3D OBJECT DUPLICATE DETECTION FOR VIDEO RETRIEVAL

Peter Vajda, Ivan Ivanov, Lutz Goldmann, Jong-Seok Lee, Touradj Ebrahimi

Multimedia Signal Processing Group – MMSPG
Institute of Electrical Engineering – IEL
Ecole Polytechnique Fédérale de Lausanne – EPFL
CH-1015 Lausanne, Switzerland
{peter.vajda, ivan.ivanov, lutz.goldmann, jong-seok.lee, touradj.ebrahimi}@epfl.ch

ABSTRACT

Content-based video retrieval has become a very active research area in the last decade due to the increasing number of video shared on social networks such as YouTube and DailyMotion. While most of the content-based video retrieval approaches employ visual low-level features for a global analysis of the video, this paper proposes an object-based retrieval method as an alternative. The goal of the proposed method is to retrieve those key frames and shots of a video that contain a particular object, which is a challenging task due to different viewpoints, illuminations and partial occlusions. In order to increase the reliability for 3D objects, our approach combines viewpoint-invariant region descriptors to describe the appearance of an object with a graph model to describe the spatial layout of the individual regions. Given a query object, provided by the user in form of an image and a region of interest, the system retrieves shots containing this object by analyzing a set of key frames for each shot. The robustness of our approach is demonstrated using a video in which one 3D object is recorded in from different view points and with partial occlusions.

1. INTRODUCTION

In the past few years, sharing photos and video in social networks has become very popular. The number of video grows rapidly on social networks like YouTube[1], DailyMotion[2] and Blip.tv[3]. For instance, 20 hours of video are uploaded to YouTube every minute [1] and DailyMotion contains over 975 million video clips [2]. Therefore, fast video retrieval systems, which allow users to search desired video clips in an efficient way, are becoming increasingly important. Most of the existing popular video search engines rely on text-based annotations and manual descriptions of the video. However, recent developments have shown that content-based video retrieval based on visual features extracted from the video content itself provides a promising alternative. Most of the content-based video retrieval approaches rely on the query by example paradigm where a user is required to provide a query video which is compared to other video in a database. Since a representative query video may not be available, either a single query image or an object of interest may be used to describe the scene.

Initially most content-based video retrieval (CBVR) approaches were based on global representations such as color, texture and motion characteristics. Recent approaches have turned towards object-based representations to facilitate the search of objects or regions of interest within a video. However, the retrieval of objects is a challenging problem because an object's visual appearance may change considerably due to variations in viewpoint, illumination, deformation and partial occlusions. Different approaches have been developed to handle these multiple visual aspects of an object. Sivic and Zisserman [3] propose a method where descriptors are extracted from local affine-invariant regions and quantized into visual words, reducing the noise sensitivity of the matching. Inverted files are used to match the video frames to a query object and retrieve those which are likely to contain the same object. However, this work considers only 2D objects, such as posters, signs, ties, and the front side of clocks, and does not take into account real 3D objects. An extension of this approach by Sivic *et al.* [4] uses keypoint tracking to retrieve different views of the same object and to group video shots based on the objects appearance. The tracked object is then used as an implicit representation of the 3D structure of the objects to improve the reliability of the object recognition. This method has proven to be more effective than a query with a single image, but it requires that all relevant visual aspects of the desired object are present in the query shot, which limits its applicability. The system by Rothganger *et al.* [5] is based on a rigid 3D model of the object of interest which is created from several instances of the object within a single shot. The 3D object model is matched to a shot by reprojecting it to the 2D video. While 3D models provide a more reliable description of a real-world object, their creation re-

[1] http://www.facebook.com
[2] http://www.dailymotion.com
[3] http://www.blip.tv

quires a large number of images from various angles, which may not be available in a query.

In this paper, we go one step further and propose an efficient 3D object-based video retrieval system which requires only a single query image. This work is an extension and adaptation of our previous work on graph-based 2D and 3D object duplicate detection in still images [6, 7]. Given a query image with the object of interest, the proposed system retrieves keyframes with duplicates of that object. Due to invariance of the object duplicate detection approach to minor appearance changes, the retrieved frames usually contain also variations from the object of interest. Therefore the retrieved objects are considered as iterative queries to retrieve object duplicates with larger variations. For example, given the frontal view of the car as the initial query, the iterative query mechanism may retrieve the back side of the car if intermediate views of the car are available.

The remaining sections of this paper are organized as follows. Section 2 describes our iterative approach for 3D object duplicate detection in video. Experiments and results are shown in Section 3. Finally, Section 4 concludes the paper with a summary and some perspectives for future work.

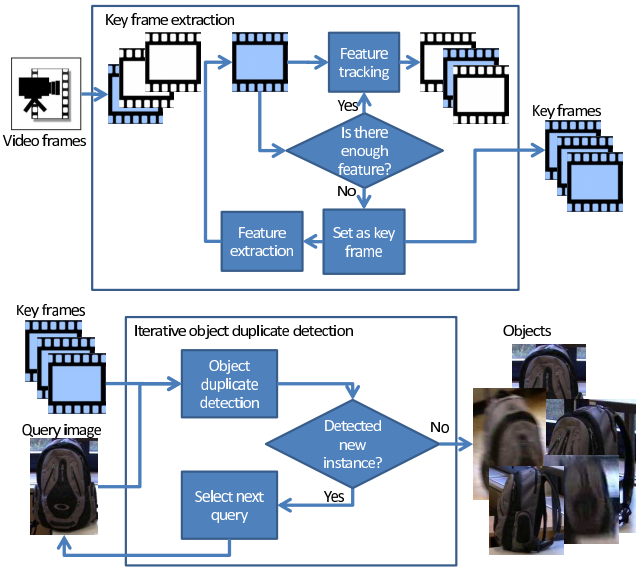

Fig. 1. Overview of the system for object duplicate detection in videos. It runs an iterative search for the target objects on extracted keyframes.

2. PROPOSED ALGORITHM

In this section, we present our solution for object duplicate detection in video clips.

The main innovation is to apply object duplicate detection in an iterative way by considering retrieved objects within key frames as additional queries beside the initial query object. The system architecture which consists of two phases, namely keyframe extraction and iterative object duplicate detection, is illustrated in figure 1. In the proposed system, a user is able to search for video clips containing the desired object by providing a snapshot or photo of that object.

2.1. Key-frame extraction

The goal of the keyframe extraction module is to detect representative frames of the video which contain a considerable change in comparison to the previous keyframe which may correspond to significantly different views or a completely different object or scene. This definition is specialized for object duplicate detection in video and differs from that typically used for keyframes of video shots.

Our approach is to detect stable and robust salient points in the video and to track them using optical flow. Harris corner detection is applied to detect salient points and an iterative Lucas-Kanade method [8] is used to compute the optical flow. If a tracked point disappears in further frames or moves very close to another salient point (≤ 5 pixels), it is considered as lost and not tracked anymore. If the ratio between the number of the tracked points of the previous and the current frame decreases more than a threshold ($T_k = 0.5$), then this frame is saved as a key frame. Otherwise the point tracking continues. This method results in several key frames extracted from a given video, which contain significant changes of the object or scene.

2.2. Iterative object duplicate detection

The goal of the object duplicate detection module is to detect the presence of a target object based on an object model created from a query image depicting this object. Duplicate objects may vary from their perspective, have different size, or be modified versions of the original object.

Our approach for object duplicate detection described in [6, 7] is robust to minor appearance changes, viewpoint variations, and partial occlusions due to the combination of invariant local features and a graph model which describes their relationships. Given a training (query) image, features are extracted and a spatial graph model is created for the object of interest. We use sparse features in order to resolve the localization problem efficiently. These features are robust to arbitrary changes in viewpoint. A spatial graph model is used to improve the detection accuracy, which considers the scale, orientation, position and neighborhood of features. To detect the presence of the object in a test image, the features are extracted and a graph matching algorithm is applied to match the created graph model to these features and derive a matching score. As a result, a match score matrix is produced which represents the pair-wise comparison of training and test images.

The appearance of a 3D object in a video sequence may vary a lot due to different viewpoints and deformations. There-

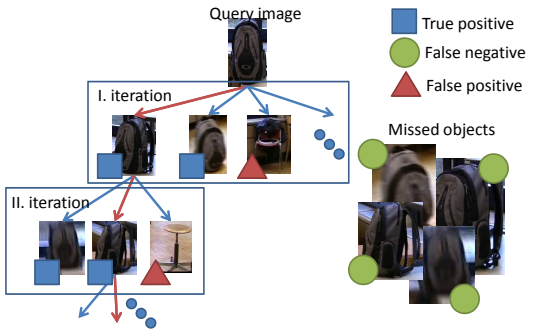

Fig. 2. Illustration of the iterative object duplicate detection on the keyframes of the video. Predicted objects of an iteration are used as query objects in the next iteration.

fore, the detection based on a single query image only may fail at some points due to the large difference between the trained object model and the object present within the considered keyframe. In order to solve this problem, we apply the object duplicate detection method iteratively, as shown in figure 2. In other words, we use a detected instance of the object, which may have a slightly different viewpoint from that of the current query image, as a new query image for the object duplicate detection of the next iteration. At each iteration, we randomly choose one of the objects for the next iteration. In the new iteration, object duplicates are searched only in the key frames that were not predicted to contain the object before.

3. EXPERIMENTS

3.1. Dataset

In order to evaluate the proposed method, a video sequence was recorded in which a "bag" was chosen as the target object. In contrast to the movies used in [4], generating a new video enabled us to have the object in various challenging conditions, such as changes of the background and the distance from the object, different view points, changes in the illumination of the room, and partial occlusions. Some examples of the key frames extracted from the video are shown in figure 3. The movie was recorded in a resolution of 1440×1080 pixels, with the frame rate of 25 fps. It lasts 44 minutes, which makes overall 66000 frames. The object "bag" appears approximately during 40% of the total length of the sequence.

3.2. Experimental setup

Due to the iterative algorithm, the precision rate decreases with the number of iterations (N), while the recall rate increases until the algorithm retrieves all key frames and the recall reaches 100%. We estimate the expected precision as a function of N, and a proper value of N is determined so that

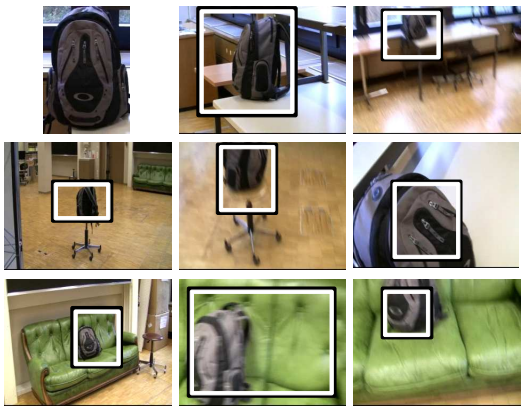

Fig. 3. Representative keyframes containing the object "bag" extracted from the used video. The first image represents the manually selected and cropped training image. Detected objects are marked with bounding boxes.

the expected precision remains higher than a target precision rate $T_{prec} = 0.75$.

The precision can be seen as a probability function, which determines the probability that a randomly selected object among the detected objects is true. If the object duplicate detection algorithm selects a false object as the next query, all the predicted objects will be false in the next iteration. Here, we assume that, in each iteration, the number of predicted objects is the same. Then, the expected precision E_{prec} can be written as

$$E_{prec}(N) = \sum_{i=1}^{N} \frac{1}{N} \cdot P_{odd} \cdot P_{it}^{i-1} \qquad (1)$$

where P_o and P_{it} are the precisions of the object duplicate detection algorithm for the first iteration and the subsequent iterations, respectively. These values can be obtained a priori based on the results in [7]. They are dependent on the threshold parameter that is applied to the matching score between the object model and the query image. We set different values of the threshold for the first and subsequent iterations, i.e., $T_o = 60$ and $T_{it} = 80$, because we target a higher precision for less reliable query objects selected among the key frames. As a result, we obtain $P_o = 85\%$ and $P_{it} = 92\%$. By using these values, the maximum value of N satisfying the inequality $E_{prec} \geq T_{prec}$ is obtained as $N = 3$.

3.3. Results

The precision and recall calculated on key frame level for each iteration are shown in figure 4. Each of the points within the figure represents the result of a single query object which has been either manually or randomly selected. For the first iteration the query object corresponds to the manually selected one (the first image in figure 3) and for the further iterations to

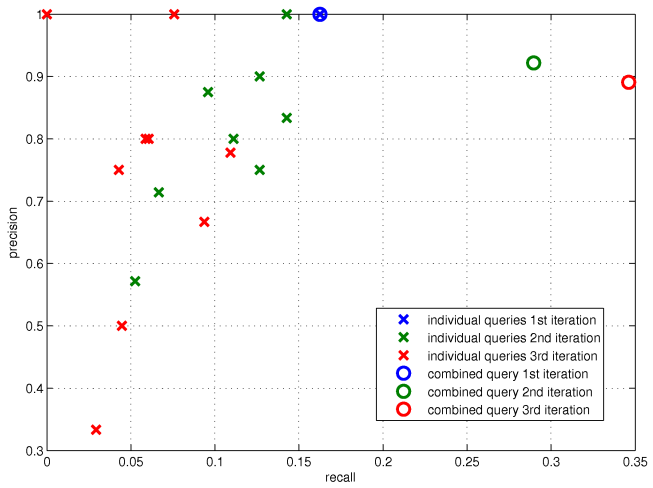

Fig. 4. Precision vs. recall for the individual queries and the overall system for $1-3$ iterations.

those randomly selected from the retrieved objects. Depending on the selected query object the performance within one iteration varies.

The result shows that the first iteration has the highest precision and recall values and the precision and recall tend to decrease through the second and the third iterations. This can be explained by the fact that the bounding box of an automatically detected object is usually less precise than the manually selected one. Therefore it may contain a considerable part of the background, which may lead to false detections in the next iteration. A more precise segmentation of the detected object could solve this problem and improve the performance.

The overall performance of the iterative object duplicate detection is calculated by considering all selected query objects for a certain iteration. This leads to the curve in figure 4 which shows the recall improvement due to the iterations. After the third iteration we obtain a precision of 89%, a recall of 34%, and a F-measure of 50%. As estimated before, the final precision rate remains higher than the lower bound which was set to $T_{prec} = 75\%$.

The dataset contains fast camera movements and thus some of the key frames are blurred. The viewpoints of the object also significantly vary across the whole video. However, our algorithm robustly detects instances of the target object which are blurred or acquired from different viewpoints. Figure 3 shows successfully detected instances of the target object with viewpoint changes of more than 90 degrees, partial occlusions of more than 50% and a large amount of blurring.

4. CONCLUSION

In this work, we have proposed a robust 3D object duplicate detection algorithm for video retrieval. An iterative procedure has been introduced to detect robustly objects in different conditions, such as significant variations of viewpoint, size, lighting conditions and motion blur. The results show that the recall is improved by a factor of 2 using the iterative detection procedure in comparison to the non-iterative object duplicate detection algorithm, while the precision value is kept around 90%.

As future work, we will consider extensive evaluation on more test data. We will also explore the combination of object tracking and the proposed object duplicate detection method. We will consider using precise segmentation of the object detection on video, to improve the accuracy of the algorithm.

5. ACKNOWLEDGEMENTS

This work was supported by the Swiss National Science Foundation Grant "Multimedia Security" (number 200020-113709), the Swiss National Foundation for Scientific Research in the framework of NCCR Interactive Multimodal Information Management (IM2), and partially supported by the European Network of Excellence PetaMedia (FP7/2007-2011).

6. REFERENCES

[1] "YouTube Fact Sheet," Available at: `http://www.youtube.com/t/fact_sheet`.

[2] "DailyMotion Statistics," Available at: `http://www.heroesforhire.info/press/press_release_los_angeles_june_9.htm`.

[3] J. Sivic and A. Zisserman, "Video Google: Efficient visual search of videos," in *Toward Category-Level Object Recognition*. 2006, vol. 4170 of *Lecture Notes in Computer Science*, pp. 127–144, Springer.

[4] J. Sivic, F. Schaffalitzky, and A. Zisserman, "Object level grouping for video shots," *International Journal of Computer Vision*, vol. 67, no. 2, pp. 189–210, 2006.

[5] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce, "Segmenting, modeling, and matching video clips containing multiple moving objects," in *Proc. Int. Conf. CVPR*, 2004, pp. 914–921.

[6] P. Vajda, F. Dufaux, T. Ha Minh, and T. Ebrahimi, "Graph-based approach for 3D object duplicate detection," in *Proc. Int. WIAMIS*, 2009, pp. 254–257.

[7] P. Vajda, L. Goldmann, and T. Ebrahimi, "Analysis of the limits of graph-based object duplicate detection," in *Proc. Int. Symposium on Multimedia*, 2009.

[8] J. Y. Bouguet, "Pyramidal implementation of the lucas kanade feature tracker: Description of the algorithm," Tech. Rep., 2002.