The Leiden Augmented Reality System (LARS)

Qi Zhang and Michael S. Lew

LIACS Media Lab, Leiden University, Leiden, The Netherlands
{lim,mlew}@liacs.nl

Abstract. Most augmented reality toolkits require special markers to be used. In our system any designated object in the environment can be used instead of special markers. Furthermore, our system was designed to work with low contrast surfaces (such as the wrinkles on the hands of the users). We used a constellation of maximally discriminative salient points derived from the environment to position a 3D rendered entity. These salient features are combined with local texture to give greater detection stability which results in less jitter to the user. We present a real time system which has sufficiently low computational requirements that it works with typical hardware found on modern laptops, tablets, and smartphones.

Keywords: augmented reality, visual analysis, interfaces, HCI.

1 Introduction

Augmented reality (AR) systems have the potential to change every facet of our daily lives, from reading the news to new interactive search interfaces [4], from playing games to understanding scientific theories and results [1]. Augmented reality seeks to add additional information to reality frequently though visual and/or audio overlays. In this work we demonstrate a research-level tool called LARS, the Leiden Augmented Reality System which is designed to work with low memory devices on common objects in the user's environment including the user's hands.

In most popular augmented reality toolkits, the systems can typically place a correctly oriented and scaled 3D object on a special marker. In the case of the AR Toolkit [1], the markers are binary patterns which are optimized for their visual analysis algorithms.

In many situations, it is not trivial to have the right binary marker on hand. Instead it can be more practical to use any object which is nearby (markerless), from a book, to a cup, or even one’s hand. Related research is ongoing and has used diverse features (i.e. SIFT) [5-9] and techniques [1].

2 Leiden Augmented Reality System (LARS)

To overcome the limitations of binary printed markers, we turn to the paradigm of salient points combined with high performance nearest neighbor algorithms.

A. Fusiello et al. (Eds.): ECCV 2012 Ws/Demos, Part III, LNCS 7585, pp. 639–642, 2012.
© Springer-Verlag Berlin Heidelberg 2012

We designed our system to use two different salient point algorithms: SIFT [2] and MOD [3] and give some comments of our own experiences in using them for real time interaction. Due to space limitations, a thorough introduction and analysis are beyond the scope of this paper.

The SIFT algorithm was designed by David Lowe and has been widely used in object recognition. It is known to be a good salient point algorithm and is probably the most frequently used benchmark for new salient point detectors. It was initially designed to be “shift-invariant” as per the name and has good performance in a wide variety of applications [2]. A recent augmented reality system which uses SIFT points with good results was discussed by Lima, et al. [5].

The MOD algorithm [3] was designed at Leiden University to overcome several challenges in the SIFT approach. First, the SIFT algorithm requires substantial memory per image, from 100KB to 1MB depending on the number of salient points it finds. Second, the SIFT algorithm is a static, generic, grayscale salient point detector. The user can not easily optimize it for particular contexts. An example would be where the user wants to find more salient points on the clouds in an image as opposed to the landscape. The SIFT algorithm will find the salient points close to the high contrast edges in the landscape.

The MOD algorithm allows the user to indicate what the interesting parts of an image are and it will select the best combination of salient and texture features to maximize the discriminatory power of the salient points in that region. For our system, we implemented the Harris corner detector [9], wavelet salient point detection [3], and the SUSAN interest point detector[9]. For representing the local textures, we chose optimized Gabor filters [10], LBP [3], and Laws[10]. The positional surface is found in LARS as follows:

(1) User labels part of an object in his environment as a positional surface
(2) **U** = set of salient positions are extracted from the user labeled region based on all salient point detectors
(3) **B** = set of salient positions not from the labeled region
(4) **M** = Select set of discriminative (based on translation, rotation, and non-marked salient point information) salient and texture features based on **U** as compared to **B** according to the MOD approach [3].
(5) **V** = set of salient points are extracted from each captured frame using **M**
(6) If there is a near planar transformation from **U** to **V,** then mark region with a green rectangle and display the 3D object (Blender or PDB object) with location, pose and scale relative to green rectangle.

3 Experiments

For our experiments, we captured a total of 200 videos: 20 (3 minutes per video for a total of 600 minutes) videos (5 contexts: slow movement, medium movement, fast movement, fronto-planar orientation, oblique orientation; with each context at 4 lighting levels) per object for 10 common office objects (book1 (Spellman Files), book2 (Digital Image Processing), greeting card, CD (the label side), cup, hand, journal, keyboard, desk phone, smartphone). A 5 year old PC (512MB memory, 2.4 Ghz AMD64 X2 processor) and a Logitech Quickcam Pro 9000 webcam were used.

Table 1 displays a comparison between the MOD and SIFT for our system over a set of 10 common office objects (book, phone, cup, hand, etc.). The framerate measures the frames per second of LARS. The Jitter is a typical problem in interactive augmented reality systems and is meant to measure the fine grain error in interactive usage. It was measured using the average pixel distance error of the bounding box for the graphical object from the correct position. The tracking error is meant to capture gross tracking errors and was measured as the probability that the system would make large errors for which we set a threshold at 10%.

Table 1. Comparison between MOD and SIFT in LARS

	Framerate	Jitter Error (% pixel dist.)	Tracking Error
SIFT	14.3	2.79	0.058
MOD	24.9	1.44	0.037

We designed LARS to support two widespread graphical formats: Blender and PDB. The Blender (http://www.blender.org) format is a common 3D graphics format for an open source 3D graphics editing program. Among its features is the ability to take as input dozens of 3D formats and output them to others. The PDB format is a widely used 3D molecular data format. An example of our system displaying DNA using a book as a positional surface (shown as green rectangle) is in Fig. 1.

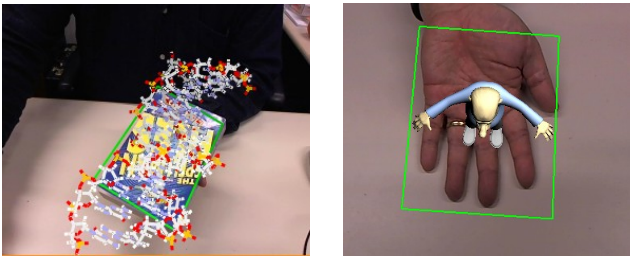

Fig. 1. DNA on a book as positional surface (left) or using a hand (right)

From our experiments, we found that the SIFT algorithm does not find sufficient salient points to allow the stable positioning of the 3D object. The integration of texture features allows low contrast ridges (wrinkles on a hand or texture pattern on a box) to be used as the positioning surface as shown in Fig. 1.

On average over the set of test videos, the MOD approach significantly outperformed SIFT. However, in certain cases both methods performed poorly such as the combination of both low lighting and low contrast objects.

4 Conclusions

Many popular augmented reality toolkits require the usage of special markers for placement of the 3D graphical objects. We presented a real time interactive system which allows the usage of nearby objects to be used to position and orient the augmented reality entities using MOD salient points which integrate texture features to allow low contrast regions to be used as positional surfaces. In our tests, the MOD approach had better jitter and tracking error than SIFT and is currently being actively used for scientific visualization within the Faculty of Science at Leiden University.

Acknowledgments. Leiden University, Helios Technologies and NWO supported this project.

References

1. Fuhrt, B.: Handbook of Augmented Reality. Springer, New York (2011)
2. Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
3. Oerlemans, A., Lew, M.S.: Minimum explanation complexity for MOD based visual concept detection. In: Proc. ACM International Conference on Multimedia Information Retrieval (2010)
4. Thomee, B., Lew, M.S.: Interactive search in image retrieval: a survey. International Journal of Multimedia Information Retrieval 1(2) (2011)
5. Lima, J., Pinheiro, P., Teichrieb, V., Kelner, J.: Markerless Tracking Solutions for Augmented Reality on the Web. In: Proc. of IEEE Symposium on Virtual and Augmented Reality (2010)
6. Lee, A., Lee, J., Lee, S., Choi, J.: Markerless augmented reality system based on planar object tracking. In: Proc. of Frontiers of Computer Vision (2011)
7. Maidi, M., Preda, M., van Hung Le: Markerless tracking for mobile augmented reality. In: Proc. of IEEE Int. Conference on Signal and Image Processing Applications (2011)
8. Cho, H., Jung, J., Cho, K., Seo, Y., Yang, H.: AR postcard: the augmented reality system with a postcard. In: Proc. of ACM International Conference on Virtual Reality Continuum and Its Applications in Industry (2011)
9. Tuytelaars, T., Mikolajczyk, K.: Local Invariant Feature Detectors: A Survey. Foundations and Trends in Computer Graphics and Vision 3(3) (2007)
10. Melendez, J., Puig, D., Garcia, M.A.: Comparative Evaluation of Classical Methods, Optimized Gabor Filters and LBP for Texture Feature Selection and Classification. In: Kropatsch, W.G., Kampel, M., Hanbury, A. (eds.) CAIP 2007. LNCS, vol. 4673, pp. 912–920. Springer, Heidelberg (2007)