are merged together. Each bounding box is representing an object in the test image.

3. EXPERIMENTS AND RESULTS

The parameter settings of our algorithm are the following based on experiments and heuristics: For “vocabulary tree” 128 clusters are generated by hierarchical k-mean clustering algorithm. In the feature matching step, a distance threshold is set to $T_d=10^5$. In the spatial neighborhood graph k_{nn} is set to 15, and T_{outdeg} is set to 4, to accept true features. The normalized scale ratios R_u and R_d are 2 and 0.5. Finally, the threshold $T_{bb} = 0.1$ is chosen to merge bounding boxes.

Two evaluations were performed. We manually evaluate the results by estimating the true positive, true negative, false positive and false negative values regarding the position of the predicted bounding boxes. If the overlap of the predicted and the real object is less than half or more than double of the object then the prediction is considered as false positive. Otherwise it is considered as a true positive.

First, we took 22 images from two objects: “Coca Cola can” and “JPEG book”. Difficult images were taken, considering different views, occlusions, and poor image qualities due to various reasons. For the training image dataset, we used a good quality image of the specified object. Detected images can be seen in Figure 3. The overall results in this database are: Recall = 94% Precision = 94%. These are rather good, taking into account the difficulty of the test dataset. These datasets contain lots of different features for SIFT descriptors. However, a limitation appeared when we tried our algorithm with small, simple or shiny objects such as a pen or a phone. In this case, too little SIFT features were matched to be able to proceed with an efficient object detection.

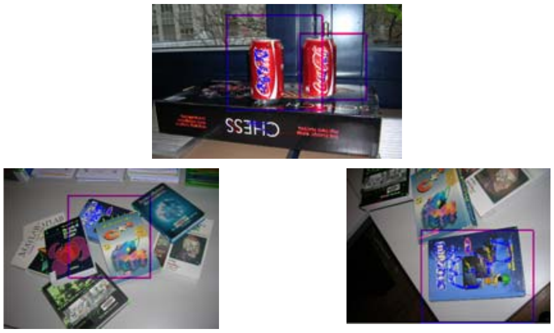

Figure 3: Examples of object duplicate detection for target objects “Coca Cola can” and “JPEG book”

Second, we used the “Model House” (MH) and the “Valbonne Church” (VC) multi-view images from the Oxford Visual Geometry Group databases (Figure 4). Both categories contain several images from a building with different views. One, two, and three most different images were chosen from the sequence for training dataset. The rest of the images were used for test and evaluation. We also tested our algorithm without using graph for bounding box merging. The results can be seen in Table 1. In this dataset our algorithm always worked perfectly, however if we disable the bounding box merging via graph method, some images will give false results. The reason is simple; non-maximum suppression algorithm does not work well in the case when lots of bounding boxes are computed, hence generating some false bounding boxes. Examples of detection can be seen in Figure 5.

Regarding to the time complexity, the most time consuming part of our method is the feature extraction.

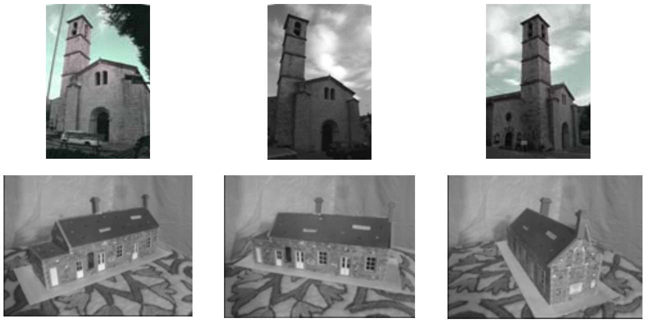

Figure 4: Three training images from “Valbonne Church” and “Model House” databases. The first only, the first and the second only, and finally all of the three were used as training images.

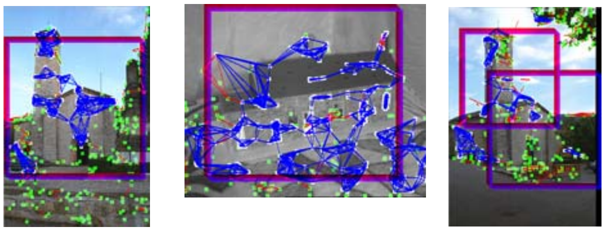

Figure 5: Three results. The last image was false detection.

Test image	MH,+gm	MH,-gm	VC,+gm	VC,-gm
One	100%	100%	100%	86%
Two	100%	100%	100%	93%
Three	100%	100%	100%	93%

Table 1: Performance results of our algorithm on "Model House" (MH) and "Valbonne Church" (VC) multi-view databases. Each row shows the number of training images. “+/-gm” shows the usage of graph based bounding box merging method.

4. CONCLUSION

A new 3D object duplicate detection and localization algorithm was presented in this paper. The query is specified by images showing different views of a given object. Our approach uses graph model to avoid building an explicit 3D object model. This approach was shown to be robust when using only one or few images for training. Moreover, it was successfully tested for object duplicate detection, even when the object is captured from different views.

As future work, we will explore the extension of this method to deal with duplicate object detection in video.

5. ACKNOWLEDGEMENTS

This work was supported by the Swiss National Science Foundation Grant ‘‘Multimedia Security’’, number 200020-113709, partially supported by the European Network of Excellence VISNET II (IST Contract 1-038398), and the PetaMedia (FP7/2007-2011).

6. REFERENCES

[1] J. Zhang and M. Marszalek and S. Lazebnik and C. Schmid, “Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study,” in *International Journal of Computer Vision*, 2007, vol. 73, no.2, pp. 213–238.

[2] Felzenszwalb, P. and Mcallester, D. and Ramanan, D., “A Discriminatively Trained, Multiscale, Deformable Part Model,” in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) Anchorage,* Alaska, June 2008

[3] Josef Sivic and Frederik Schaffalitzky and Andrew Zisserman, “Object Level Grouping for Video Shots,” in *International Journal of Computer Vision*, 2006, vol. 67, no. 2, pp. 189–210.

[4] Fred Rothganger and Svetlana Lazebnik and Cordelia Schmid and Jean Ponce, “Segmenting, modeling, and matching video clips containing multiple moving objects,” in *In Conference on Computer Vision and Pattern Recognition*, 2004, pp. 914–921.

[5] Mikolajczyk, K. and Tuytelaars, T. and Schmid, C. and Zisserman, A. and Matas, J. and Schaffalitzky, F. and Kadir, T. and Van Gool, L., “A Comparison of Affine Region Detectors,” in *International Journal of Computer Vision*, 2005, vol. 65, no. 1–2 pp. 43–72.

[6] David G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” in *International Journal of Computer Vision*, 2004, vol. 60, no. 2 pp. 91–110.

[7] Neubeck, A. and van Gool, L., “Efficient Non-Maximum Suppression,” in *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition*, 2006, pp. 850–855.

[8] Leibe, Bastian and Leonardis, Aleš and Schiele, Bernt, “Robust Object Detection with Interleaved Categorization and Segmentation,” in *International Journal of Computer Vision*, 2008, vol. 77, no. 1 pp. 259–289.

[9] David Nister and Henrik Stewenius, “Robust Scalable Recognition with a Vocabulary Tree,” in *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006, pp. 2161–2168.

[10] Stephen Warshall, “A Theorem on Boolean Matrices,” in *Journal of the ACM (JACM)*, 1962, pp. 11–12.