Title An AR Benchmark System for Indoor Planar Object Tracking

Authors Wu, Ziming; Guo, Jiabin; Zhang, Shuangli; Zhao, Chen; Ma, Xiaojuan

Source Proceedings - IEEE International Conference on Multimedia and Expo, v. 2019-July, July 2019, article number 8784868, p. 302-307

Version Accepted Version

DOI 10.1109/ICME.2019.00060

Publisher IEEE

AN AR BENCHMARK SYSTEM FOR INDOOR PLANAR OBJECT TRACKING

Ziming Wu[1,*], *Jiabin Guo*[2,*], *Shuangli Zhang*[2], *Chen Zhao*[2] *and Xiaojuan Ma*[1]

[1]Hong Kong University of Science and Technology, China
[2]Netease AR, China
zwual@connect.ust.hk, {guojiabin19, shunley.zhang, zhaoch21}@gmail.com, mxj@cse.ust.hk

ABSTRACT

Planar object tracking (POT) is the basis of many indoor AR applications. However, there still lacks a systematic way to assess AR trackers. Existing benchmarks usually focus on the tracking accuracy of an algorithm without sufficient details about its sensitivity to various object properties and user behaviors, shedding limited light on possible resulting usability issues. We therefore propose a comprehensive POT benchmark system to understand the weakness of a tracker and derive cues for system improvement. We first identify a set of objects that are commonly used as indoor mobile AR markers and specify their vision-related properties. We then construct a video collection to record typical user interactions with these markers, and statistically quantify the consequent changes as a result of individual or multiple basic manipulations. Evaluation shows that this work can expose a tracker's sensitiveness to different object properties and user behaviors, drawing insights for system improvement and algorithm design.

.

***Index Terms*—** planar object tracking, augmented reality

1. INTRODUCTION

The goal of planar object tracking(POT) is to estimate the visual state, *i.e.,* the positions of the four corners, of a planar object in a video sequence given its initial location [1,2]. There have been many POT algorithms employed in different application domains, from augmented reality [1, 3] to robotics [4]. Existing benchmarks for POT algorithms mostly aim to determine the performance of algorithms [1,2,5]. Such a benchmark usually consists of video sequences that capture tracked object(s) in various scenarios, together with groundtruth labels generated either manually or semi-automatically [1, 2]. While enabling direct comparison among different POT algorithms on a single performance measure like accuracy, these benchmarks do not explain why one outperforms another. Moreover, the evaluation on such benchmarks does not establish know-hows for system developers to understand which aspects could be improved.

In the field of object detection, works have been done to provide details rather than a single average performance score by thorough analysis into the categories of objects, such as the ImageNet Challenge [6]. However, to have such a evaluation in AR is more challenging, given an AR system is device-dependent and involves user interaction. The high precision of a POT algorithm might not indicating a good user experience. Perception and cognition issues resulting from the use of systems have been identified in existing studies [7]. For example, system latency can degrade the illusion of stability [8] and perceived jitter has shown negative impacts on users' task performance [9].

Moreover, to designers of AR systems, only knowing which type(s) of usability issues (and to what extent) the POT algorithm can be deployed in their application may proof it inadequate for improving the user experience. They would need to know how the severity of perceptual interference varies in respect to typical user interactions under common conditions, to deliver more informed design decisions in practice. In the context of mobile AR for instance, different POT trackers may have different sensitivities to changes in object illuminance, scale, occlusion, *etc*. during user interaction. If designers could be informed how tracking quality degrades with ranging conditions, they could either improve the algorithm accordingly or constrain usage scenarios to avoid bad cases. Although some of the existing tracking benchmarks *e.g.,* [1, 5] provide videos captured under different scenes and motion patterns, they do not provide sufficient details of these external conditions. Evaluations on these benchmarks can not answer 1) which factor, *i.e.,* target object properties and basic manipulations involved in user interactions, is more likely to cause certain types of perception-focused usability issue, *e.g.,* jitter, stiffness, and error [7, 10]; and 2) how sensitive a tracker is to changes in each factor.

To fill the gap, we propose a new benchmark system for indoor planar object tracking to support perception-focused usability diagnosis and comprehensive comparison across different POT algorithms and/or systems. Our tracking benchmark system mainly comprises three parts: 1)we design and

∗ denotes equal contribution.
Our dataset is available at `https://github.com/NetEaseAI-CVLab/ARBenchmark_PlanarObjTrack`

construct a benchmark dataset in AR scenarios consisting of varying targets and motion. Moreover, we quantify target and motion properties(we call these properties independent variables,*a.k.a.,* IVs) and semi-manually annotate the groundtruth; 2)we summarize four perception measurement metrics(*i.e.,* latency, temporal jitter, spatial jitter, and alignment error) for AR systems to quantify a tracker's performance under our dataset(we call these quantitative results dependent variables *a.k.a.,* DVs); 3)we conduct a sensitivity analysis to understand how a DV responds to the change in an IV and seek to guide the system design and improvements accordingly. We test three algorithms and two systems under our benchmark system to verify its efficacy. Their performance under different IVs measured by the user-center performance metrics is reported. A case study is presented to showcase how these analyses can be used to guide system improvement, demonstrating the value of our proposed benchmark system. The main contributions of this paper are as follows:

- To the best of our knowledge, it is the first benchmark that captures videos in a way which involves possible variations of dedicated factors. It supports comprehensive diagnoses of a tracker's weakness towards common types of markers in natural settings and condition changes caused by user interaction.
- In contrast to previous evaluations that merely emphasize trackers' precision, we incorporate perception-focus metrics to allow comprehensive evaluations from a user-center perspective. Then, through regression analysis, we derive the influence of each IV on DV, to help engineers improve systems accordingly and to help designers have a better understanding of the design space, thereby communicating with users an appropriate affordance of the application.

2. METHOD

Our goal is to diagnose 2D planar tracking systems and provide insights for improvement. To this end, we first build a POT dataset that considers both object properties and user behaviors. Then we evaluate five typical trackers using the dataset and apply unique statistic analysis to understand their affordance to dedicated DVs.

2.1. Dataset Construction

2.1.1. Dataset Design

As taking all real scenarios into consideration is impractical, we summarize the typical properties of objects and user behaviors involved in a POT dataset by surveying previous work and representative applications in real scenarios. We then collect a dataset covering these properties to ensure its generality. For the object, we consider four properties: (1) **Texture Richness**, the complexity of the visual texture of a target image; (2) **Homogeneity**, the similarity of different regions in a target image; (3) **Smoothness**, whether the surface of the object is smooth so that specular reflection will be introduced when being captured by a camera; (4) **Deformation**, whether the object is deformed when used in real scene.

To quantify the object properties, we use the Gray-Level Co-occurence Matrix [11] and the Neighbourhood Gray-Tone Difference Matrix [12] to measure the homogeneity and texture richness respectively for each object. We classify all objects into two groups (i.e. low and high) for these two properties based on the quantified value. For smoothness and deformation, we categorize objects into positive and negative groups based on human observation. In total, we select nine objects to cover the combinations of the four-group properties.

We classify seven types of user behaviors interacting with trackers: (1)**Object Scale**, changing the proportion of the object in the camera image; (2)**In-plane Rotation**, rotating the angle of the camera around its optical axis; (3)**Out-plane Rotation**, the angle between the normal axis of the reference object and the optical axis of the camera;(4)**Speed**,representing the liner velocity of the camera; (5)**Illumination**, the illumination intensity of the environment near a object; (6)**Occlusion**, the proportion of the occluded part in the object; (7)**Out-of-view**, the proportion of the out of image part in the object.

For each object, we record eight videos capturing the following user behaviors interacting with the object: **Far-near movement (FN), In-plane Rotation (IP), Out-plane Rotation (OP), Fast Movement (FM), Illumination Change (ILL), Occlusion (OCC), Out-of-view (OV), Unconstrained (UC)**, as shown in Fig1-1. The first seven videos corresponds to seven user behaviors respectively, whereas the last one is the random combination of multiple user behaviors. We use a typical hand-held mobile device, iPhone 7, to record all the videos. The videos are at 30 fps with a resolution of 1280×720. The process of quantifying user behaviors will be described in Sec2.1.2.

2.1.2. Obtaining the Groundtruth

To attain the groundtruth of our videos, we semi-manually annotate four reference points (four corners of the object) on the target object in each frame. For Occlusion and Out-of-view cases, we use four edges and the middle-corner of the target object to annotate its location in the frame. Besides, for these cases, we annotate the occlusion/out-of-view points for calculating the corresponding occlusion/out-of-view proportion. After annotating the four corners of the target object, we calculate the real 6DOF pose of the camera through PnP [13] with the known actual size of all objects. Then we get the corresponding in-plane angle and out-plane angle directly from the recovered poses. With the known corresponding timestamp, angular velocity and liner velocity can be also measured. We finally collect 72 video sequences, containing 37485 frames in total together with the corresponding groundtruth.

2.2. Performance Metrics

To evaluate a tracker in respect to a specific target and user behavior, we take advantage of usability studies for general AR systems [7, 8, 14, 15] and choose the following metrics related to the performance of mobile AR applications:

- **Tracking Accuracy**: Following [1], we use the alignment error by quantifying the position difference between the four corner points of tracking results and the corresponding Ground-Truth as the tracking error. To avoid the large error resulting from the failed frames, we use area-under-the-curve (AUC) [16] of alignment error to quantify overall tracking accuracy.
- **Latency:** Latency is defined as the time interval between a user query and the system's response. It has an effect on user experience in immersive environments as it can alter the perceived illusion of stability [8], stiffness of virtual objects [17], and the users' sense of presence [15],*etc*.
- **Jitter:** Jitter can be classified into spatial jitter and temporal jitter [10]. Temporal jitter is the latency variation with respect to time. Spatial jitter is perceived as noise produced in the relative tracking result between consecutive frames.

3. EVALUATION

3.1. Selected Algorithms and Systems

Through our benchmark system, we aim to expose a tracking system's sensitivity to various conditions. We choose three state-of-the-art tracking algorithms and two popular commercial AR systems to evaluate our benchmark: **Gracker** [18],**Ferns** [4],**SOL** [19],**ARToolKit**[1],**insightAR**[2]. For the three algorithms (Gracker,Ferns,SOL), we use their open-source codes implemented on a PC platform with the default parameters, and test them on a desktop PC(Intel core i5-6500 CPU). For the two systems (ARToolKit, insightAR), we test them on a selected mobile smart-phone device (iphone 7).

3.2. Result Analysis

3.2.1. Overall performance

The performance of the five trackers measured by the given metrics is shown in Table 1. Among the three algorithms, Gracker shows highestlatency and temporal jitter in all cases, indicating its high and unstable computation cost. This is because Gracker formulates POT tracking as graph-based structure matching instead of direct keypoints matching, thereby has higher computation complexity than the other two feature-based methods. SOL has smaller latency and temporal jitter than Gracker, but still performs worse than Ferns. It deploys an online structured learning scheme to improve keypoints matching. It thus has relatively high and unstable computation complexity as well. Instead, Ferns uses an offline training method, which saves training results offline rather, reducing online computation cost. Though the temporal jitter of ferns is low, it has worse performance in ”OV”,”OCC”,”UC”, possibly due to frequently occurring target lost and re-detection process.

When considering spatial jitter and alignment error, Gracker has stable performance in all cases except ”OV”,”OCC”,”UC”. While showing poorer overall performance, Gracker outperforms others in ”OV” and ”OCC”. Such advantage comes from the graph-based matching method that considers structured information. SOL performs worse in most cases. It is possibly because the binary feature it used can not effectively handle cases like rotation, distortion and blur. Ferns shows the best performance in all cases except ”OV” and ”OCC”. As Ferns always matches under-tracking images with a trained template, the performance degrades rapidly when the target is partial invisible. As for the two systems, insightAR1.0 has smaller latency than ARToolKit while with larger temporal jitter. Regarding the resulted spatial jitter, insightAR1.0 outperforms ARToolKit in all cases except ”OP” and ”UC” but it is worse in terms of tracking accurary.

3.2.2. Sensitivity Analysis

Sensitivity to object properties Understanding the sensitivity of a tracker to objects' properties is critical to AR marker design. Our benchmark provides this value by observing how a tracker responses differently to objects with different properties. In particular, we first estimate a tracker's performance on each object when keeping the other conditions identical. Then for each tracker, we sort the objects based on the tracker's mean performance over the four measurements. The objects are listed in descending order, where the one with the tracker's best performance is at the leftmost as shown in Figure 1-3. The result shows that the five trackers perform differently on the nine objects. As shown in Figure.1-3, Ferns and SOL can better handle the objects like *Book2* and *ARCard2* with high surface smoothness and the two systems can more easily track objects which have deformation like *Newspaper*. To look deeper into the result, we run a Wilcoxon signed-rank test analysis [20] to see whether a tracker's performance on two objects is significantly different. We first normalize all measurements into zero to one and pair the comparison to keep other IVs identical. The paired DVs are then taken as the input of the analysis. Overall, the three algorithms($P_{avg} = 0.041$) are comparatively less sensitive to object properties than the two systems ($P_{avg} = 0.013$) given their higher average P-value.

To allow for detailed analysis of object properties' impact on trackers, we further compare each tracker's performance on two designated settings in which only one of the object properties varies whereas the other IVs are managed to be identical. We choose the pairs of *onmyoji-card*(Low) and *life-poster*(High), *minion-poster*(Low) and *life-poster*(High),

[1] http://www.artoolkit.org
[2] http://dongjian.163.com

	Far-Near(FN)				Fast(FM)				Illumination(ILL)				In-Plane(IP)			
	L	TJ	SJ	AE	L	TJ	SJ	AE	L	TJ	SJ	AE	L	TJ	SJ	AE
AT	6.06	**0.03**	0.10	0.87	6.07	**0.15**	0.10	0.82	6.20	**0.03**	0.08	**0.90**	6.11	**0.03**	0.08	**0.91**
IA	5.87	0.59	**0.08**	0.86	5.94	2.13	**0.09**	0.75	**4.71**	1.89	0.08	0.79	4.68	0.23	**0.07**	0.84
FE	**2.01**	0.48	0.09	**0.89**	**2.47**	6.00	**0.09**	**0.85**	6.13	1.96	0.08	0.85	**2.80**	5.83	0.10	0.89
SO	58.12	38.94	0.12	0.79	46.19	98.46	0.13	0.60	57.61	27.04	0.09	0.80	56.46	32.07	0.11	0.74
GT	780.50	1527.00	0.11	0.78	695.00	2172.00	0.13	0.71	724.00	524.00	**0.07**	0.71	819.40	179.80	0.12	0.76
	Out-Plane(OP)				Out-of-View(OV)				Occlusion(OCC)				Unconstrained(UC)			
	L	TJ	SJ	AE	L	TJ	SJ	AE	L	TJ	SJ	AE	L	TJ	SJ	AE
AT	6.48	**0.02**	0.06	**0.87**	6.40	**0.14**	0.10	0.52	6.40	**0.14**	0.10	0.52	6.12	**0.09**	**0.08**	**0.81**
IA	4.31	1.68	0.08	0.77	**3.89**	1.50	0.10	0.40	**3.89**	1.50	0.10	0.40	6.09	2.06	0.10	0.70
FE	**2.04**	0.17	**0.05**	0.76	8.47	37.03	**0.09**	0.51	8.47	37.03	**0.09**	0.51	**5.70**	25.81	0.01	0.73
SO	55.85	4.76	0.06	0.75	58.85	242.01	0.11	0.48	58.85	242.01	0.11	0.48	56.00	360.8	0.09	0.57
GT	791.90	36.13	0.07	0.80	699.60	6444.00	**0.09**	**0.55**	699.60	6444.00	**0.09**	**0.55**	1563	7859.00	0.13	0.54

Table 1. The summary of the five trackers' performance under different IVs. AT, IA, FE, SO and GE are the abbreviations of ARToolKit, insightAR, Ferns, SOL and Gracker respectively. L($/ms$), SJ($/px$), TJ($/ms$) and AE(/%) correspond to latency, spatial jitter, temporal jitter and the AUC of alignment error respectively.

geography-book(Low) and *onmyoji-card*(High), *Christmas-card*(Low) and *simple-book*(High) to represent the two settings of richness, homogeneity, smoothness and deformation respectively, in a way that the other properties are approximately identical in terms of their numerical value. For each tracker, we report its performance by the four measurements. We then conduct statistical analysis as the process mentioned above. The result is shown in Table.3. It can be found that existing deformation would always degrade trackers' performance. For all the trackers, they have either significant or marginal differences on objects' richness and smoothness. We also observe that both systems are robust to the variation of homogeneity whereas two of the trackers show a significant difference.

Sensitivity to user behaviors To show the sensitivity of a tracker to different users' behaviors, we estimate how a tracker responds to the changes of a designated independent variable using regression analysis. Regression analysis is used to model the relationship among variables,which allows for the analysis of to what extent each IV contributes to the DV [21]Specifically, we take each user behavior as an observed variable and use all observed variables to regress one of the four measurements. Then we can get the weight of each IV in terms of its contribution to the prediction. Empirically, we choose decision tree regression [22] as the solver. We have also tried other regression methods but decision tree regression achieves the best performance in terms of the R2 score.

The results are shown in Figure.1-2. On one hand, the trackers share some common behaviors, e.g., all the trackers show high sensitiveness to out-of-plane rotation but high robustness against the change in illumination. And out-of-plane rotation is the most influential IV for spatial jitter in all cases. It reveals that the trackers can handle well the change of illumination but still need improvement for out-of-plane rotation. On the other hand, the five trackers response differently to different independent variables. For example, latency of Ferns is sensitive to out-of-view motion whereas its alignment error is largely affected by size change. But for ARToolkit, out-of-plane rotation affects the latency and alignment error significantly. Based the identified pattern, we propose specific directions for enhancing system robustness.

		FN	FM	ILL	IP	OP	OV	OCC	UC
L	IA1.0	5.87	**5.94**	**4.71**	**4.68**	**4.31**	**3.89**	5.40	**6.09**
	IA1.1	**5.67**	6.70	5.83	5.50	7.31	4.84	**5.16**	7.65
TJ	IA1.0	0.59	2.13	1.89	0.23	1.68	1.50	1.58	2.06
	IA1.1	**0.21**	**2.05**	**0.86**	**0.08**	**0.18**	**1.35**	**1.14**	**1.61**
SJ	IA1.0	0.08	0.09	0.08	0.07	0.08	0.10	0.12	0.10
	IA1.1	**0.06**	**0.07**	**0.07**	**0.03**	**0.06**	**0.09**	**0.10**	**0.07**
AE	IA1.0	0.86	0.75	0.79	0.84	0.77	0.40	**0.48**	0.70
	IA1.1	**0.87**	**0.79**	**0.81**	**0.88**	**0.87**	**0.50**	0.47	**0.82**

Table 2. Improvement from insightAR1.0 to insightAR1.1

3.2.3. *Case Study For System Improvement*

According to Figure.1-2, we find that the performance of insightAR 1.0 on spatial jitter, alignment error, latency are all sensitive to out-plane rotation. It implies that perspective distortion caused by the out-plane rotation primarily leads to the degraded tracking performance. We thereby propose an improved version insightAR 1.1 to deal with the perspective distortion accordingly. Specifically, during the tracking process, we update the target model with the newly tracked target when the distortion is up to bound. This solution leads a significant improvement on the tracking performance as shown in Table2. The improved version (namely insightAR1.1) performs better than insight1.0 on jitter and error in out-plane and unconstrained cases although the latency is larger. It is likely due to the increased computational complexity. But its latency is still under the bound that users can perceive. Overall, with the cost of a little increased latency, the insightAR1.1 achieves more accurate and smoother tracking performance.

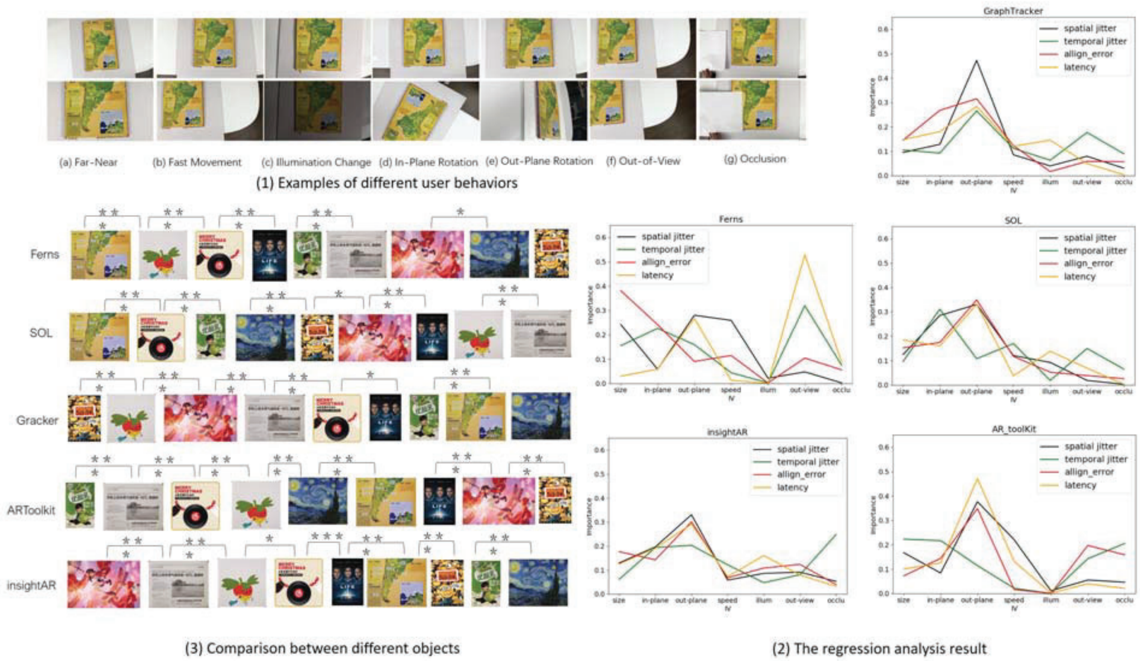

Fig. 1. (1): Examples of different user behaviors. (2): The regression analysis gives the importance of each IV that contributes to the corresponding DV for different trackers. (3): Comparison between different markers. ($* : 0.05 < p < 0.1, *** : p < 0.05$).

4. DISCUSSION AND CONCLUSION

Existing works have analyzed the performance of the selected algorithms, e.g. SOL [18] demonstrated limited performance in texture-less, low illumination and motion blur conditions. While our result is consistent with the previous findings, our analysis uniquely identifies that objects with higher homogeneity can be more easily tracked by SOL. Moreover, we show that SOL is highly sensitive to objects' smoothness and out-of-plane rotation. For Ferns, a previous study revealed its poorer performance under out-of-plane rotation (perspective distortion) than that under scale change or in-plane rotation. We further find that Ferns cannot well cope with out-of-view rotation and occlusion, which indicates its high sensitiveness to size change, objects' smoothness and deformation. Gracker has illustrated its robustness against occlusion, motion blur(fast speed),in-plane rotation and out-plane rotation [18], but we find that it fails to deal with occlusion and out-of-plane rotation on our dataset. While most of the existing evaluations only focus on precision measurement for trackers, our benchmarks provide evaluations from a user-center perspective.

In this paper, we present a benchmark system with a comprehensive dataset to evaluate planar object tracking algorithms/systems. In contrast to previous studies, we aim to enable comprehensive diagnosis of a tracker's weakness and robustness in term of a user-center perspective. We summarize representative object properties and user behaviors that are commonly deployed in AR applications and collect videos to cover the variations of those designated IVs. Three perception-focused metrics are utilized to assess a trackers' performance in terms of user experience. Evaluation shows that our benchmark system can derive the influence of each IV on DV, thus shedding light on system improvement and design.

5. REFERENCES

[1] Pengpeng Liang, Yifan Wu, and Haibin Ling, "Planar object tracking in the wild: A benchmark," *arXiv preprint arXiv:1703.07938*, 2017.

[2] Ankush Roy, Xi Zhang, Nina Wolleb, Camilo Perez Quintero, and Martin Jägersand, "Tracking benchmark and evaluation for manipulation tasks," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 2448–2453.

[3] Georg Klein and David Murray, "Parallel tracking and mapping for small ar workspaces," in *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*. IEEE, 2007, pp. 225–234.

[4] Mustafa Ozuysal, Michael Calonder, Vincent Lepetit, and Pascal Fua, "Fast keypoint recognition using ran-

	Ferns					Gracker					SOL					ARToolKit					insightAR				
	L	TJ	SJ	AE		L	TJ	SJ	AE		L	TJ	SJ	AE		L	TJ	SJ	AE		L	TJ	SJ	AE	
R-	2.01	**0.17**	0.13	**0.96**	*	**461.53**	576.26	0.14	0.86	**	**51.14**	33.24	0.14	0.74	**	6.26	0.06	0.12	**0.94**	*	**9.90**	2.09	0.14	0.88	**
R+	**1.91**	0.19	**0.07**	0.93		1069.90	**32.40**	**0.08**	0.86	*	60.00	**21.44**	**0.12**	**0.94**	*	**6.12**	**0.02**	**0.10**	0.92		10.30	**0.32**	**0.06**	**0.93**	*
H-	2.01	0.24	0.09	**0.95**		**743.77**	**22.43**	0.09	**0.88**	**	64.85	24.37	0.12	0.73	**	6.15	**0.01**	0.10	0.81		10.32	0.37	0.09	**0.95**	
H+	**1.91**	**0.19**	**0.07**	0.93		1069.90	32.40	0.08	0.86	*	**60.00**	**21.44**	**0.12**	**0.94**	*	**6.12**	0.02	0.10	**0.92**		**10.30**	**0.32**	**0.06**	0.93	
S-	2.01	0.17	0.13	**0.96**	**	**461.53**	**576.26**	0.14	**0.86**	**	**51.14**	**33.24**	0.14	**0.74**	**	6.26	**0.06**	0.12	**0.94**	**	9.90	2.09	0.14	**0.88**	**
S+	**1.94**	**0.15**	**0.07**	0.95	*	997.96	5954.66	**0.09**	0.69	*	60.43	61.09	**0.05**	0.71	*	**6.09**	0.07	**0.09**	0.88	*	**9.35**	**1.38**	**0.07**	0.81	*
D-	1.94	**0.15**	**0.07**	**0.95**	**	997.96	5954.66	**0.09**	**0.69**		60.43	61.09	**0.05**	**0.71**		6.09	0.07	0.09	0.88	*	9.35	1.38	**0.07**	**0.81**	**
D+	**1.93**	0.17	0.10	0.92	*	**767.00**	**5163.45**	0.15	0.59		**33.41**	**7.06**	0.13	0.68		**5.88**	**0.03**	0.12	0.88		**7.45**	**1.04**	0.10	0.73	*

Table 3. The table shows trackers' performance given different object properties and statical analysis is deployed to reveal whether there is a significant difference between two settings.($*: 0.05 < p < 0.1, ***: p < 0.05$) R, H, S, and D denotes Texture Richness, Homogeneity, Smoothness, and Deformation respectively.

dom ferns," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 3, pp. 448–461, 2010.

[5] Sebastian Lieberknecht, Selim Benhimane, Peter Meier, and Nassir Navab, "A dataset and evaluation methodology for template-based tracking algorithms," in *Mixed and Augmented Reality, 2009. ISMAR 2009. 8th IEEE International Symposium on*. IEEE, 2009, pp. 145–151.

[6] Olga Russakovsky, Jia Deng, Zhiheng Huang, Alexander C Berg, and Li Fei-Fei, "Detecting avocados to zucchinis: what have we done, and where are we going?," in *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, 2013, pp. 2064–2071.

[7] Zhen Bai and Alan F Blackwell, "Analytic review of usability evaluation in ismar," *Interacting with Computers*, vol. 24, no. 6, pp. 450–460, 2012.

[8] Robert S Allison, Laurence R Harris, Michael Jenkin, Urszula Jasiobedzka, and James E Zacher, "Tolerance of temporal delay in virtual environments," in *Virtual Reality, 2001. Proceedings. IEEE*. IEEE, 2001, pp. 247–254.

[9] Robert J Teather, Andriy Pavlovych, Wolfgang Stuerzlinger, and I Scott MacKenzie, "Effects of tracking technology, latency, and spatial jitter on object movement," in *3D User Interfaces, 2009. 3DUI 2009. IEEE Symposium on*. IEEE, 2009, pp. 43–50.

[10] Aline Normoyle, Gina Guerrero, and Sophie Jörg, "Player perception of delays and jitter in character responsiveness," in *Proceedings of the ACM Symposium on Applied Perception*. ACM, 2014, pp. 117–124.

[11] Robert M Haralick, Karthikeyan Shanmugam, et al., "Textural features for image classification," *IEEE Transactions on systems, man, and cybernetics*, , no. 6, pp. 610–621, 1973.

[12] Moses Amadasun and Robert King, "Textural features corresponding to textural properties," *IEEE Transactions on systems, man, and Cybernetics*, vol. 19, no. 5, pp. 1264–1274, 1989.

[13] Kenneth Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quarterly of applied mathematics*, vol. 2, no. 2, pp. 164–168, 1944.

[14] Mark A Livingston and Zhuming Ai, "The effect of registration error on tracking distant augmented objects," in *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*. IEEE Computer Society, 2008, pp. 77–86.

[15] Michael Meehan, Sharif Razzaque, Mary C Whitton, and Frederick P Brooks, "Effect of latency on presence in stressful virtual environments," in *virtual reality, 2003. Proceedings. IEEE*. IEEE, 2003, pp. 141–148.

[16] Luka Čehovin, Aleš Leonardis, and Matej Kristan, "Visual object tracking performance measures revisited," *IEEE Transactions on Image Processing*, vol. 25, no. 3, pp. 1261–1274, 2016.

[17] B Knorlein, Massimiliano Di Luca, and Matthias Harders, "Influence of visual and haptic delays on stiffness perception in augmented reality," in *Mixed and Augmented Reality, 2009. ISMAR 2009. 8th IEEE International Symposium on*. IEEE, 2009, pp. 49–52.

[18] Tao Wang and Haibin Ling, "Gracker: A graph-based planar object tracker," *IEEE transactions on pattern analysis and machine intelligence*, 2017.

[19] Sam Hare, Amir Saffari, and Philip HS Torr, "Efficient online structured output learning for keypoint-based object tracking," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 1894–1901.

[20] Sidney Siegal, *Nonparametric statistics for the behavioral sciences*, McGraw-hill, 1956.

[21] J. Scott Armstrong, "Illusions in regression analysis," *International Journal of Forecasting*, vol. 28, no. 3, pp. 689–694, 2012.

[22] Jane Elith, John R Leathwick, and Trevor Hastie, "A working guide to boosted regression trees," *Journal of Animal Ecology*, vol. 77, no. 4, pp. 802–813, 2008.