IMPROVING AUGMENTED REALITY APPLICATIONS WITH OPTICAL FLOW

Kyriakos Herakleous and Charalambos Poullis

Immersive and Creative Technologies Lab,
Cyprus University of Technology

ABSTRACT

This paper presents an augmented reality application framework which does not require specialized hardware or pre-calibration. Features extracted, using SURF, are matched between consecutive frames in order to determine the motion of the detected known object with respect to the camera. Next, a bi-directional optical flow algorithm is used to maintain the performance of the system to real-time. The system has been tested on two case studies, a children's book and advertisement, and the results are reported.

Index Terms— Augmented reality, marker-less tracking

1. INTRODUCTION

Recently, there have been considerable advancements in computer graphics and computer vision, both in terms of better performing and cheaper hardware as well as novel optimized algorithms. In particular, the area of augmented reality has advanced considerably and new applications have started to emerge. Augmented reality has already been successfully used in applications such as training and assisting to maintenance [1], military training [2], education [3], [4], and many more.

Despite the reported success applications following the augmented reality paradigm, there is still considerable burden in terms of the requirements and pre-configurations. The majority of augmented reality applications require high performance hardware in order to operate properly. Moreover, most applications require that the user performs a pre-configuration step involving a calibration of the camera prior to its use. Failure to calibrate the camera often results in misalignments when augmenting the scene with virtual objects.

In this paper, we propose an augmented reality system which works with commodity hardware and does not make any assumptions about the users' level of knowledge. Firstly, the proposed system employs the state-of-the-art algorithm SURF [5] for feature extraction and matching, in order to track features through sequential images described in Section 4. Secondly, a bi-directional optical flow algorithm is employed which significantly improves the performance of the system and maintains a real-time frame rate on commodity hardware, and is described in Section 6.

2. EXISTING WORK

A plethora of work has already been reported on augmented reality and its applications. In [6] the authors propose an augmented reality system which employs 3D model-based tracking in order to detect and track the 3D structure of the object in the scene as opposed to its image. [7] Present a marker-less augmented reality application based on SIFT [8] features with a reported frame-rate of 4-5. Similarly, [9] focuses specifically on the archaeological outdoor scenes and proposes a marker-less tracking system which is trained with environmental images.

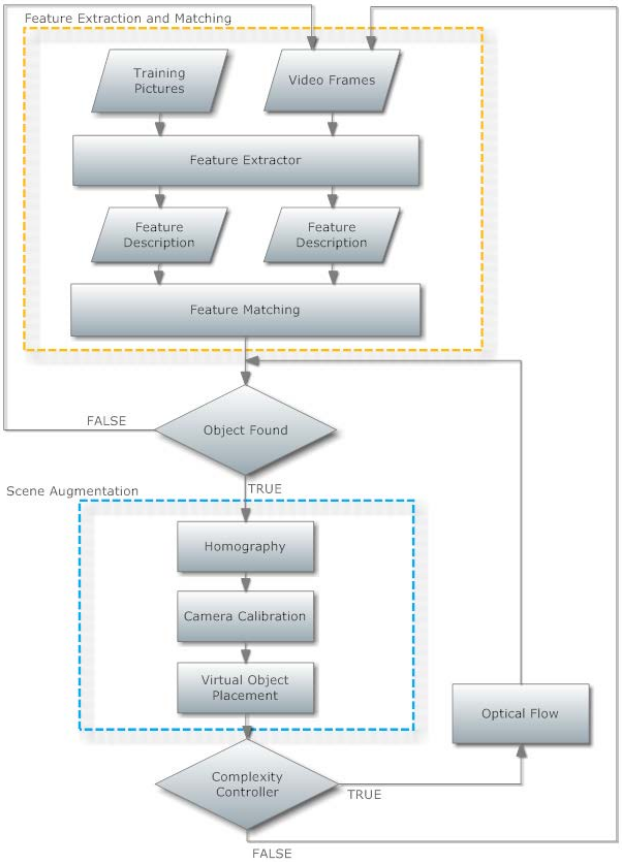


Fig. 1. System overview diagram

3. SYSTEM OVERVIEW

The proposed system comprises of three components as depicted in the system overview diagram in Figure 1. Firstly, features are extracted using SURF and then matched, as described in Section 4. Secondly, using tracked features the spatial relationship between consecutive frames is estimated and used to place virtual objects in the scene, as described in Section 5. Lastly, a complexity controller which records the frame rate determines whether an

optical flow variant is employed, in order to maintain the real-time performance of the system, as described in Section 6.

4. FEATURE EXTRACTION AND MATCHING

Augmented reality applications require the calculation of the camera motion in relation to the scene which is achieved by detecting a known object in the scene. This section describes the steps required for training the system with the known objects and then detecting the objects in the images captured by the camera, which may appear in many configurations with different orientations, scales, and occlusion levels.

For the training phase, two feature detectors/descriptors have been tested, namely SIFT and SURF. Both techniques work by first extracting distinguishable features in a given image and then representing those features as *n*-dimensional vectors. In the case of SIFT, each feature vector has $n = 128$ and in the case of SURF it has $n = 64$. As a result, an image I is described by a set of *n*-dimensional feature vectors f_i with $0 \leq i \leq m$, where m is the number of features detected in the image.

The feature extraction is performed on an image I_{object} containing the known object as well as the images captured by the camera I_{camera} and their features are compared in order to determine whether the known object appears in the captured image.

In this work we have compared both aforementioned feature extraction methods, SIFT and SURF, and have deduced that both techniques perform quite well. On one hand, SIFT has the advantage of being slightly better in some cases. On the other hand, SURF has the advantage of being much faster, since its feature vectors have less dimensions and it uses integral images. Taking into consideration the fact that the system has to be real-time, thus having to compute 25-30 frames per second captured by the camera, SURF was chosen as the best solution for the feature extraction and matching. A comprehensive comparison and elaborate discussion can be found in [10].

5. SCENE AUGMENTATION

The feature matching process described in Section 4 is repeatedly performed until there is a match between the image captured by the camera and the image of the known object. Upon successful matching, the augmentation of the scene with virtual objects is performed.

Fig. 2. Virtual object placement.

In order to place the virtual object in the scene, the homography and the camera calibration are calculated and are decomposed into the object motion parameters (rotation, translation, scale).

Rotation: The orientation of the virtual object is determined by the camera rotation, which is calculated while calibrating the camera for the current frame.

Translation: The position of the virtual object is calculated in 2D, with the use of homography. Using the correspondence of points between the captured and training images, the translation is estimated as the distance in pixels between the points in the two images.

Scale: For the calculation of the scale factor of the virtual object the following metric is used based on the ratio of the diagonal of the image of the known object, and the diagonal of the object as it appears in the captured image.

Figure 2 shows examples of the virtual object placement using all motion parameters: rotation, translation and scale.

6. IMPROVING PERFORMANCE USING OPTICAL FLOW

Using feature extraction and matching works well, however the performance in terms of frame rate is not real time. Our tests have shown that the best frame rate achieved using SURF for feature extraction and matching was a maximum of 10 frames per second.

For this reason, we propose an improvement to the existing technique using bi-directional optical-flow [11], where the features extracted using SURF are tracked using optical flow in a bi-directional fashion for a consecutive set of images. Firstly, features extracted using SURF in image $I_{camera}(i)$ are detected using optical flow in $I_{camera}(i+1)$. In order to eliminate false positives the same process is reversed so that the features in $I_{camera}(i+1)$ are detected using optical flow in the previous image $I_{camera}(i)$. All features which correspond to the original features and are detected in both directions are used as features to track in subsequent captured images.

7. EXPERIMENTAL RESULTS

The proposed system has been tested on two different scenarios.

A children's book scenario, where children can read an illustrated book in front of a computer and the system would annotate virtual objects on the images captured by the camera. The children can move and rotate the book in order to investigate the virtual objects from different angles, producing a more interactive and immersive way of educating children. As a result, book reading becomes more fun and enjoyable. Frames of the usage of the system, in this scenario, can be seen in Figure 3.

The second scenario focuses on printed advertisements in magazines. The idea is to give the reader the opportunity to interactively explore the advertised products (in our case a car) more closely by visiting the magazine's website and placing the magazine in front of the web camera. The system will annotate the video captured with (1) information about the advertised products and (2) virtual object placements. Figure 4 shows example frames of the car advertisement module.

Unlike other techniques, the proposed system uses marker-less tracking which means that even in the case where the user moves and/or occludes part of the known object it will still perform well. Moreover, the fact that camera calibration occurs at every captured frame it means that the user can move the book freely which in turn will change the focal length of the camera. If camera calibration was required prior to use then this behavior would not have been possible since the focal length would have to

remain fixed for the entire duration; something that cannot be guaranteed in the case where the user is a child.

All experiments where conducted on the same laptop computer with specifications: Intel Core i5 M480 2.67GHz, 4GB RAM with OpenSuse 12.1 64- bit Linux. The web camera used in all experiments is Quickcam 9000 by Logitech.

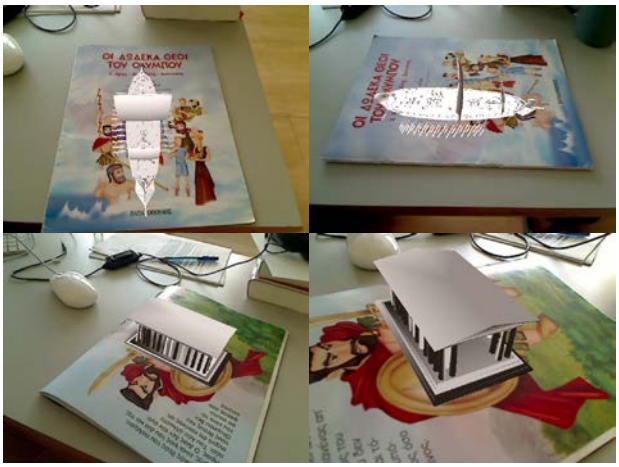


Fig. 3. Scenario I: Children book. The models placed in the captured frames vary according to the page's content. Model complexities: ship – 85200 polygons, Greek temple 4338 polygons.

Fig. 4. Scenario II: Advertising. Model complexity: 29997 polygons.

8. EVALUATION

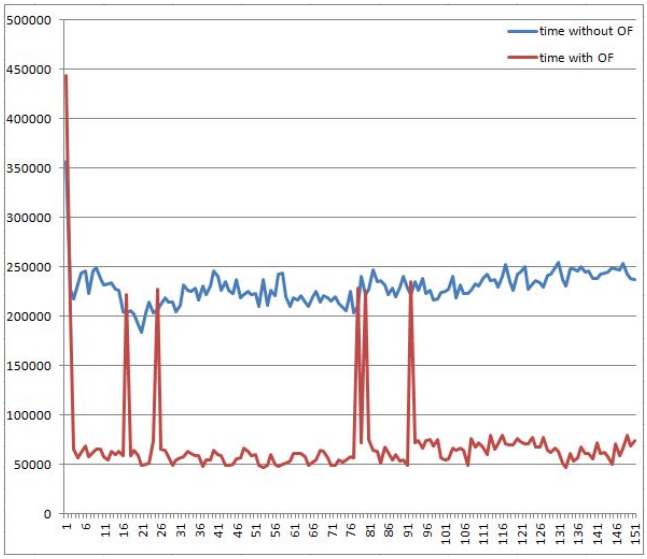
Quantitative evaluation of the system is performed. A video is recorded and used as input in the system with and without using optical flow. The performance of the system is measured in terms of the time needed in milliseconds to process each frame. Figure 5 shows graphs of time needed for each frame of a 150-frame video.

As it can be seen from the blue graphs, in the cases of not using optical flow each frame is processed individually: features are extracted from each frame and matched against the features extracted offline during the training phase. This considerably increases the processing time of each frame.

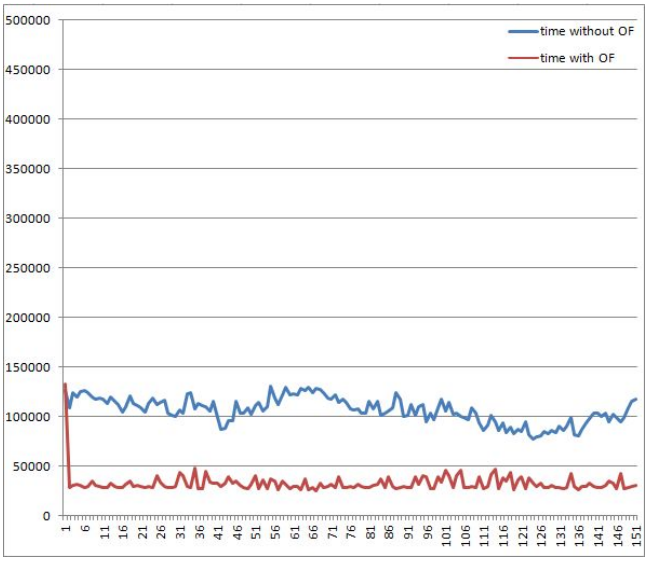
The red graphs show the cases of using the optical flow. In those cases, the first frame is processed normally: features are extracted and matched against the features extracted offline during the training phase. However, for subsequent frames optical flow is used to detect the previously extracted features. Depending on the motion of the camera and/or known object optical flow may perform satisfactorily.

Yet, problems arise when features tracked by optical flow are occluded or can no longer be found, for example due to extreme angles between the known object and camera. As soon as the number of successfully detected features reaches a predefined threshold of 15, the system reverts back to the feature extraction and matching using SURF followed by optical flow.

Figure 6 shows an example of using optical flow without the bi-directional check. The result is that the features are swept away following the hand's motion. This is resolved with the use of the bi-directional optical flow method as described in Section 6.



(a)



(b)

Fig. 5. Time (in microseconds) vs frames. The video duration is 150 frames. Blue graphs indicate not using optical flow and red graphs indicate the cases of using it, where after the first frame, optical flow is enabled and a considerable improvement in performance is noticed. The size of the images captures by the camera are 640x480. 5(a) Size of training images is 581x800 and number of detected features used for the training is 599. 5(b) Size of training images is 600x900 and number of detected features used for the training is 783.

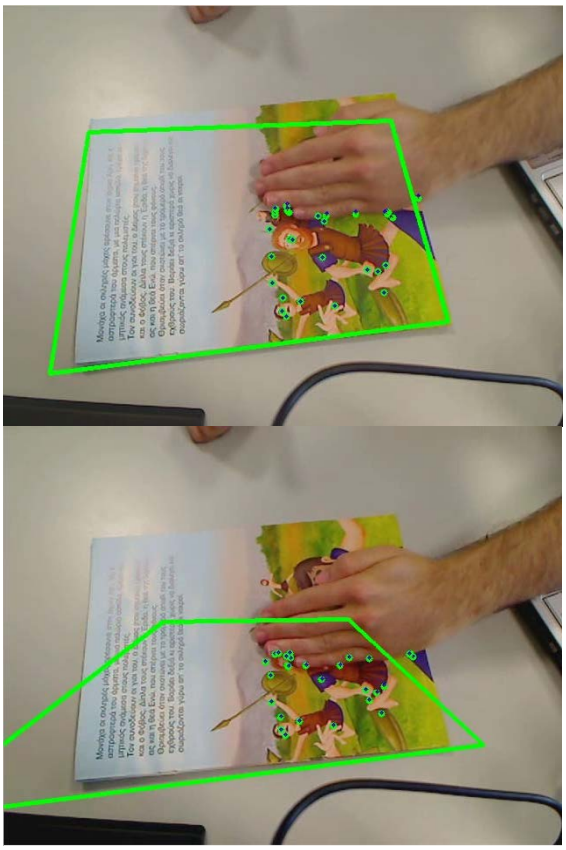


Fig 6. A common problem with optical flow algorithms without the bi-directional check. Tracked features are swept away following the hand's motion. This results in erroneous calculation of the homography as indicated by the green rectangle.

9. CONCLUSION

We have presented a real-time augmented reality system based on marker-less tracking for general purpose applications. The proposed system leverages the strengths of established techniques such as SURF and integrates a bi-directional optical flow algorithm for improving the performance of the system, as well as the results. The proposed system was designed so that it will have minimal hardware requirements and will eliminate the need for pre-configuration steps such as camera calibration. This allows users such as children to be able to use the system without having to be previously trained. Similarly, in the case of advertising anyone with a commodity computer and web-camera can use the system.

A future extension to the proposed system will be the estimation of the existing lighting in the captured image so that the virtual object placed in the scene can be relit so that it naturally matches the existing lighting.

10. ACKNOWLEDGEMENTS

This work was supported by EC FP7 Marie Curie IRG-268256 3DUNDERWORLD – http://www.3dunderworld.org

11. REFERENCES

[1] B. Schwald, "An augmented reality system for training and assistance to maintenance in the industrial context,", Journal of WSCG, vol. 11, Feb. 2003.

[2] S. J. Henderson and S. Feiner, "Evaluating the benefits of augmented reality for task localization in maintenance of an armored personnel carrier turret," *Mixed and Augmented Reality, IEEE / ACM International Symposium on*, vol. 0, pp. 135–144, 2009.

[3] K. Schrier, "Revolutionizing history education: Using augmented reality games to teach histories.," *Unpublished master thesis*, 2005.

[4] H. Kaufmann and D. Schmalstieg, "Mathematics and geometry education with collaborative augmented reality," *Computers and Graphics*, vol. 27, pp. 339–345, June 2003.

[5] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Comput. Vis. Image Underst.*, vol. 110, pp. 346–359, June 2008.

[6] A. I. Comport, E´. Marchand, M. Pressigout, and F. Chaumette, "Real-time markerless tracking for augmented reality: The virtual visual servoing framework," *IEEE Trans. Vis. Comput. Graph*, vol. 12, no. 4, pp. 615–628, 2006.

[7] I. Skrypnyk and D. G. Lowe, "Scene modelling, recognition and tracking with invariant image features," in *ISMAR*, pp. 110–119, IEEE Computer Society, 2004.

[8] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[9] D. Stricker, "Tracking with reference images: a real-time and markerless tracking solution for out-door augmented reality applications," in *Virtual Reality, Archeology, and Cultural Heritage* (D. B. Arnold, A. Chalmers, and D. W. Fellner, eds.), pp. 77–82, ACM, 2001.

[10] L. Juan and O. Gwun, "A comparison of sift , pca-sift and surf," *International Journal of Image Processing IJIP*, vol. 3, no. 4, pp. 143–152,2009.

[11] J.-Y. Bouguet, "Pyramidal implementation of the lucas kanade feature tracker description of the algorithm," Intel Corporation, Microprocessor Research Labs 1999.