

3D Hand Movement Measurement Framework for Studying Human-Computer Interaction

Kuronen Toni, Eerola Tuomas, Lensu Lasse, Häkkinen Jukka, Kälviäinen Heikki

This is a Author's accepted manuscript (AAM) version of a publication

published by Springer, Cham

in Cyber-Physical Systems and Control. CPS&C 2019. Lecture Notes in Networks and Systems.

DOI: 10.1007/978-3-030-34983-7_50

Copyright of the original publication: © Springer Nature Switzerland AG 2020

Please cite the publication as follows:

Kuronen T., Eerola T., Lensu L., Häkkinen J., Kälviäinen H. (2020) 3D Hand Movement Measurement Framework for Studying Human-Computer Interaction. In: Arseniev D., Overmeyer L., Kälviäinen H., Katalinić B. (eds) Cyber-Physical Systems and Control. CPS&C 2019. Lecture Notes in Networks and Systems, vol 95. Springer, Cham

**This is a parallel published version of an original publication.
This version can differ from the original published article.**

3D Hand Movement Measurement Framework for Studying Human-Computer Interaction

Toni Kuronen¹, Tuomas Eerola¹, Lasse Lensu¹, Jukka Häkkinen², and
Heikki Kälviäinen¹

¹ Computer Vision and Pattern Recognition Laboratory (LUT CVPR),
Department of Computational and Process Engineering,
School of Engineering Science, LUT University,
P.O.Box 20, FI-53851 Lappeenranta, Finland
firstname.lastname@lut.fi

² Institute of Behavioural Sciences, University of Helsinki,
P.O.Box 9, FI-00014 Helsinki, Finland

Abstract. In order to develop better touch and gesture user interfaces, it is important to be able to measure how humans move their hands while interacting with technical devices. The recent advances in high-speed imaging technology and in image-based object tracking techniques have made it possible to accurately measure the hand movement from videos without the need for data gloves or other sensors that would limit the natural hand movements. In this paper, we propose a complete framework to measure hand movements in 3D in human-computer interaction situations. The framework includes the composition of the measurement setup, selecting the object tracking methods, post-processing of the motion trajectories, 3D trajectory reconstruction, and characterizing and visualizing the movement data. We demonstrate the framework in a context where 3D touch screen usability is studied with 3D stimuli.

Keywords: high-speed video, hand tracking, trajectory processing, 3D reconstruction, video synchronization, human-computer interaction

1 Introduction

In the human-computer interaction (HCI) research, it is necessary to accurately record hand and finger movements of test subjects in tasks related to user interfaces. Advances in gesture interfaces, touch screens, and augmented and virtual reality have brought new usability concerns that need to be studied in a natural environment and in an unobtrusive way [21]. Data gloves with electromechanical, infrared or magnetic sensors can measure the hand and finger location with high accuracy [5]. However, such devices affect the natural hand motion and cannot be considered feasible solutions when pursuing natural HCI. Consequently, image-based solutions which provide an unobtrusive way to study and to track human movement and tenable natural interaction with the technology have become a pronounced subject of research interest.

Commercially available off-the-shelf measurement solutions such as Leap Motion ³ and Microsoft Kinect ⁴ do not allow frame rates high enough to capture all the nuances of rapid hand movements. Moreover, Leap Motion limits the hand movement to a relatively small area. The field of view of the sensor is an inverted pyramid, with an angle of 150 degrees in the left-right direction, and an angle of 120 degrees in the front-back direction, and the measurement distance ranges from 25 mm to 60 cm above the device [6]. Kinect is sufficient for detecting arm and full body gestures, but it is imprecise for accurate finger movement measurements [13]. Furthermore, such commercial solutions lack the inspecting capability of a camera-based system and do not allow further analysis of hand pose beyond the limitations of the sensors.

An alternative approach for accurate recording of fast phenomena including rapid and subtle hand movements is high-speed imaging. High-speed videos provide the basis for building a system that is more versatile than the existing black-box solutions. From the implementation viewpoint, high-speed imaging requires more light than the conventional imaging to allow short exposure times, which imposes additional demands on the measurement setup. Moreover, the bright illumination can disturb the user performing the HCI experiment since it reduces the perceived contrast and, making it difficult to see the stimulus. Thus, careful planning of the measurement setup is important to ensure that the conditions for the interaction are as natural as possible.

To record hand movements in 3D with a camera-based measurement, at least two cameras with different viewing angles are required. However, a setup consisting multiple high-speed cameras is both expensive and difficult to build. This motivates to use of a normal-speed camera in addition to the high-speed camera to provide the depth information for reconstructing the 3D trajectories.

Recent progress in object tracking techniques has made it possible to automatically determine motion trajectories from videos. Gray-scale high-speed imaging is commonly used to keep illumination requirements at a reasonable level, and consequently, the use of hand tracking methods relying specifically on color information becomes impractical. These matters motivate the utilization of general object trackers. For example, in [11], several object tracking methods were compared using high-speed videos, and the best methods were found to be suitable for the problem of measuring hand movements in the context of HCI.

The main problem with using existing object tracking methods for accurate measurement of hand and finger movements is that they were developed for applications where high spatial accuracy is not crucial, as the research focus was on developing more computationally efficient and robust methods. For these methods, losing the target is considered a much more severe problem than a small spatial shift of the tracking window. This is not the case in hand trajectory measurement based on high-speed videos where small hand movements between the frames and a controlled environment help to achieve robustness. Thus, high spatial accuracy is the main concern. Even small errors in the spatial locations

³ Leap motion: <https://www.leapmotion.com/product>

⁴ Microsoft Kinect: <http://www.xbox.com/en-US/kinect>

can lead to large fluctuations in the speed and acceleration determined from the location data. Therefore, existing tracking algorithms alone are insufficient for the accurate measurements of hand movements and further processing of the hand position data is needed.

Raw trajectory data contain small spatial location fluctuations that can make calculation of accurate velocities and accelerations impossible. Smoothing raw trajectory data with an appropriate filtering method provides a solution for small irregularities in the trajectory data without compromising the tracking results [16]. After smoothing, it is possible to compute the velocities and accelerations, i.e., the first and second derivatives of the position, with greater accuracy.

To process large amounts of video data, it is advantageous to automatically detect tracking failures, i.e., cases where the tracking is lost or the tracking window drifts from the target. When a failure is detected, either the tracking can be repeated with a different tracking method, or the incorrect trajectory can be excluded from further analysis. A common approach to detect failures is to use backtracking and compare the tracked target to an earlier sample of the object (e.g., [12,23,7]). Such methods perform well when the tracking is lost, causing large displacements between the tracking result and the actual object location. However, when a tracker slowly drifts away from the target, failure detection with backtracking methods becomes more challenging.

To address the above issues and requirements, we present a multi-camera framework for measuring hand movements in HCI studies, focusing on touch and gesture user interfaces. The framework is developed for a measurement setup consisting of a high-speed camera and a normal-speed camera with different viewing angles. The high-speed camera makes it possible to detect fast and subtle changes in the trajectories while the normal-speed camera provides the necessary additional information to construct the 3D trajectories. The framework includes the construction of the measurements setup, selection of object tracking methods, detection of tracking failures, post-processing of the trajectories, and characterization and visualization of the movement data. The framework is generic in nature, and in this work, it is demonstrated with an application in which 3D touch screen usability is studied with 3D stimuli.

2 Overview of the framework

An overview of the proposed hand movement measurement framework for the HCI studies is shown in Fig. 1. The dashed line in the figure represents the use of the camera calibration results in the computation of the real-world features. The first step is to design and to build the measurement setup which comprises cameras, illumination, a display, and other interacting devices, and the required hardware for triggering and storing recordings. The main considerations when designing a HCI measurement setup are that it should not interfere with the usability of the user interface and that it should offer a natural setting for test subjects performing the selected HCI task. This aspect is particularly important in design of the illumination because high-speed imaging setup requires a lot of light and

bright illumination can disturb test subjects. Moreover, the illumination should not result in flickering on the recorded videos so the available flicker-free light sources should be used, including LED light panels with reliable and constant power sources and Hydrargyrum medium-arc iodide (HMI) lamps where flicker can be avoided by using electronic ballast that operate at high frequencies.

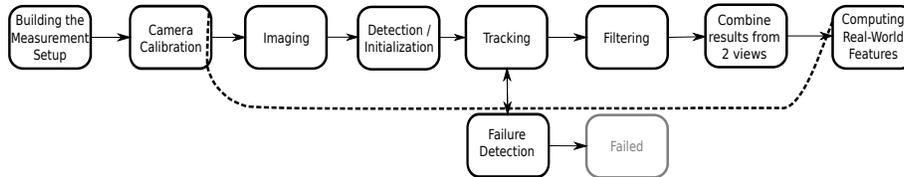


Fig. 1: Overview of the measurement framework.

The second step in the framework is to geometrically calibrate the cameras by determining the intrinsic and extrinsic camera parameters to obtain the mapping from the image point locations to the real-world coordinates. The pinhole camera model parameters can be determined by using known and imaged interest point coordinates of an imaging target designed for calibration. These parameters can then be used to transform the image point locations to real-world coordinates via a perspective projection. The pinhole camera model can be further enhanced by taking into account the lens and sensor distortions.

After imaging the object of interest, the object needs to be detected before its movement can be tracked. In a typical controlled HCI study, the hand or finger movement starts from a static trigger box or another predefined location. However, if the initial position is unknown, a detection component is needed before tracking. The detection can be performed using state-of-the-art detection methods, such as Faster R-CNN [19] or YOLO9000 [18]. If the background is static, a simple method such as frame differencing or background subtraction can be used for the detection. The initialization of the object position has an important role in the tracking process since a typical tracking method utilizes the initial position to generate the object model used for tracking.

Tracking is applied in order to follow the position of the detected or otherwise initialized target object while it is moving. In general, the idea is to repeatedly estimate the transformation of an object from time step t to $t + 1$, i.e., from one image frame to the next one. In most cases, the transformation is simply the translation of an object. However, there are situations where a more advanced motion model is required that takes into account, e.g., rotation, skew, and scale changes. An extensive comparison of object tracking algorithms for measuring hand movements in the HCI study is presented in [16] with the high-speed videos and in [17] with the normal-speed videos.

Extracting higher level features from the tracking results can be challenging [16]. Although a list of center locations of an object over time produced by tracking is usable for tasks such as checking the position of an object at a certain

time, sub-pixel accuracy is preferred when derived quantities such as velocity or acceleration are required. Typical object trackers operate at the pixel level and the resulting trajectory often contains noise. The desired level of spatial accuracy and the noise cause challenges for the determination of derived quantities such as velocity and acceleration. High-speed videos can be challenging where movements between the frames are very small (often less than a pixel). Consequently, filtering of the trajectories is required. Finally to reconstruct hand trajectories in 3D, the tracking results from two views obtained from the normal and high-speed videos are combined and various features of the trajectories are computed.

3 Stereoscopic 3D touch display experiment

The framework is demonstrated with a HCI experiment using a stereoscopic 3D touch screen setup. In the experiment, test subjects were advised to perform intentional single finger pointing actions from a trigger-box toward a target that was on a different parallax than others on the touch screen. Hand movements were recorded with a high-speed camera and normal-speed camera. The trigger-box and the touch screen were placed on a table as shown in Fig. 2. The flow of the experiments was controlled by a middleware program specially coded for the experiments. A detailed description of the setup can be found in [15].

Similar to earlier pointing action research, e.g., [3], the experiment focused on studying intentional pointing actions. The stimuli were generated by a 3D display with the touch screen to evaluate the effect of different parallaxes, i.e., perceived depth. This arrangement enables study of (potential) conflict between visually perceived and touch-based sensations of depth.

The 3D stereoscopic touch screen experiment was executed as follows. 20 test subjects conducted 4 different sessions of pointing actions with different parameters. These were divided into nine blocks. The test image contained a fixation cross in the middle of the screen and 10 rectangular blocks around it in a circle formation. The aim of the experiment was to locate and touch the target that appears on a different parallax to the others in the test image.

The high-speed videos were recorded at 500 fps and 800×600 resolution. The normal-speed videos were recorded using interlaced encoding with 50 field rate and 1440×1080 (4:3) resolution. For deinterlacing the normal-speed videos the yet another deinterlacing filter (yadif) [1] was utilized with field-to-frame conversion producing double frame rate (50 fps) videos. In total, 2597 pointing actions were recorded with the both cameras.

3.1 Camera calibration

To calibrate the cameras, a standard calibration board with 26.5 mm checkerboard patterns was used. A set of calibration images was captured and used to compute the intrinsic camera parameters. The Camera Calibration Toolbox for Matlab [2] was used to perform the calibration, as it is a robust and well-established calibration tool, based on [24] and [9].

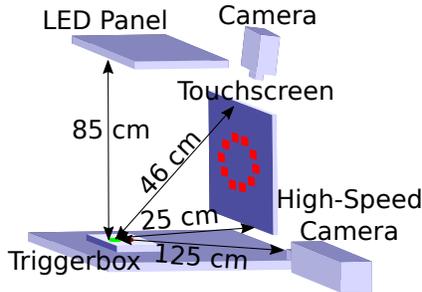


Fig. 2: 3D touch display experiment.

3.2 Hand tracking

Since trackers specifically designed for hand tracking rely on color information whereas gray-scale imaging is used in this work the selected state-of-the-art general object trackers were utilized. Based on the comprehensive evaluation on the same video dataset reported in [16], Kernelized Correlation Filters (KCF) based tracker [10] was selected for tracking in high-speed videos. The tracking window was initialized by a manually placed initial bounding box on the trigger box button image.

The normal-speed videos were processed with motion detection near the monitor area. The motion detection was performed using background subtraction (frame differencing). The detected motions were used to obtain the location of the finger tip which was further used to initialize the tracking window for the normal-speed videos. Comprehensive evaluation of state-of-the-art object trackers for finger tracking from normal-speed videos with the presented experimental setup has been provided in [17]. Based on the results, the KCF tracker extended by a scale estimation and color-names features (KCF2) [22] was selected as a normal-speed video tracker for the final measurement framework.

3.3 Trajectory post-processing

Failure detection In situations where a highly robust tracking system is required or massive datasets are processed, there is a need for a failure detection system to identify failed trajectories as it was identified in [14]. One of the methods to detect tracking failures is to use backtracking to estimate the trajectory from the current point to the beginning of the tracking, or another earlier point of time, and to check if the backtracked trajectory matches the original “forward-tracked” trajectory [7]. Other methods include gathering samples of the earlier appearances of the object and comparing them to the currently tracked window using similarity measures [12,23].

Typically, failures are easier to detect when the drift is large. However, when the tracker slowly drifts off the target, it is more difficult to detect the failure, and the above-mentioned methods become unreliable, especially if high spatial

accuracy is desired. In HCI studies, the end point of the trajectory is often known. Moreover, in some studies also the start point of the trajectory is known, for example, a trigger-box button. In the touch screen experiment, for example, the point on the screen that the test subject touches is known, and this information can be used to implement a reliable method to detect failures in tracking. When failures are detected, either the tracking can be repeated with another tracking method, or the incorrect trajectory can be excluded from the further analysis.

If the end position of the trajectory is unknown a reliable backtracking or drifting detection method should be applied. In the HCI studies, methods such as good features to track [20] and metrics for the performance evaluation of video object segmentation and tracking without the ground-truth [4], based on earlier templates of an object, work relatively well since the target object that is usually a hand or a finger. These objects contain well identifiable features that can be used to detect if the tracker loses the target. Moreover, the object detection methods used for the tracker initialization can be applied for the last frames to test if the end point of the tracked trajectory contains the correct object.

It should be noted that many of the tracking failure cases could be avoided by giving the test subjects precise instructions and by ensuring sufficient practice before the actual data collection so that the test subjects are comfortable with the task. Erroneous behavior by test subjects can include, for example, test subjects withdrawing their hand from the touching position before the recording ends, incorrect positioning of the hand in the beginning of an individual test, and obstruction of the pointing finger with other fingers.

In the 3D touch display experiment, the tracking failure detection method needed to be able to reliably process a large amount of trajectories. The implemented failure detection system was based on the fact that the trajectory had to end within a specific area of the projected touch screen point. If the correct end point was not reached with the default gray-level features used by the KCF tracker, the tracking was repeated with more computationally demanding HOG features. If the tracking failed again, the trajectory was considered as incorrect and was excluded from the further analysis.

Trajectory filtering Based on the results of the high-speed trajectory filtering in [16] the LOESS filter was selected also for the normal-speed trajectories. The filtering window size was selected to be the same 80 milliseconds as in high-speed case. This translates into 4 samples window size with 50 fps. Comparison between different window sizes can be found in [17].

Video synchronization In order to automatically align the normal-speed videos with the high-speed videos, the ratio of the framerates and the delay (difference between the camera-produced time information) were determined in [15]. The synchronization process uses timestamps from the high-speed videos and known starting time of the normal-speed videos to coarsely align the videos and to identify blocks of corresponding actions from both videos. The known location from the both views are used to set up an event which can be then used

to align the video sequences accurately. A more detailed explanation of the video synchronization method is given in [15].

Reconstruction of 3D trajectories To obtain a 3D trajectory, the 2D trajectories estimated using the calibrated cameras with a different viewpoint need to be combined. The task of computing a 3D trajectory from multiple 2D trajectories is essentially equivalent to the process of 3D scene reconstruction. For this purpose, we utilized the 3D reconstruction method presented in [8]. Detailed explanation of the 3D reconstruction and results are available in [17] and [15].

4 Data analysis

4.1 3D trajectory reconstruction

The success rate of the finger tracking was measured as the proportion of trajectories which reached the predefined end points. For the high-speed videos, the end points were the touch target areas re-projected onto the image plane, and for the normal-speed videos, the defined end point was the trigger-box button. 77% of point actions were tracked correctly from the high-speed videos and 69% from the normal-speed videos. In total, 1237 (62%) of the pointing actions were correctly tracked from the both videos and were aligned correctly. Since there was no ground truth for the 3D trajectories, the 3D reconstruction accuracy was assessed by using the re-projection error measure [8]. The mean re-projection error over all the trajectory points used from 1172 videos in the 3D reconstruction experiment was 31.2 pixels. This corresponds to approximately 10 millimeters in the real world.

4.2 Trajectory features

When hand movements are considered in HCI studies, the most important measurements are the velocities and accelerations of the hand [3]. Velocity and acceleration can be computed as the first and second derivatives of the position with respect to time using the tracked hand trajectories and Euclidean distances. Trajectory filtering makes it possible to compute the velocity using differences in consecutive trajectory points, and the acceleration can be computed using consecutive filtered velocity points.

For visualization purposes, velocity and acceleration curves can be plotted with respect to either time or position. For example, in the 3D touch screen experiment, the distance of the fingertip to the monitor surface is a useful measurement. It should be noted that this measurement fails to capture movement that occurs in a plane parallel to the screen which is not crucial when simple intentional pointing actions are studied.

In a typical experiment, individual movement trajectories vary considerably. To detect small recurring events and phenomena, as well as to identify inter-subject behavioral differences in slightly different tasks, it is important to be

able to analyze and visualize a large number of trajectories. One option is to determine the average position, velocity, and acceleration curves. In order to do this, the different trajectories need to be normalized in such a way that the mean values can be computed for a certain moment of time or a certain position.

In the 3D touch screen experiment, the trajectories were normalized so that all of them were at a distance of 250 mm from the touch screen measured from the initial finger position at the trigger-box. The visualization of the grouped acceleration and speed plots was used to detect submovement intervals of the trajectories similarly to [3]. The primary submovement starts with the initial acceleration, and ends when there is a sign change from negative to positive. This is the starting point of the secondary submovement of intentional pointing actions where minor adjustments to the trajectory are made and the movement is fixed to the final target position. Similar submovement events can be seen in the visualization of the 3D touch screen experiment in Fig. 3. The results indicate that differences in acceleration and speed of the pointing actions when using different parallaxes are small, but nevertheless observable. Fig. 4 shows the velocity and acceleration curves for the last 25 mm before the touch display. There are small differences in velocities and accelerations on movements towards different disparities so parallax information seems to affect the hand trajectories slightly.

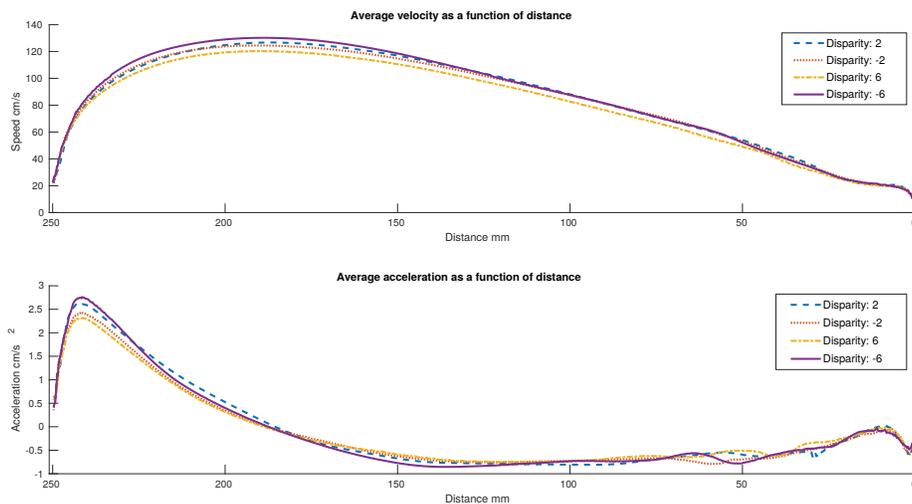


Fig. 3: Averaged results of all test subjects performing the 3D touch screen experiment with disparities -6, -2, 2, and 6.

In [15], eleven features were computed from the obtained trajectories: the mean velocity, median velocity, maximum velocity, maximum velocity during the 2nd submovement, maximum acceleration during the 2nd submovement, mean velocity during the 2nd submovement, and mean acceleration during the 2nd

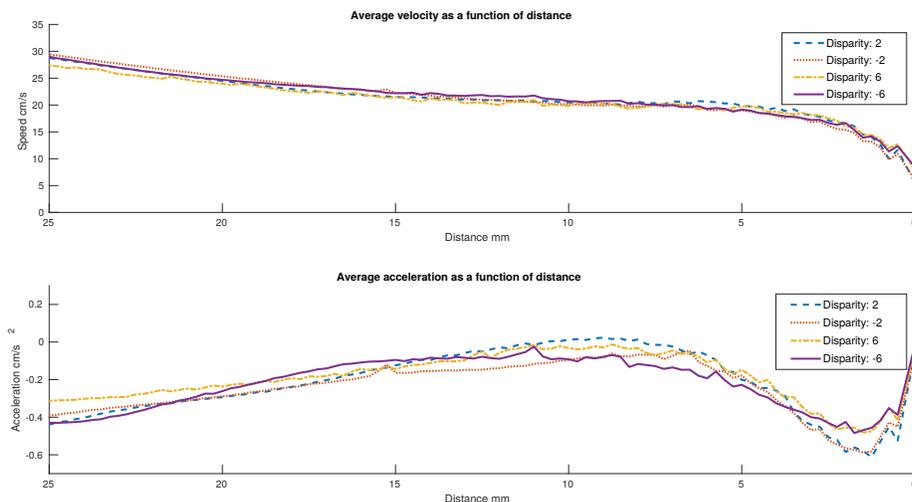


Fig. 4: Last 25 mm of the averaged velocity and acceleration curves with disparities -6, -2, 2, and 6.

submovement. Moreover, a two sample T-test, with 5% significance level, was used to analyze the trajectory features. As expected, it was concluded in [15] that the smaller disparity changes 2 and -2 had only minor impact to the hand movements according to the computed features whereas the disparity values 6 and -6 had more significant impact to the movements. Moreover, the large positive disparity 6 (the target object in front of the screen) seemed to have a more prominent effect on the pointing actions than the others. Furthermore, the velocity features seem to be better than the acceleration features to distinguish the pointing actions towards different disparity values.

5 Conclusion

A framework for measuring hand movements, in particular pointing actions, in human-computer interaction situations using a multiple camera system containing a high-speed and normal-speed cameras was introduced. Suitable object trackers to perform the finger tracking in both high and normal-speed videos were proposed based on earlier comprehensive studies. Selected KFC tracker for high-speed videos and KFC2 for normal-speed videos perform well in the given hand tracking task, achieving low error rates and operating at high processing speeds. In order to process large amounts of videos, we proposed a tracking failure detection method that excludes incorrect trajectories from further analysis, including cases where the test subject failed to follow the given instructions. By using trajectory filtering, the tracked trajectories could be smoothed to obtain reliable acceleration and velocity curves for visualization purposes. Finally, a method to construct 3D trajectory from the two 2D trajectories was proposed. The framework provides

real-time processing speeds. Including video loading, tracking, filtering and visualization, the processing speed of the framework was on average around 100 fps for high-speed videos. The framework was demonstrated in a context where 3D touch screen usability was studied with 3D stimuli. Some feature correlation with different parallaxes were already detected, but deeper analysis of the effects of different parallaxes on the trajectories is planned for the future research. The work provides valuable information about the suitability of general object tracking methods for high-speed hand tracking while producing appropriate velocity and acceleration features computed from filtered tracking data.

Acknowledgements

The authors would like thank Dr. Jari Takatalo for his efforts in implementing the experiments and producing the data for the research.

References

1. FFmpeg. <https://ffmpeg.org/> (2018), [Online; accessed 01-May-2018]
2. Camera calibration toolbox for matlab. http://www.vision.caltech.edu/bouguetj/calib_doc/ (2016), [Online; accessed 03-Feb-2016]
3. Elliott, D., Hansen, S., Grierson, L.E.M., Lyons, J., Bennett, S.J., Hayes, S.J.: Goal-directed aiming: two components but multiple processes. *Psychological Bulletin* 136(6), 1023–44 (2010)
4. Erdem, C.E., Tekalp, A.M., Sankur, B.: Metrics for performance evaluation of video object segmentation and tracking without ground-truth. In: *Proceedings 2001 International Conference on Image Processing*. pp. 69–72 (2001)
5. Erol, A., Bebis, G., Nicolescu, M., Boyle, R.D., Twombly, X.: Vision-based hand pose estimation: A review. *Computer Vision and Image Understanding* 108(1-2), 52 – 73 (2007)
6. Guna, J., Jakus, G., Pogacnik, M., Tomazic, S., Sodnik, J.: An analysis of the precision and reliability of the leap motion sensor and its suitability for static and dynamic tracking. *Sensors* 14(2), 3702–3720 (2014)
7. Hariyono, J., Hoang, V.D., Jo, K.H.: Tracking failure detection using time reverse distance error for human tracking. In: *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*. pp. 611–620 (2015)
8. Hartley, R.I., Zisserman, A.: *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edn. (2004)
9. Heikkilä, J., Silven, O.: A four-step camera calibration procedure with implicit image correction. In: *Conference on Computer Vision and Pattern Recognition*. pp. 1106–. Washington, DC, USA (1997)
10. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37(3), 583–596 (2015)
11. Hiltunen, V., Eerola, T., Lensu, L., Kälviäinen, H.: Comparison of general object trackers for hand tracking in high-speed videos. In: *International Conference on Pattern Recognition (ICPR)*. pp. 2215–2220 (2014)

12. Kalal, Z., Mikolajczyk, K., Matas, J.: Forward-backward error: Automatic detection of tracking failures. In: 20th International Conference on Pattern Recognition. pp. 2756–2759 (2010)
13. Khoshelham, K., Elberink, S.O.: Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors* 12(2), 1437–1454 (2012)
14. Kuronen, T.: Moving Object Analysis and Trajectory Processing with Applications in Human-Computer Interaction and Chemical Processes. Ph.D. thesis, Lappeenranta University of Technology (2018)
15. Kuronen, T., Eerola, T., Lensu, L., Kälviäinen, H.: Two-camera synchronization and trajectory reconstruction for a touch screen usability experiment. In: *Advanced Concepts for Intelligent Vision Systems*. pp. 125–136 (2018)
16. Kuronen, T., Eerola, T., Lensu, L., Takatalo, J., Häkkinen, J., Kälviäinen, H.: Image Analysis: 19th Scandinavian Conference, SCIA 2015, Copenhagen, Denmark, June 15-17, 2015. Proceedings, chap. High-Speed Hand Tracking for Studying Human-Computer Interaction, pp. 130–141. Springer International Publishing, Cham (2015)
17. Lyubanenko, V., Kuronen, T., Eerola, T., Lensu, L., Kälviäinen, H., Häkkinen, J.: Multi-camera finger tracking and 3D trajectory reconstruction for HCI studies. In: *Advanced Concepts for Intelligent Vision Systems*. pp. 63–74 (2017)
18. Redmon, J., Farhadi, A.: Yolog000: Better, faster, stronger. arXiv preprint arXiv:1612.08242 (2016)
19. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence* 39(6) (2017)
20. Shi, J., Tomasi, C.: Good features to track. In: *Conference on Computer Vision and Pattern Recognition*. pp. 593–600 (1994)
21. Valkov, D., Giesler, A., Hinrichs, K.: Evaluation of depth perception for touch interaction with stereoscopic rendered objects. In: *ACM International Conference on Interactive Tabletops and Surfaces*. pp. 21–30 (2012)
22. Vojir, T.: Tracking with Kernelized Correlation Filters. <https://github.com/vojirt/kcf/> (2017), [Online; accessed 01-May-2018]
23. Wu, H., Chellappa, R., Sankaranarayanan, A.C., Zhou, S.K.: Robust visual tracking using the time-reversibility constraint. In: *IEEE 11th International Conference on Computer Vision (ICCV)*. pp. 1–8 (2007)
24. Zhang, Z.: Flexible camera calibration by viewing a plane from unknown orientations. In: *International Conference on Computer Vision*. pp. 666–673 (1999)