Enhanced Tracking Method with Object Detection for Mixed Reality in Outdoor Large Space

Airi Kinoshita[1], Tomohiro Fukuda[2], Nobuyoshi Yabuki[3]
[1,2,3]Osaka University
[1]kinoshita@it.see.eng.osaka-u.ac.jp [2]fukuda.tomohiro.see.eng@osaka-u.ac.jp
[3]yabuki@see.eng.osaka-u.ac.jp

Mixed-reality landscape simulation is one of the visual methods used in landscape design studies. A markerless tracking method using image processing has been proposed for properly aligning the real and virtual worlds involved with landscape simulations in large spaces. However, this method is challenging because tracking breaks down if a dynamic object is encountered during the mixed-reality execution. In this study, we integrated deep-learning object detection with natural feature-based tracking, which tracks manually defined feature points (tracking reference points), with the aim of reducing the impact of moving objects such as people and cars on mixed-reality tracking. The prototype system was implemented and tracking was performed on pre-recorded video taken outdoors. Performance was verified in terms of the number of errors associated with tracking the reference points and the accuracy of the mixed-reality display results (camera pose estimation results). Compared to the conventional system, our system was able to reduce the influence of moving objects that cause errors when tracking reference points. The accuracy of the camera pose estimation results was also verified to be improved. This research will contribute to developing mixed-reality simulation systems for large-scale spaces that are accessible to everyone, including users in the architectural field.

Keywords: *Landscape Visualization, Mixed Reality, Object Detection, Tracking, Deep Learning.*

INTRODUCTION

When considering landscape design, landscape simulation is an effective visualization tool for helping stakeholders, including non-specialists, understand the plan (Honjo and Lim, 2001). Mixed reality (MR), which merges the real and virtual worlds, has been attracting attention for its use in landscape simulation (Milgram and Kishino, 1994). In this paper, MR specifically refers to a fusion of the real and virtual worlds in three dimensions that is interactive in real-time, such as in augmented reality (AR) as defined by Azuma (1997).

To properly align the real and virtual worlds in MR, the performance of tracking systems in determining the relative position of the camera is important (Pentenrieder, Meier, and Klinker, 2006). Here, vision-based methods for tracking that use information captured by cameras offer the potential for the development of accurate and low-cost MR systems (Neumann and You, 1999). In particular, natural feature tracking, which uses elements already present in the environment, maybe better suited to landscape studies and similar applications than methods that place artificial markers on the target site.

Inoue et al. (2018) demonstrated a robust tracking method that used natural features. In their method, the 3D coordinates of the tracking reference points were manually defined and their positions on the MR execution screen were tracked using image processing technology. Then, the tracking reference points whose tracking error exceeded a certain threshold were deleted as outliers, and the camera pose was computed based on the tracking results of the remaining points. This method has the advantage that the MR system user can arbitrarily define the 3D coordinates of the tracking reference point based on the local topography, thereby allowing information to be accurately registered even for points located far away. However, there is still the problem that tracking may break down if a moving object such as a person or a car enters the frame during MR execution. When this occurs, a larger error is introduced to the position on the screen of any tracking reference point near the moving object.

Therefore, the objective of this study is to integrate deep-learning object detection with natural-feature tracking methods. A prototype system using our proposed method was developed and implemented, and its performance was verified by running it with a video in which a moving object enters the frame. The system detects objects in the scenery captured by the camera that are predicted to move in each frame. Based on the results, it then determines which points to use for pose estimation in natural feature tracking. This reduces the negative impact that moving objects have on tracking and contributes to stabilizing MR in large, dynamic spaces.

In addition, our proposed system uses only images acquired from a monocular camera as input and does not use a model-dependent application programming interface (API) such as ARKit or ARCore. Therefore, it contributes to the realization of a simulation system that is easy for everyone to use, including users in the architectural field.

RELATED WORK

This section summarizes previous research relevant to this study.

Natural feature tracking

In natural feature tracking, tracking is generally performed by extracting characteristic points in an image and using image processing techniques to track the feature points from frame to frame. Thus, artificial markers in the landscape do not get in the way when considering urban design. In this method, the first step is to estimate the 3D coordinates of the feature points and to set the initial camera pose. In subsequent processing, the feature points are tracked on the image so that the camera pose can continue to be estimated. The method of tracking feature points on the image is called optical flow estimation, and typical algorithms for this include the Lucas-Kanade method (Lucas and Kanade, 1981) and the Horn-Schunk method (Horn and Schunk, 1981). The estimation of the camera pose is performed by solving the perspective-n-point (PnP) problem based on the correspondence between the 3D and 2D coordinates of the feature points.

One natural-feature-based method has been proposed that targets images with natural features rather than fiducial markers for detection and tracking. Blanco-Pons et al. (2019) proposed using photos of building facades at various times of the day as targets. This project addressed the problem of image recognition malfunctioning at different times of the day when using outdoor AR. Al-Zoube (2022) proposed a method that uses convolutional neural networks (CNN) to classify and initially align targets, extract feature points from the identified targets, and track them using optical flow estimation. This method involves less processing than the detection of target images by feature matching, and it enables the development of a multi-target AR application that runs in real-time on a Web browser.

On the other hand, another method has been proposed for manually defining the 3D coordinates of feature points to be tracked (tracking reference points) by using the 3D point cloud data of the AR execution target area recovered by structure from motion (SfM, Tomasi and Kanade, 1992), a technique for recovering point cloud data from photographs. In this approach, the random sampling consensus (RANSAC, Fischler, and Bolles, 1981) method is used to remove tracking reference points for which the error in the optical flow estimation is greater than some threshold value, thereby eliminating their effects and increasing the robustness of the system's tracking capability (Sato et al., 2016; Inoue et al, 2018). However, one issue with using this method is that if a moving object, such as a person or a car, intrudes during MR execution, then the tracking may fail because the optical flow estimation errors for the tracking reference points near the object will become large.

Another approach that is based on natural features is to use visual simultaneous localization and mapping (vSLAM), which uses only a camera to perform SLAM for AR and MR. Similarly, the impact that moving objects have on mapping is an issue in vSLAM. Liu and Miura (2021) proposed a real-time vSLAM algorithm that uses semantic segmentation and optical flow estimation to eliminate the influence of moving objects.

PROPOSED METHOD

The execution process of the tracking system proposed in this study can be divided into preprocessing and main processing. Preprocessing defines the tracking reference points and performs the initial camera alignment.

Figures 1 and 2 respectively show a schematic diagram and flowchart of the main process in our proposed system. The process shown by the double line in Figure 2 is an addition to the previous system that was proposed by Inoue et al. (2018)

The first step in the main process is to acquire input from the webcam. Next, object detection is performed on the input image and the results are recorded. Then, we focus on the tracking reference points inside the bounding boxes of the objects detected in the image of the previous and current frame. These are excluded once from the processing that is to be performed later, and the information of their 3D coordinates is saved. By comparing all of the tracking reference points (2D coordinates on the MR execution screen) to the bounding boxes of every detected object (maximum and minimum 2D coordinates), the system determines whether the tracking reference point is inside the bounding box. Next, optical flow estimation is performed for the tracking reference points outside of the bounding box, and their positions on the MR execution screen are tracked frame by frame. The current camera pose is then estimated from the correspondence between the 2D and 3D coordinates of the tracking reference points that remain after excluding those that contained large errors in the optical flow estimation. Finally, from the estimated camera pose and the saved 3D coordinates of the tracking reference points inside the bounding boxes, their 2D coordinates in the current frame are obtained and returned as the target for the tracking process in the next frame.

This process makes it possible to secure the tracking reference point while maintaining the robustness of the tracking system, as it is less susceptible to obtaining errors from moving objects.

IMPLEMENTATION AND VERIFICATION OF THE PROTOTYPE SYSTEM

This section describes the prototype system that was implemented to verify the effectiveness of the proposed method and the details of the verification procedure.

System implementation

For the verification, the main processing for the tracking system was performed in the Unity game engine, and OpenCV for Unity, an asset for image processing that uses the OpenCV functionality in Unity, was employed. The main functions of OpenCV that were used are listed in Table 1.

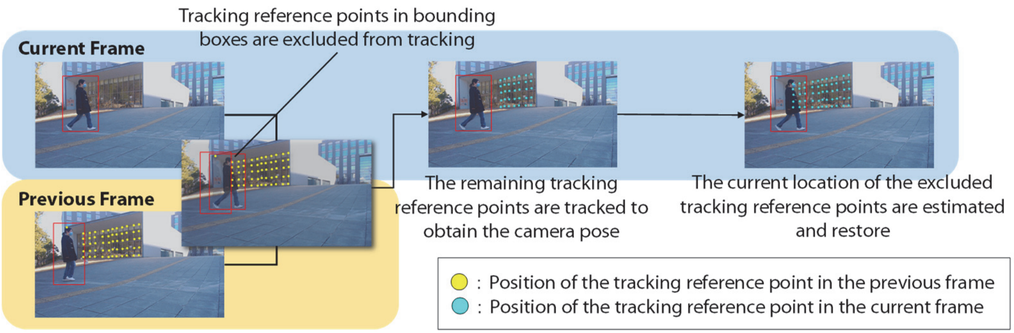


Figure 1
Schematic diagram of the proposed method

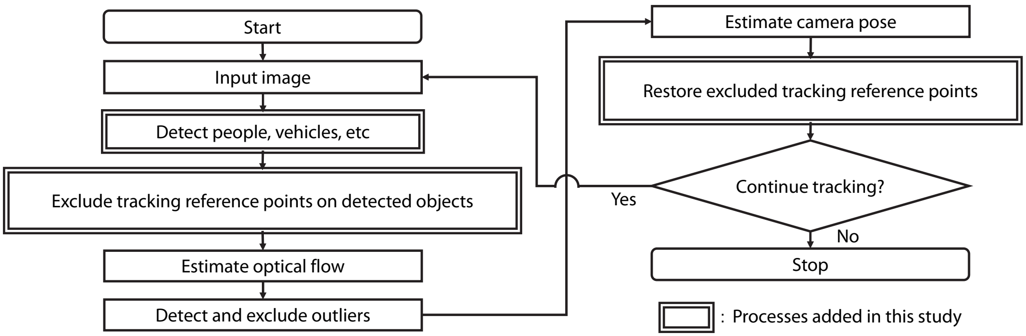


Figure 2
Flowchart of the proposed tracking method

Table 1
The main functions of OpenCV used

Function Name	Purpose of Use in the Proposed System
calcOpticalFlowPyrLK()	Performing optical flow estimation using the Lucas-Kanade method
solvePnPRansac()	Detecting of outliers using the RANSAC method
solvePnP()	Solving PnP problems
cvtColor()	Converting color space

The processes that were involved with object detection were executed in the Python virtual environment. As shown in the flow diagram in Figure 3, camera images that were input into the tracking execution environment were converted into binary data that could be sent and received via the hypertext transfer protocol (http) and were then sent to the object detection execution environment. Object detection was then performed to obtain the class label of the detected object and the coordinates of the two vertices (maximum and minimum) that determine the shape of the bounding box. Finally, the data acquired from the object detection results is returned to the tracking system execution environment in JavaScript object notation (JSON) so that it can be used for tracking.

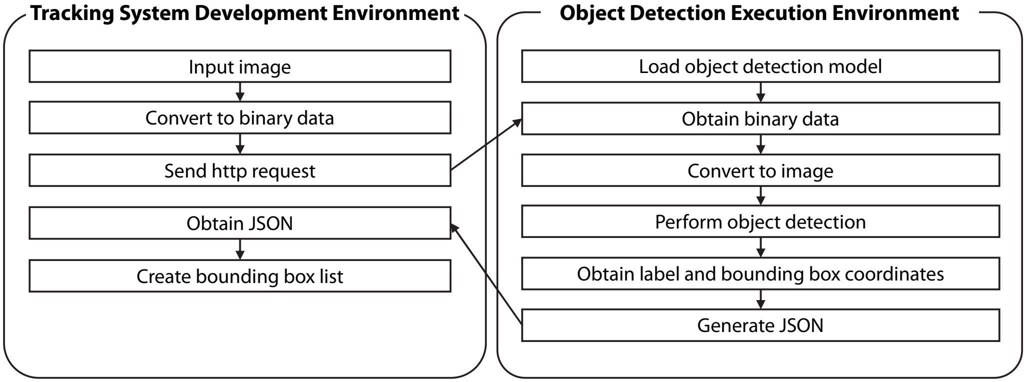


Figure 3
Flow diagram for performing object detection

The object detection model that we used was the pre-trained single shot multibox detector (SSD, Liu et al., 2016) with the Pascal VOC (2007 + 2012) dataset. SSDs are fast, can be detected in real-time, and are easy to implement since implementation methods for them are available on the internet for a variety of environments.

To reduce false positives, we decided not to output bounding box coordinates for any objects other than people and cars. We assumed that any other objects were unlikely to intrude into the MR execution screen.

Verification summary

To verify the performance of the proposed tracking system, we applied both the previous system and the proposed system (with integrated object detection) to a video of a moving object (a passing pedestrian) and compared the results. Details of the equipment that was used are shown in Table 2. Note that to reduce the processing load, the video was shot in full HD (1,920 px x 1,080 px) and then reduced to 1,024 px x 576 px in Unity. The reason for using the same pre-recorded video for the verification process instead of real-time video is because using real-time video makes exact comparisons difficult due to variations in multiple factors, such as pedestrian speed and daylight conditions. A map and photo of the area that was to be verified are shown in Figure 4. The video was taken at approximately 11:30 a.m. on January 30, 2022, and the weather was cloudy.

The pedestrian passed over the whole MR image in about 10 seconds. Since the image was 1,024 px wide, the average relative velocity was about 102 px/s. The webcam was fixed on a tripod, panned counterclockwise over a period of approximately 6 seconds, and then panned clockwise over a period of approximately 5 seconds. In this study, to make it easier to verify the movement of the virtual camera is consistent with that of the webcam and to work from a simple example, we moved the camera in this manner. Testing of the system's operation in moving viewpoints more freely, assuming a handheld device, will be done in the future.

Then, using the Harris corner detector (Harris and Stephens, 1988) on this verification video, no corners were extracted from the ground in the video. Therefore, it is hard to perform optical flow estimation for the ground in this verification video. Thus, 100 tracking reference points were placed only on the building wall, which is a vertical surface.

Note that to align the data from the previous system with that of this system, the frame rate for both systems was adjusted to 13 frames per second (fps), and tracking was started 10 frames after the start of the video.

Category	Name of Product	Specification		
Desktop PC	home-built	OS	Windows10 64bit	
		CPU	Intel Core i7-8700 CPU @ 3.20 GHz	
		RAM	32.0 GB	
		GPU	GeForce GTX 1080	
Web Camera	Logicool C922n PRO HD Stream Web Camera	Max. fps	Full HD 1080 p/30 fps	HD 720 p/60 fps
		Diagonal field of view	78 °	

Table 2
Details of the equipment used

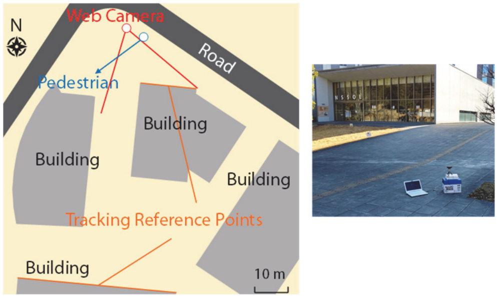


Figure 4
Map (left) and photo (right) of the area around the site to be verified

Number of remaining tracking reference points

We compared the number of remaining tracking reference points in the previous (Inoue et al, 2018) with those in the proposed system.

Since each tracking reference point is removed when the error in the optical flow estimation becomes large, the number of tracking reference points that did not disappear can be used as an indicator for evaluating system performance. Figure 5 shows the system in action immediately after tracking begins, during the pedestrian passage, and after the pedestrian passage. We confirmed that in the previous system, most of the tracking reference points that were in areas that pedestrians passed through were deleted as outliers. In the proposed system, however, many points remained.

Figure 6 shows a graph of the number of tracking reference points in each system from the start of the video until the pedestrian passes by. The number of tracking reference points that remained is given on the vertical axis, and the number of frames elapsed from the start of the video until the pedestrian passed by is given on the horizontal axis. Here, the number of defined tracking reference points other than those lost due to movement off the screen is shown in the graph as the maximum number of possible tracking reference points. Of the 100 tracking reference points defined, the maximum number of them that could be taken after the pedestrian passed was 62. Of these, our system was able to leave 55 (88.7%) after the pedestrian passed, compared to 10 (16.1%) in the previous system.

Tracking Accuracy

We compared the tracking performance of the previous system (Inoue et al., 2018) with that of the proposed system by displaying the 3D model in MR and observing how it looked after tracking started. The 3D model of the tree shown in Figure 7 (left) was placed as shown in Figure 7 (right) and tracking was started.

In the previous system, the position of the 3D model fluctuated significantly as pedestrians passed by.

DISCUSSION

This section discusses the verification results that were obtained.

There was an improvement in the percentage of tracking reference points that remained from 16.1% to 88.7%. Our system of initially excluding and then restoring tracking reference points makes it harder for errors in the optical flow estimation to occur for points that are near the moving object, thus resulting in significantly fewer being deleted as outliers. However, some tracking reference points still disappear. This could be due to some of the points being affected by moving objects that are

Figure 5
The system performance

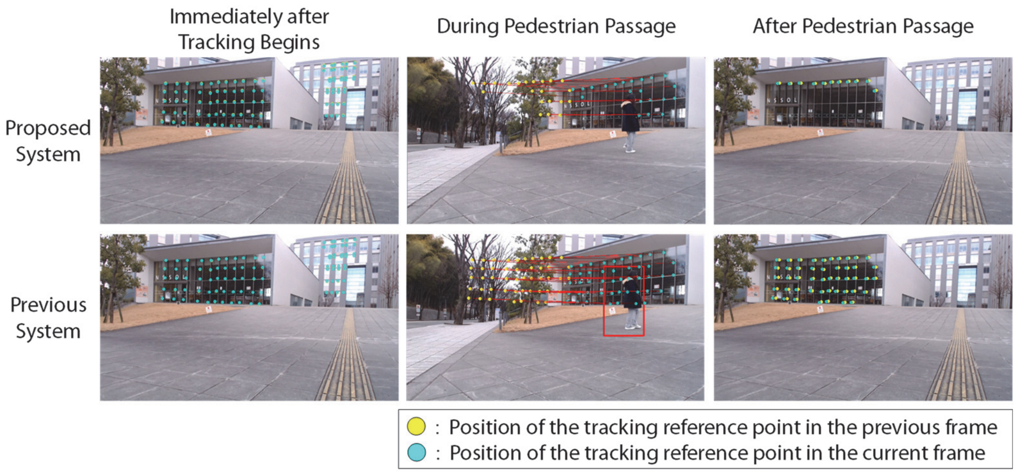
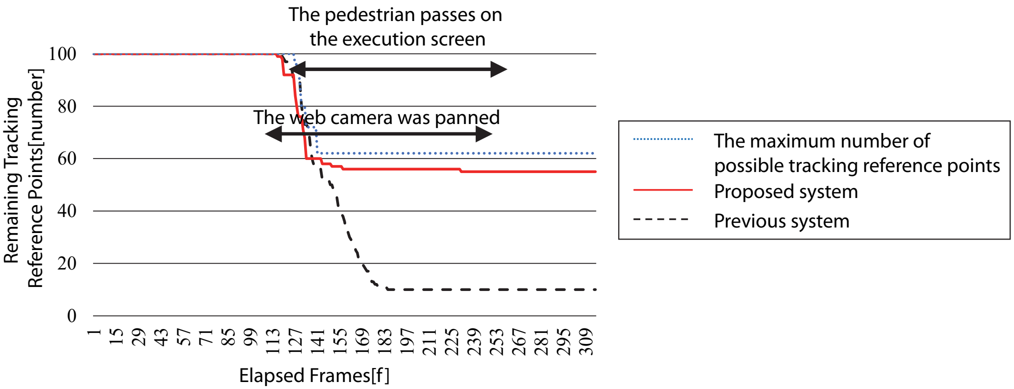


Figure 6
The graph representing the number of tracking reference points in each system



outside the boundary of the bounding box. Therefore, a parameter study is necessary for determining how to properly size the bounding box.

When a 3D model of a tree was placed into the frame, we confirmed that the fluctuation of the 3D model's position was much smaller than that of the previous system. However, the accuracy of the system needs to be improved since gradual deviations still occurred. The accuracy of the initial registration is a factor that could be influential in determining tracking accuracy since the estimation of the camera pose is made relative to the initial pose. This and other factors will be studied in the future.

Since this study only verified video of a single pedestrian passing by, it will be necessary in the future to test whether the system can verify videos with other moving objects, such as a car or multiple pedestrians passing by simultaneously. If there are multiple moving objects or if the area occupied by the moving objects is large, then the proposed system may not yet be able to handle this situation due to there being too few tracking reference points

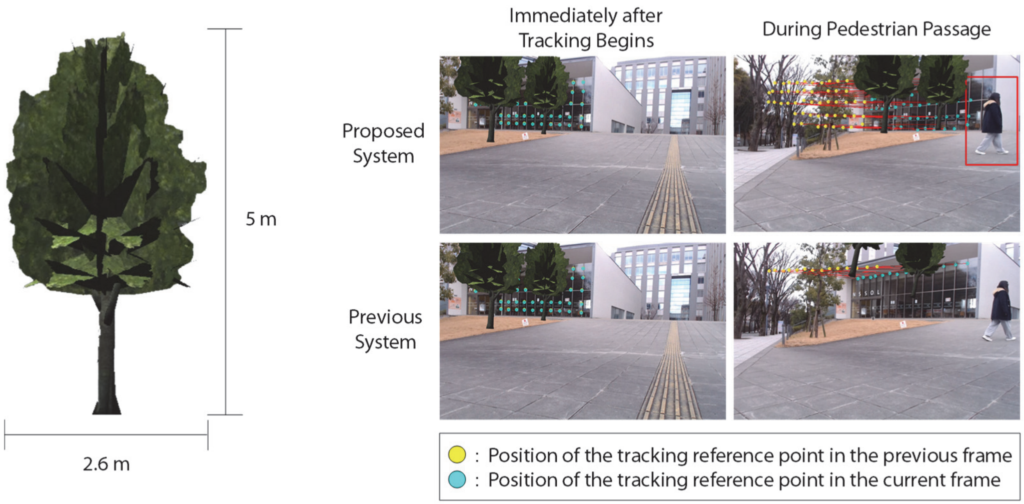


Figure 7
The 3D model of the tree (left) and the system performance (right)

available for the tracking process. In addition, shadows that are generated by moving objects and that cannot be detected by object detection are considered to be part of the moving object. This is an issue that cannot be handled by our proposed system and that will have to be addressed in the future.

We defined the tracking reference points manually. This allows the 3D coordinates to be defined more accurately for points on the surface far from the viewpoint than would be obtained by estimation. In addition, the optical flow estimation used in the proposed system assumes that the same point has a constant luminance value. So, surfaces such as water, whose luminance is prone to change, are more likely to cause errors. It is possible to eliminate tracking reference points from such surfaces in advance by defining them manually. For these reasons, it is advantageous for the user to be able to set points arbitrarily. However, this procedure is cumbersome and hinders its practical application, so we also plan to address this issue in future research.

In addition, with the introduction of object detection processing into the system, blurring and other processing can be applied based on the information of detected objects. It would be easy to run MR simulations in real-time and generate privacy-conscious simulation videos simultaneously.

CONCLUSION

MR has gained attention for potentially being an effective visual method for landscape studies. MR requires a proper registration of the real and virtual worlds to provide the user with an immersive experience. One of the main sources of positioning errors is those made in tracking the camera's real-time pose in an actual space.

The breakdown of tracking when a moving object passes over a tracking reference point in the MR execution screen is a challenge in natural-feature-based tracking that was emphasized by Inoue et al. (2018) To address it, this study proposed an approach that enhances tracking by integrating object detection into the system, the aim being to reduce the influence of moving objects. After it was implemented, the performance of the system was verified by using pre-recorded video to compare it to the previous system.

The conclusions of this study are as follows.

- Performance verification results show that our system was able to retain tracking reference points that were deleted by the previous system when a moving object passed by.
- Regarding the camera pose estimation, we confirmed that blurring was reduced compared to the previous system, but a gradual shift was still observed.

In our system, some errors occurred during the estimation of the camera pose when the system was running. Future work is needed to determine the cause of these errors and to resolve it. Additionally, this study only examined the case of a single pedestrian passing by. In future work, it will be necessary to examine how our system handles more complex situations, such as parking lots and roads, where there are many fast-moving objects, and times of day when shadows are cast by moving objects.

MR-based landscape studies can help engage non-professional citizens as well as professionals in design and decision-making because they provide a real-time, full-scale experience in the field. However, dynamic objects in outdoors can become noise and can prevent the simulation from working well and hinder smooth discussions. Implementing an MR system with the proposed tracking system is expected to reduce such risks.

In addition, since the proposed system estimates the camera pose using only the camera image as input, it does not depend on the device model. Of course, it is also possible to display 3D models other than the trees used in the verification and to change the models in real-time.

ACKNOWLEDGEMENTS

This research was partly supported by the Japan Society for the Promotion of Science, KAKENHI Grant Number JP19K12681. All images were created by the authors from 2021 to 2022.

REFERENCES

Al-Zoube, M.A. (2022) 'Efficient vision-based multi-target augmented reality in the browser', Multimedia Tools and Applications, Available at: https://doi.org/10.1007/s11042-022-12206-6 (Accessed: 14 March 2022).

Blanco-Pons, S., Carrión-Ruiz, B., Duong, M., Chartrand, J., Fai, S., and Lerma, J. L. (2019) 'Augmented reality markerless multi-image outdoor tracking system for the historical buildings on parliament hill', Sustainability, 11(16), p.4268, Available at: https://doi.org/10.3390/su11164268 (Accessed 19 March 2022).

Comport, A. I., Marchand, E., Pressigout, M., and Chaumette, F. (2006) 'Real-time markerless tracking for augmented reality: the virtual visual servoing framework', IEEE Transactions on Visualization and Computer Graphics, 12 (4), pp. 615–628, Available at: https://doi.org/10.1109/TVCG.2006.78 (Accessed 19 March 2022).

Fischler, M. A., and Bolles, R. C. (1981) 'Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography', Communications of the ACM, 24 (6), pp. 381–395, Available at: https://doi.org/10.1145/358669.358692 (Accessed 19 March 2022).

Harris, C., Stephens, M. (1988) 'A combined corner and edge detector', Proceedings of the Alvey Vision Conference (AVC), pp. 147-151.

Honjo, T., and Lim, E. (2001) 'Visualization of landscape by VRML system', Landscape and Urban Planning, 55 (3), pp.175-183. Available at: https://doi.org/10.1016/S0169-2046(01)00152-9 (Accessed: 7 March 2022).

Horn, B. K. P., and Schunck, B. G. (1981) 'Determining optical flow', Artificial Intelligence, 17 (1), pp. 185-203, Available at: https://doi.org/10.1016/0004-3702(81)90024-2 (Accessed 19 March 2022).

Inoue, K., Fukuda, T., Cao, R., and Yabuki, N. (2018) 'Tracking robustness and green view index estimation of augmented and diminished reality for environmental design: PhotoAR+DR 2017 project', *Proceedings of the 23rd International Conference on Computer-Aided Architectural Design Research in Asia (CAADRIA)*, 17-19 May 2018, Beijing, China, pp. 339-348.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C., and Berg, A.C. (2016) 'SSD: Single shot multibox detector', European Conference on Computer Vision (ECCV), pp. 21-37, Available at: https://doi.org/10.48550/arXiv.1512.02325 (Accessed 8 March 2022).

Lucas, B. D., and Kanade, T. (1981) 'An iterative image registration technique with an application to stereo vision', Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI), 24-28 August 1981, Vancouver, British Columbia, Canada, pp. 674-679.

Milgram, P., and Kishino, F., (1994) 'A taxonomy of mixed reality visual displays', IEICE Transactions on Information Systems, E77-D (12), pp. 1321-1329.

Neumann, U. and You, S., (1999) 'Natural feature tracking for augmented reality', IEEE Transactions on Multimedia, 1 (1), pp. 53-64, Available at: https://doi.org/10.1109/6046.748171 (Accessed 13 March 2022).

Pentenrieder, K., Meier, P., and Klinker, G. (2006). 'Analysis of tracking accuracy for single-camera square-marker-based tracking', Third Workshop on Virtual and Augmented Reality of the GI-Fachgruppe VR/AR, Koblenz, Germany.

Sato, Y., Fukuda, T., Yabuki, N., Michikawa, T., and Motamedi, A. (2016) 'A marker-less augmented reality system using feature-based image registration and structure from motion technologies for building and urban environment', *Proceedings of the 21st International Conference on Computer-Aided Architectural Design Research in Asia (CAADRIA)*, Melbourne, Australia, pp. 713-722.

Tomasi, C., and Kanade, T. (1992) 'Shape and motion from image streams under orthography: a factorization method', International Journal of Computer Vision, 9 (2), pp. 137-154, Available at: https://doi.org/10.1007/BF00129684 (Accessed 19 March 2022).

Vacchetti, L., Lepetit, V., and Fua, P. (2003) 'Fusing online and offline information for stable 3d tracking in real-time', Proceedings of 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 18-20 June 2003, Madison, WI, USA, pp. II-241, Available at: https://doi.org/10.1109/CVPR.2003.1211476 (Accessed 8 March 2022).