Object Matching Using Feature Aggregation Over a Frame Sequence

Mahmoud Bassiouny*
Computer and Systems Engineering Department, Faculty of Engineering
Alexandria University, Egypt
eng.mkb@gmail.com

Motaz El-Saban
Cairo Microsoft Innovation Lab, Microsoft Research
Cairo, Egypt
motazel@microsoft.com

Abstract

Object instance matching is a cornerstone component in many computer vision applications such as image search, augmented reality and unsupervised tagging. The common flow in these applications is to take an input image and match it against a database of previously enrolled images of objects of interest. This is usually difficult as one needs to capture an image corresponding to an object view already present in the database, especially in the case of 3D objects with high curvature where light reflection, viewpoint change and partial occlusion can significantly alter the appearance of the captured image. Rather than relying on having numerous views of each object in the database, we propose an alternative method of capturing a short video sequence scanning a certain object and utilize information from multiple frames to improve the chance of a successful match in the database. The matching step combines local features from a number of frames and incrementally forms a point cloud describing the object. We conduct experiments on a database of different object types showing promising matching results on both a privately collected set of videos and those freely available on the Web such that on YouTube. Increase in accuracy of up to 20% over the baseline of using a single frame matching is shown to be possible.

1. Introduction

Object instance matching is a key step in many computer vision-based applications such as image search, auto-tagging and augmented reality-type of applications. The state-of-the art approach is to extract a set of interest points on views of objects of interest and then describe these using a geometric and photometric invariant descriptor to be used in matching against captured images. While affine geometric invariance may be achieved using available descriptors such as ASIFT [19], this comes at an extra computational cost and cannot handle large 3D viewpoint changes. Another approach is to store a large collection of views for each object (or synthesize them as in [12]) and use them in matching. This is infeasible in many cases as one may not have full control on database creation. In this paper, we investigate an alternative approach. Instead of taking a single picture of the object to be matched, we capture a frame sequence. The main idea is that while single frame information might not lead to a good database image match, information incrementally aggregated over a time window can lead to better matching probability. Specifically, the main contributions in this paper are:

- Proposing a time incremental object description method by accumulation of individual frame features and using it in performing image matching.
- Investigating and quatitatvely comparing different strategies for incremental object description using experiments on objects of different natures and under different capturing conditions.

To the best of our knowledge, the presented investigations and results have not been reported before in the literature. Besides, it is worth pointing out that the idea of shifting part of the burden of matching a certain object to the user's side is motivated by a number of factors:

- Increasingly, users would be capturing objects with their hand held devices such as mobile devices. These days, those devices are commonly equipped with good quality video cameras.

*The first auhtor performed the majority of the work while being an intern at Cairo Microsoft Innovation Lab.

- Taking a short video of a certain object, possibly from a number of views and aggregating information over time is an easier alternative compared to snapping a picture of the object and waiting for a successful match such as in Google goggles [2] and Bing for Mobile [1]. This process may need to be repeated a few times due to mismatch in viewpoint between the captured image and that in the database as well as light conditions or reflections.

The rest of the paper is structured as follows. Related research work is reviewed in section 2. Section 3 gives an overview of the proposed matching algorithm. In section 4, the proposed sequence-based matching techniques are presented in detail. Experimental results are reported in section 5. Some conclusions and future research directions are given in section 6.

2. Related Work

The presented work in this paper touches on two research areas: a) object instance matching and recognition and b) utilization of video information in matching. Most modern object instance recognition techniques are based on local features matching. The common pipeline is composed of feature detection, description and matching. Many detectors and descriptors have been proposed in the literature (Mikolajczyk *et al*. [17] and Mikolajczyk and Schmid [16] presented a good survey on exiting methods along with experimental evaluation of the different techniques). These surveys have concluded that there is no clear winner detector, nonetheless highest detection scores were obtained by the MSER detector [14] followed by the Hessian-Affine [15]. On the front of descriptors, Mikolajczyk and Schmid [16] reported on experimental evaluations of different interest region descriptors in the presence of real geometric and photometric transformations. They compared shape context [4], steerable filters [7], PCA-SIFT [9], differential invariants [10], spin images [11], SIFT [13], complex filters [23] and moment invariants [8]. They observed that descriptor ranking in terms of matching scores, is mostly independent of the used interest region detector and that the SIFT-based descriptors performed best. Hence, we adopt a combination of MSER detector and SIFT descriptor in our work.

Object instance matching and recognition techniques build on local features descriptors while putting an emphasis on indexing procedures for scalability purposes. For example, Lowe [13] uses a KD-tree with a best-bin-first modification to find approximate nearest neighbors to descriptor vectors of a query. Sivic and Zisserman [24] describe a text-based approach to object and scene retrieval. In this technique, descriptors extracted from local affine invariant regions are quantized into visual words by K-means performed on descriptor vectors from training frames. Term Frequency Inverse Document Frequency (TF-IDF) is used to rank the search results. In a way, the approach proposed by Sivic et al. [24] can be considered as the opposite of ours. In their case they use a query image from one frame to retrieve from a video database, whereas we supply a video query of an object to match against an image database. In [21], they devised a graph-based connectivity structure to represent different stored views for a given object in the database. They addressed very briefly the idea of using a video to perform object matching. However, they have not investigated how many frames are needed for matching nor their sampling as opposed to our proposed work. Nister and Stewnius [20] proposed a hierarchical TF-IDF scoring using hierarchically defined visual words that form a vocabulary tree. This allows a much more efficient look up of visual words and a larger vocabulary that can be used. Torresani *et al*. [25] described a scalable algorithm for similar image search. It uses a form of fast prefiltering on the database, based on Boolean conjunctions, before applying a more expensive analysis or ranking step.

Literature also includes work related to the usage of video information in object matching and tagging. For example, Mooser *et al*. [18] presented a unified approach for object recognition and tracking using local features and optical flow. Similarly, Sakagaito *et al*. [22] presented a form of simultaneous tracking and recognition but using a global object template, thus would suffer from occlusion. Real time recognition and tracking in [18] is achieved by incrementally extracting keypoints on objects that do match the image database as well as being consistent with object's pose estimated through optical flow tracking. The goal in [18] is to unify the matching and tracking processes in a single approach rather than treating them separately in an effort to equalize computational burden over video frames. This has the benefit of avoiding the need of an expensive real-time matching system in every frame as in Lepetit *et al*. [12] as well as avoiding the case of uneven processing time over frames if one utilizes a matching process on the first frame and then tracking for others [3]. While there are commonalities with Mooser's approach [18], the presented method in this paper is computationally more efficient as it does not require explicit tracking and pose estimation by targeting applications where the object occupies most of the field of view. Besides, Mooser has not presented any quantitative results related to matching accuracy and how it is affected by utilizing multiple frames information or by varying the number of considered previous frames and their sampling.

3. Matching Algorithm Overview

The proposed object instance matching in this paper is based on utilizing sequence information in building key-

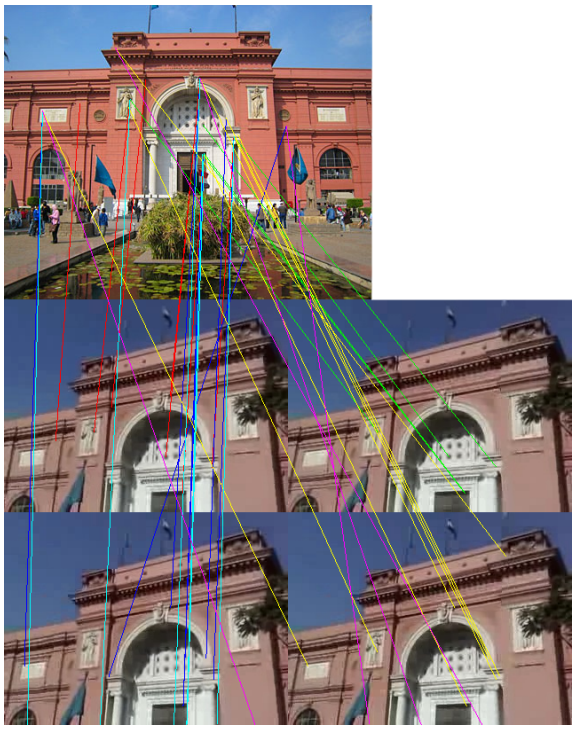


Figure 1. Illustration of the concept of utilizing multiple frame information in image matching. Matches with each video frame are shown in a different color.

points relevant to the object at hand. An illustrating example is shown in Figure 1 where we show four time sampled frames from a captured video of an object. By utilizing information from a set of sampled frames within a time window, one can optimize between aggregating complementary information about a certain object while avoiding excessive redundancy[1] [2]. Colored lines correspond to matching features between each one of the four frames and one specific database image. It is interesting to note that while the four frames are quite similar in content, each has a different and small number of matching features with the database image. Hence, the aggregation of features from multiple frames has the potential of leading to a higher number of matching features with the target database image and hence can boost its ranking in the final list of ranked images.

Another example illustrating the benefit in aggregating features over a time sequence is shown in Figure 2. What the figure shows is that feature aggregation can lead to a higher percentage of correctly matched database images among the top M similar images as compared to just using the current frame keypoints. Specifically, in the case shown in Figure 2 using one or two frames is not enough for achieving database matches compared to the case of using three frames. Another observation is that feature aggregation can alleviate reflection artifacts on objects. This example shows one of the strengths of the approach is that it can accommodate for different lighting conditions each of which may result in a different set of extracted keypoints.

The overall proposed matching algorithm is illustrated in Figure 3. The main idea is in concatenating keypoints from previous frames to the one being considered if a previous frame is *sufficiently* similar to the current one measured using a *suitable* similarity measure. If the similarity is above a certain *threshold*, then the keypoints are passed to the filtering stage that may discard some of the keypoints. In the experiments, we consider multiple alternatives for the filtering stage and evaluate them based on the correct matches among the top ones.

4. Detailed Object Matching Algorithm

4.1. Object Instance Matching

For each frame i, f_i, a set of keypoints are detected using the MSER detector [14]. Let's denote the keypoints set as $K_i = \{K_{i1}, K_{i2}, ...K_{iL_i}\}$ where L_i is the number of keypoints detected in frame f_i. These keypoints are described using the SIFT descriptor [13]. The state-of-the art approach is to match the set of keypoints K_i to a database of features using a K-nearest neighbor (NN) with an efficient indexing scheme such as a KD-tree while using the NN ratio for discarding likely outliers as in [13]. A voting scheme is then applied to identify top M matching images in the database[3]. Consider the j^{th} keypoint K_{ij} in frame f_i, the nearest neighbors in the database matching keypoint K_{ij}, in terms of Euclidean distance, are identified based on normalized score. The matching score of keypoint K_{ij} to the c^{th} nearest neighbor, P_c, is obtained as follows:

$$Score(K_{ij}, P_c) = \sqrt{\frac{\|desc(K_{ij}) - desc(P_1)\|}{\|desc(K_{ij}) - desc(P_c)\|}} \quad (1)$$

where $desc(X)$ is the SIFT descriptor [13]. Based on (1), images in the database are ranked by aggregating the scores of their individual keypoints.

The proposed contribution in this paper is to incorporate previous frames keypoint sets K_{i-1} through K_{i-N}, up to a maximum adjustable time window N in matching for frame f_i. However, it is not favorable to include all keypoints from a previous frame for two reasons: a) a previous

[1] A number of experiments were conducted for different sampling rates and effective time window for information aggregation.

[2] For a sampling rate and time window length, frame ordering is irrelevant. However, the fact that these frames come from a single short video sequence is important (rather than a collection of separate images) to avoid frames depicting different objects.

[3] A geometric verification step is usually applied on the top matches. In this paper, geometric verification has not been used in aggregating keypoints over frames. However, its inclusion is independent of the proposed techniques.

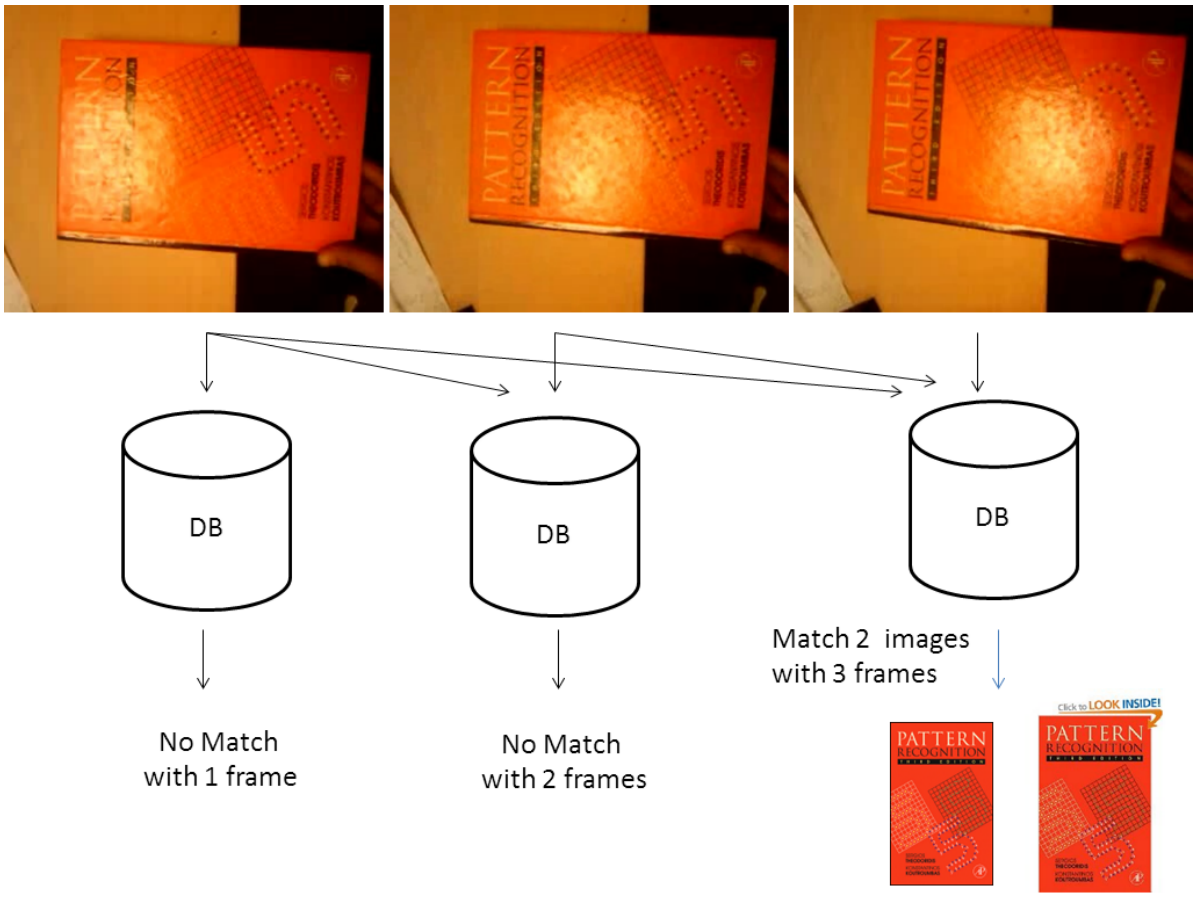


Figure 2. Utilizing previous frames keypoints can lead to higher precision matching in the top M results.

frame can be visually different from f_i and b) a previous frame may be similar but some of its keypoints may be just outliers occurring on the background for example. For the first reason, we propose using a fast similarity measure between frame f_i and a frame f_{i-h} and propagate keypoints K_{i-h} to frame f_i if and only if the measured similarity is beyond a threshold th[4]. The similarity measure used in this paper is the normalized intensity histogram similarity as it is fast enough and leads to reasonable results:

$$Sim(f_i, f_{i-h}) = 1 - (|H(f_i) - H(f_{i-h})|) \qquad (2)$$

where $H(f_i)$ is the normalized grayscale histogram for frame f_i obtained using uniform quantization.

For the second reason, when two frames f_i and f_{i-h} are deemed similar enough, we seek to propagate a subset P_{i-h} of keypoints which matches the top M images in the database. P_{i-h} is a filtered version (see Figure 3) of K_{i-h} and its cardinality $|P_{i-h}|$ will be variable depending on the number M of top matches considered. M can be fixed or adaptive based on the similarity score of the query image to the database images.

[4]Different th values were tested and a value of 60% was found reasonable.

4.2. Keypoint Filtering

We propose two different filtering schemes: a) select to propagate the keypoints from a previous frame, f_t, matching keypoints in the top M images, or b) according to a threshold on the database matching score. In the second case, M would be variable and will be determined using a ratio test comparing the matching image with respect to the highest matching image. For example, if for a previous frame, f_t, there are 10 matching images I_1 through I_{10} with scores S_1 through S_{10} and keypoints sets K_1 through K_{10}, then a specific keypoint set K_q, $1 \leq q \leq 10$ is propagated to the current frame f_i if S_q satisfies:

$$S_q > S_1 R, \quad where \quad 0 < R < 1 \qquad (3)$$

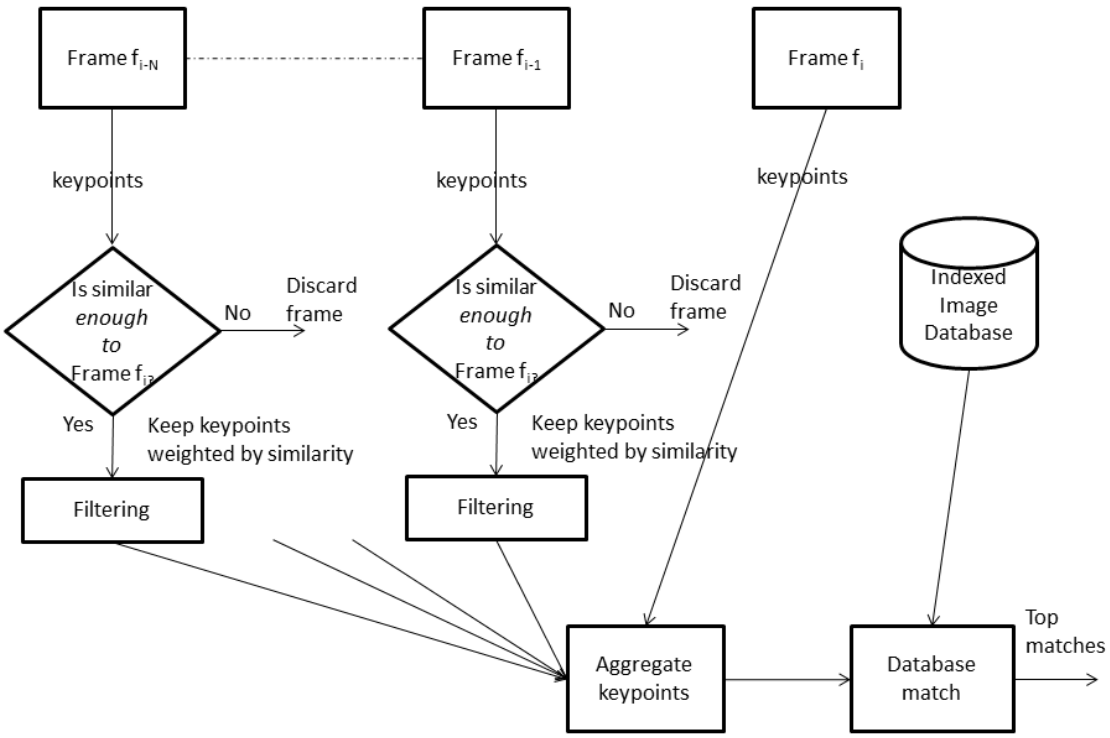


Figure 3. Overall proposed matching algorithm using previous frames information up to a maximum time window of N frames. Previous frames keypoints are aggregated subject to being similar *enough* to the current frame f_i.

5. Experiments

5.1. Dataset Description and Evaluation Metrics

Our proposed algorithm was tested on different objects captured in a set of videos under different conditions on a moderately sized database of about $5,000$ image from 250 objects whose images were collected from a commercial image search engine[5]. In total, there were seven videos which cover 2D as well as 3D objects, whose description are given in Table 1. Two of the videos were obtained from the YouTube video sharing site and thus illustrates the case of videos commonly captured by regular users to remove any authors' bias in the results. Experimental results for the authors' collection of videos and those from YouTube were consistent.

We evaluated our proposed algorithm using the precision and recall metrics. For a given frame, f_i, in a test video, let G be the ground truth set of similar database images and L be the proposed technique result set; precision and recall are defined as:

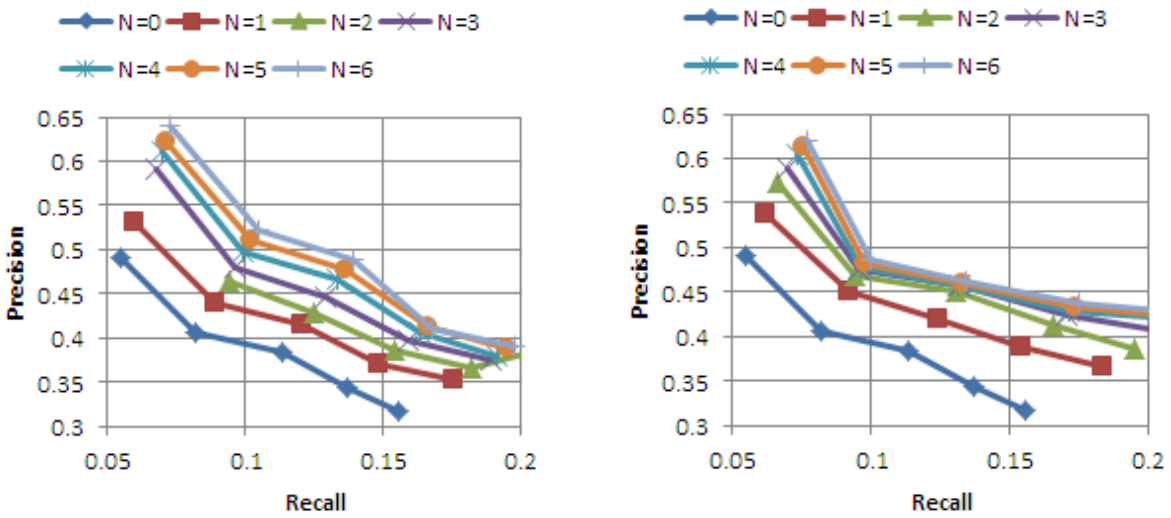
$$Precision = \frac{|L \cap G|}{|L|}, Recall = \frac{|L \cap G|}{|G|} \quad (4)$$

We evaluated the proposed algorithm while varying different parameters: time window size, N and sampling rate, S, i.e. sampling the past frames every S frames. Different N and S values have been experimented with. However, in order to avoid a prohibitively time consuming task, we have employed a cascade approach in the evaluation. Basically, we test using different values of N while restricting S to a small set of values and then vice-versa. Besides, we performed an evaluation of two filtering schemes: a) propagate the keypoints from a previous frame, f_t, matching keypoints in the top M images, or b) according to a threshold on the database matching score based on a ratio R.

[5]We may also consider some of existing video databses such as the CamVid labeled video database of semantic objects for testing the concept of multi-frame matching [6] [5]. However this has been left for future work.

Code	Data Set Description	# Frames
BOOK1	A video of a book testing different illumination conditions and lighting reflections as in Figure 2.	300
BOOK2	A video of a book testing different view-points, scales and orientations.	100
BUIL1	A video of a building, Egyptian museum, as an example of a landmark object.	150
BUIL2	A video of a building, Egyptian museum, testing different view points and orientation.	200
MED	A video of a medicine box under different view-points and scale variations	350
PAINT1	A video of a painting.	500
PAINT2	A video of a painting partially occluded in the majority of the frames.	260

Table 1. Test videos used in the evaluation



(a) (b)

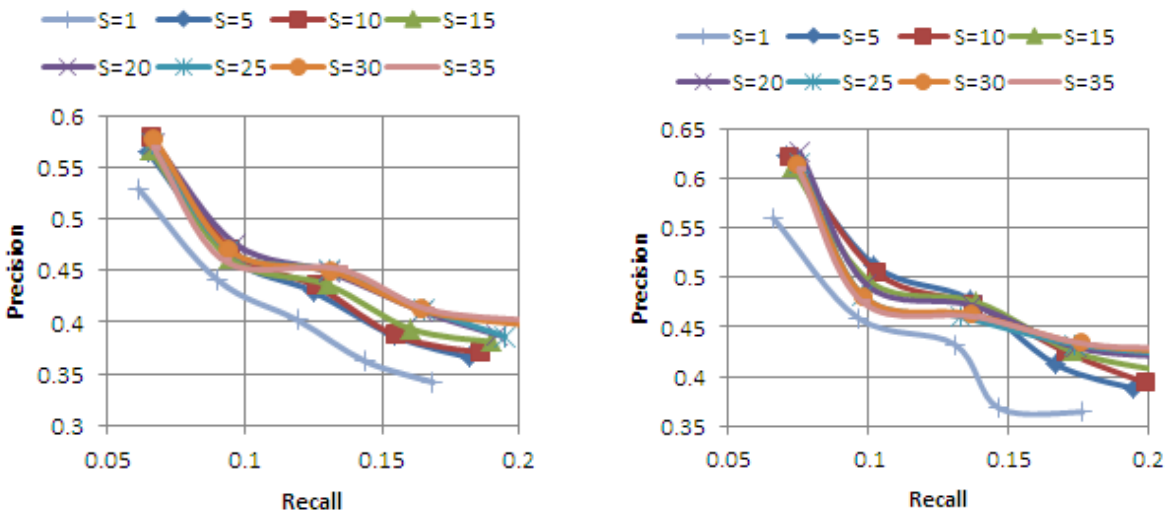
Figure 4. Effect of different time window sizes N when sampling rate: (a) $S = 5$ (b) $S = 25$.

5.2. Matching Results

Figure 4 presents the average Precision-Recall graph on the seven test videos, the graph illustrates that the algorithm based on increasing the window size, N, outperforms the classical approach using a single image. Besides, it shows that the window size is useful up to some threshold. When N gets large, the performance degrades. This is expected as frame correlation becomes weaker when N gets larger and one would risk accumulating noisy features from previous frames.

Figure 5 illustrates the effect of changing the frame sampling rate S on matching effectiveness. Increase in S leads to an increase in matching quality. This can be justified as follows. Between two consecutive frames, the similarity is very high leading to high redundancy in detected keypoints. Whereas, if we skip frames in between ($S > 1$), then keypoints aggregation over time can be more beneficial and thus leads to a better matching. However, increasing S excessively can decrease the effectiveness in matching (as in the case of $S = 35$) as the time window becomes unnecessarily long and can hurt precision/recall. Besides, the results suggest a relation between the optimal value of S and the selected value of N. If N is small, then one has to rely more on increasing S and vice-versa.

In Figure 6, we study the effect of filtering keypoints propagated from previous frames by either: a) propagating keypoints matched in the top M database images or b) propagating keypoints from top matched images based on matching score ratio R to the first match. From Figure 6.a, a good compromise M value is around 5 as it does not lead to an excessive accumulation of keypoints from previous frames which raises the matching computational complexity. For the ratio test, Figure 6.b suggests the presence of a tradeoff between using a small ratio value and a large one.



(a) (b)

Figure 5. Effect of different sampling rates S when window time size: (a) $N = 2$ (b) $N = 5$.

On one hand, if we use a small R value, we risk including all the keypoints matching database images with some of them being noisy matches as well as increasing the computation time unnecessarily. On the other hand, a large R value will mean that we are very reluctant in including keypoints from previous frames and hence previous frame information will not benefit the matching. From Figure 6.b, a compromise R value was experimentally found to be around 0.2. A final note worth making is that the results in both 6.a and 6.b reveal little difference in adopting one measure over the other in filtering keypoints for propagation across frames.

6. Conclusion and Future Work

In this paper, we have presented a novel algorithm aiming at exploiting time sequence information in matching objects captured in a video to a database of images for the purpose of object instance identification. The main idea behind the algorithm is relatively simple, yet the experimental results are very encouraging. We have investigated different choices in incorporating previous frames information such as the number of previous frames considered, their sampling rate as well as different keypoints filtering alternatives. The results were reported on a moderately sized database of images. Though the proposed algorithm is likely to improve on the traditional single frame matching approaches even for larger database sizes, it would be interesting to see how the performance is affected on larger sets. Another point worth the investigation is the computational complexity (space and time) involved in the usage of previous frames information specially when the time window is allowed to be large. Additonally, a quantitative comparative study with related methods[6] [18] [22] would be beneficial. Finally, the proposed algorithms can also be tested when used with quantized features in the currently popular framework of visual words [20] [24].

References

[1] Bing for mobile. http://www.bing.com/community/blogs/search/archive/2010/07/27/camera-scanning-on-the-iphone-app.aspx.

[2] Google goggles. http://www.google.com/mobile/goggles/text.

[3] S. Avidan. Ensemble tracking. *TPAMI*, 29(2):261–271, 2007.

[4] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *TPAMI*, 24(4):509–522, 2002.

[5] G. Brostow, J. Fauqueur, and R. Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters 2009*.

[6] G. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. In *ECCV*, 2008.

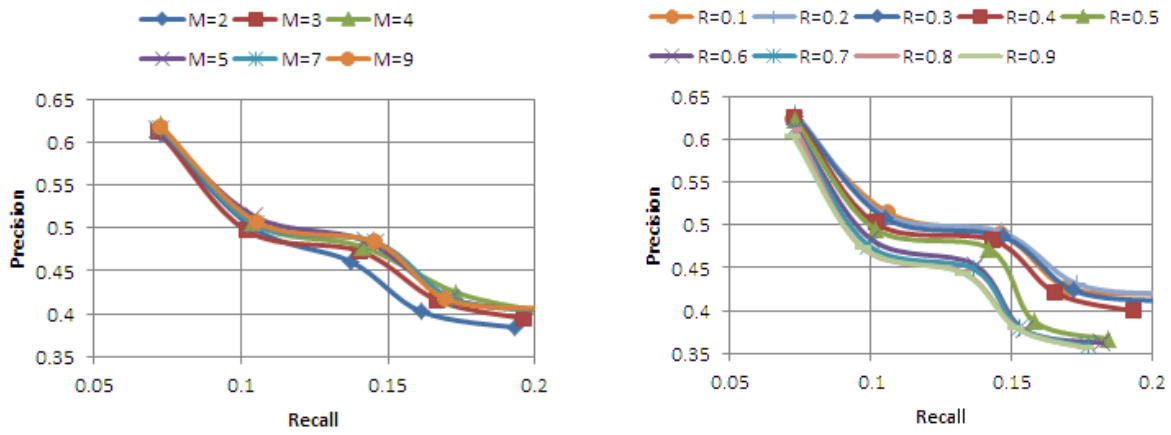
[7] W. Freeman and E. Adelson. The design and use of steerable filters. *TPAMI*, 13(9):891–906, 1991.

[8] L. V. Gool, T. Moons, and D. Ungureanu. Affine photometric invariants for planar intensity patterns. In *ECCV*, 1996.

[9] Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *CVPR*, 2004.

[10] J. Koenderink and A. van Doorn. Representation of local geometry in the visual system. In *Biological Cybernetics*, pages 367–375, 1987.

[6] It is worth mentioning that these methods do not exactly address the same problem we consider in this paper.



(a) (b)

Figure 6. Effect of filtering keypoints from top matches, $N = 4, S = 10$, using: (a) Top M matches (b) Threshold based on ratio R.

[11] S. Lazebnik, C. Schmid, and J. Ponce. Sparse texture representation using affine-invariant neighborhoods. In *CVPR*, 2003.

[12] V. Lepetit, P. Lagger, and P. Fua. Randomized trees for real-time keypoint recognition. In *CVPR*, 2005.

[13] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2(60):91–110, 2004.

[14] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767, 2004.

[15] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *IJCV*, 60(1):63–86, 2004.

[16] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *TPAMI*, 2005.

[17] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. A comparison of affine region detectors. *IJCV*, 2006.

[18] J. Mooser, Q. Wang, S. You, and U. Neumann. Fast simultaneous tracking and recognition using incremental keypoint matching. In *Proc. 3DPVT*, 2008.

[19] J. Morel and G.Yu. Asift: A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences*, 2(2), 2009.

[20] D. Nister and H. Stewnius. Scalable recognition with a vocabulary tree. In *CVPR*, 2006.

[21] H. Noor, S. Mirza, Y. Sheikh, A. Jain, and M. Shah. Model generation for videobased object recognition. In *ACM MM*, 2006.

[22] J. Sakagaito and T. Wada. Nearest first traversing graph for simultaneous object tracking and recognition. In *CVPR*, 2007.

[23] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets. In *ECCV*, 2002.

[24] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *ICCV*, 2003.

[25] L. Torresani, M. Szummer, and A. Fitzgibbon. Learning query-dependent prefilters for scalable image retrieval. In *CVPR*, 2009.