


A low-cost non-intrusive spatial hand tracking pipeline for product-process interaction

James Gopsill , Aman Kukreja, Christopher Michael Jason Cox and Chris Snider

University of Bristol, United Kingdom

 james.gopsill@bristol.ac.uk

Abstract

Hands are the sensors and actuators for many design tasks. While several tools exist to capture human interaction and pose, many are expensive and require intrusive measurement devices to be placed on participants and often takes them out of the natural working environment. This paper reports a novel workflow that combines computer vision, several Machine Learning algorithms, and geometric transformations to provide a low-cost non-intrusive means of spatially tracking hands. A $\pm 3\text{mm}$ position accuracy was attained across a series of 3-dimensional follow the path studies.

Keywords: user-centred design, virtual reality (VR), interaction design, machine learning, prototyping

1. Introduction

Hands enable individuals to connect with, actuate and interrogate products, process, and tools across Design and Manufacture (Figure 1) (Openshaw & Taylor, 2006). Examples in design include clay modelling, prototyping, and sketching enabling designers to realise their ideas in physical form. Examples in manufacturing include cottage and industrial craft processes, such as basket weaving, pottery, woodworking, panel beating, drawing coach lines and carbon fibre layup. Hands are also a primary means of interacting with the products and processes that designers have created for society. Examples include cooking implements, sports equipment, clothing, and computer games controllers. There is a near endless list of product-process interactions that require an individuals to actuate and interrogate by their hands.



Figure 1. Hand interaction across design, manufacturing and products

The design community has undertaken considerable research to understand how data and information can be elicited from an individual's experience in using a product and/or performing a process with their

hands (Chien et al., 2016; Helminen et al., 2010; Karras et al., 2017; Schifferstein & Cleiren, 2005). Methods such as think-aloud and video capture coupled with, often manual, coding schemes have been developed and successfully used. However, there remains a disconnect and an interpretative step between coding what an individual describes as their interaction and the physical reality of the interaction. Papers reporting user interaction studies often cite the challenge for individuals to describe their interaction effectively through language while also processing and reacting to the incoming sensory data of the interaction episode (Lederman & Klatzky, 2009).

The importance of capturing interaction data and information has never been greater. In design, the emergence of Virtual, Augmented and Mixed Reality is providing an exciting and novel environment in which designers can interact, modify, and develop their designs. In manufacture, many cottage craft processes are in-danger of becoming extinct with fewer and fewer individuals wishing to take on crafts and therefore, no one to pass the knowledge onto. Industry 4.0 and the digitalisation of manufacturing processes require technician interaction data so that the processes can be replicated through tools, such as soft robotics. In products, mass-customisation is pushing for individualised and tailored experiences requiring designers to have a deeper and better understanding of how users interact with their product. Human factors research has shown that supplementing traditionally qualitative methodologies with high-rate quantitative data inputs both reduces design cycle time and allows for improved product design (Johnston et al., 2022). It is not only important to capture track hands spatially and to a high degree of accuracy but to also classify what the hands and/or activity the individual is doing when interacting with the product (Joong Hee Lee & Yun, 2023; Shi et al., 2021). Higher-level classifications, such as therbligs, can help synthesise the system of interactions that an individual has performed. Statistical methods applied to the classifications can generate profiles that represent population percentiles.

Methods to track hands and, more generally, the human body have existed for some time and are commonly referred to as motion capture. They have been widely used in the film and games industry, and necessitate special costumes and gloves with markers that may or may not protrude from the outfit. The movements are also captured out of context using green screens in order to maximise the capture fidelity. The systems are expensive and, while suitable for individuals imagining and acting out a scenario, take an individual outside their usual context when interacting with a product and/or process. The removal of the contextual surroundings can influence the interaction being studied.

Advances in Digital Technology – Computer Vision, Virtual Reality (VR) and Machine Learning (ML) – have increased the fidelity at which an individual's pose can be tracked through low-cost webcam and VR technology (Buckingham, 2021; Caeiro-Rodríguez et al., 2021). This paper contributes a digital processing pipeline that utilises these technologies to spatially track product-process hand interactions without markers and/or specialist environmental setups. A preliminary validation of the pipeline was performed through point and path motion studies.

The paper continues with the related work in hand tracking methods and product-process hand interaction studies (Section 2). Section 3 provides details of the digital processing pipeline that has been developed, which combines Computer Vision, ML and VR technologies. This is followed by Section 4 that details the preliminary validation experiment). Sections 5 and 6 present the results and a discussion with respect to the pipelines utility in studying product-process hand interaction. The paper then concludes with the key findings from the work (Section 7)

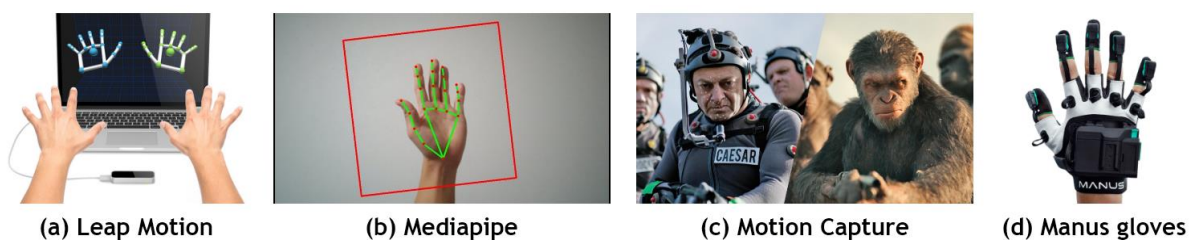


Figure 2. Example of available methods to capture interaction

2. Related work

This section provides an overview of the available hand position capture methods. The methods can be grouped into optical and worn (Figure 2).

Optical solutions use computer vision and/or Machine Learning to analyse one or more video streams (Lugaresi et al., 2019). The video capture equipment used can be of the visible (purely optical capture) or infra-red light (often used in depth-sensing technology or Motion Capture technology) variety. Mediapipe is an example of visible optical tracking and Ultraleap LeapMotion is an example of infra-red optical tracking (Zhang et al., 2020). Cox et al., 2023 analysed the accuracy of optical solutions for hand tracking and showed visible solutions performed better than for 2D tracking.

In some instances, specialist uniforms featuring reflective markers are worn. Individual are then situated in specially configured green rooms to capture their motion and interaction. These systems are known as motion capture and are used in the film and games industry (Merriault et al., 2017). The system detects and keeps track of the markers in the scene providing a highly accurate solution. They are highly accurate but require a user to operate and perform actions out of the context of their normal environment. They also cost many thousands of pounds to purchase, deploy and operate.

Worn solutions also exist. They track the hands by analysing sensor data coming from gloves worn by an individual. A combination of Inertial Measurement Units (IMUs) and/or lighthouses provide the data to determine the global position of a hand in 3D space. Examples include the HTC Vive trackers and Senso Gloves.

To determine finger positions, gloves typically employ resistive sensors along the fingers. Their resistance changes during deformation which is then mapped to how the finger must be positioned. The MANUS Prime X gloves are one such example of this approach. Magnetic tracking is another approach where fluctuations in the magnetic field can be used to determine the finger position. This technique is used by the HaptX and Manus Quantum MetaGlove gloves. A fundamental challenge of glove-based approaches is the need for the user to wear a pair of gloves which may affect the way an end-user interacts with the product. Cox et al. (2023) analysed the effectiveness of three hand tracking methods – LeapMotion, Mediapipe and Manus gloves. The results showed the ML-based Mediapipe method to be the most accurate and consistent system (Table 1). It also had the benefit of not requiring the user to wear any additional garments or have any markers placed upon them. The only challenge is that Mediapipe operated on a single camera results in issues of occlusion and the inability to detect the depth of the hand.

Table 1. Path tracking accuracy of existing approaches (from Cox et al., 2023)

	Static, intra-repeat position variance (mm ²)		Static, inter-repeat position variance (mm ²)		Dynamic, inter-repeat ang. of best-fit variance (deg ²)	
	Maximum	Average	Maximum	Average	Maximum	Average
<i>Leap.M-Open hand</i>	6.60	1.66	13.97	7.95	7.08	1.99
<i>Leap.M-Point finger*</i>	N / A	N / A	N / A	N / A	N / A	N / A
<i>M.Pipe-Open hand</i>	0.00	0.00	3.61	2.85	3.40	0.53
<i>M.Pipe-Point finger</i>	0.20	0.02	5.00	3.42	0.16	0.07
<i>Manus-Open hand</i>	7.74	2.68	24.59	16.7	16.16	4.38
<i>Manus-Point finger</i>	9.11	2.70	18.99	11.3	23.53	13.81

* Tracking failed for over 90% of repeats so no reliable data could be produced

3. A low-cost non-Intrusive hand tracking pipeline

The developed pipeline combines computer vision, ML, and Virtual Reality to provide spatial tracking of hands. The pipeline is illustrated in Figure 3. It was written in Python using the OpenCV, PyOpenXR, Numpy, Matplotlib, Mediapipe, and SKLearn packages (Harris et al., 2020; Hunter, 2007; Itseez, 2015; Pedregosa et al., 2011). The pipeline requires Python 3.10+, SteamVR, a series of web cameras and VR Head-Mounted Display (HMD). Figure 3 also provides an example of the required hardware.

The pipeline involves a three-step process to achieve spatial hand tracking:

1. Calibration
2. Training
3. Track

The Calibration step creates a dataset that relates the (x, y) positions of detected objects in view of the n web cameras to the (x, y, z) position of the object in VR. The object of interest was the HMD. A Charuco marker attached to the HMD enabled the web cameras to detect it using computer vision.

The web cameras are initialised in separate processing threads and OpenCV used to capture and display the web camera stream as well as transforming the frames to greyscale and processing the greyscale images to detect the (x, y) pixel position of the charuco marker. In parallel, PyOpenXR was used in another thread to capture the (x, y, z) position of the HMD. The code synchronised the results and produced a list of input tensors $(x_1, y_1, \dots, x_n, y_n)$ and output tensors (x, y, z) for the web camera and HMD, respectively. NaN values were used if the computer vision was unable to detect the charuco marker in a web camera frame.

Windows showing the web camera views as well as a 3D point cloud of the HMDs path is shown to the user. The user specifies a set time they want to capture for and they move the HMD through the domain they wished to spatially track hands. Two comma separated variable files containing the input and output tensors are generated at the end of the specified time and past to the Training step of the process.

The Training step accepts the tensors generated in the Calibration step and trains a set of ML algorithms to predict the (x, y, z) position of a point detected as $(x_1, y_1, \dots, x_n, y_n)$ from the web cameras. An ML is trained for each combination of web cameras where data is available. The ML selected was a linear regression model with 2-degree polynomial features (i.e., x, y, x^2, y^2, xy). The model was selected to take the curvilinear (fisheye lens) behaviour typically exhibited in many camera lenses into account. No information was required on the cameras used and their relative and absolute positions that would typically be needed for stereoscopic computer vision based calibrations. The result of the Training step was a set of ML models stored in a user-specified directory.

With the models trained, the pipeline can start to detect the spatial position of hands with respect to the VR environments global coordinate system. The Track step uses the web cameras to detect hand landmarks in 2D camera pixel coordinates and the ML models use the data to predict the location of the hands spatial position. The web cameras operate separate threads with OpenCV gathering the stream data. The stream data is then piped through Mediapipe – a ML model developed by Google researchers – that detects 21 hand landmarks (e.g., joints and knuckles) in (x, y) pixel values.

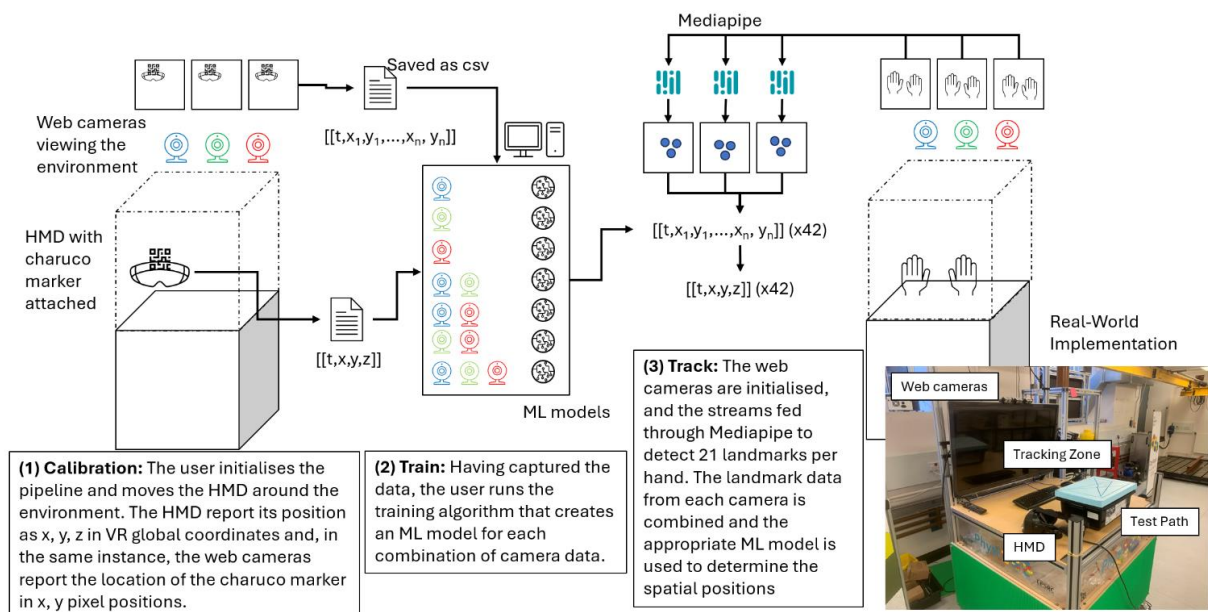


Figure 3. A low-cost non-intrusive spatial hand tracking pipeline

The landmark positions for each camera are then synchronised with NaN values used when the hands were not detected. The synchronised data passes to the ML models and, based on the combination of data available, the appropriate ML model was selected to predict the (x, y, z) for each joint.

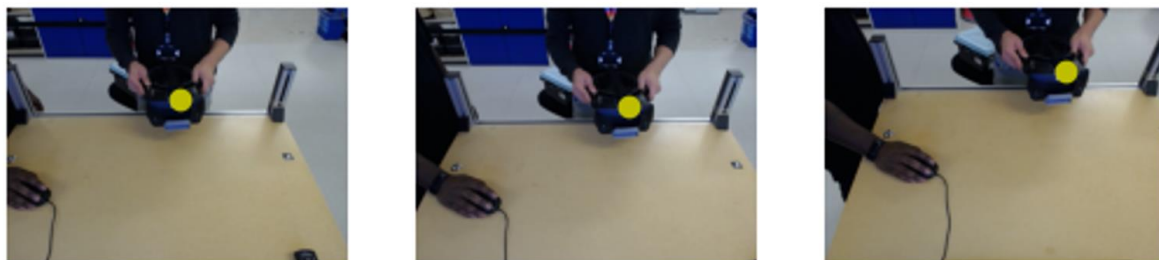
The pipeline can operate in real-time with plots showing the spatial position of an individual's hands. Data can be recorded for further post-processing and used to create spatial video replays. Data can also be streamed to a VR environment, such as Unity or Unreal, to enable a user's hands to be made visible in the VR environment where they can become game objects and used to interact with other game objects in the environment – controller-free interaction.

Affordances include the ease of set-up and operation, the low-cost in infrastructure needed to run, and scalability with n web cameras permitted. The web cameras can be positioned in any location to help with occlusion as well as where it is practical in the environment where an individual is going to be performing the product-process interaction.

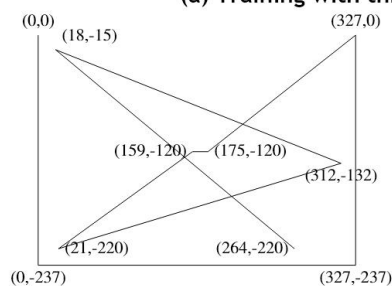
4. Experiment

Cox et al. (2023) hand tracking study was used to evaluate the performance of the pipeline. The pipeline was configured to track hands in the area shown in Figure 4a. The set-up used three cameras resulting in 7 ML models for the camera combinations.

A participant then traced a pre-defined path etched into a foam board using their index finger (Figure 4b). The participant placed their finger in the start position and the pipeline initiated. When the camera views appeared on the computer screen, and they could see their hand in the viewers, the participant was instructed to start tracing the path. The participant was asked to trace it in a steady and continuous motion and to keep their finger in the final resting position until the cameras deactivated. The participant had 1min 30secs to complete the tracing task.



(a) Training with three cameras identifying the Charuco markers on a HMD.



(b) Tracking using Mediapipe on three cameras simultaneously to identify 21 points on the hand and using the (x, y) data to predict the (x, y, z) if the points were the HMD in the virtual world.

Figure 4. An individual configuring and using the pipeline to detect hand position

The calculated hand positions were then stored and later analysed to evaluate the performance of the system. Performance was measured through four metrics:

1. Capture success rate
2. ML training statistics
3. Tracking completeness
4. Path accuracy

The capture success rate evaluated the calibration step and whether each of the cameras were able to identify the HMD charuco marker during the capture exercise. A 100% successful capture would mean all the cameras reported data during the calibration exercise. The ML training statistics examined the prediction performance of the 7 ML models. The analysis took 10% of the training data as a test set and used the Root Mean Square Error as the performance measure. Tracking completeness assessed the performance of the tracking element of the pipeline. 100% would mean that the index finger was identified by the three cameras throughout the task. Path accuracy was evaluated through two metrics. The first analysed the histogram of z values being returned from the pipeline. Given the path was on a flat plane relative to the VR's global coordinate system, the expectation was that the reported z values would be tightly bounded around a single mean value. It is important to note that each camera was angled down on the area and thus, the path would not appear flat relative to their co-ordinate systems. The second metric examined the index landmarks x , y positions captured during the task and compared them against the ideal path. The distance of the captured positions with respect to the ideal path lines was computed using the following equation:

$$d = \frac{|ax_0+by_0+c|}{\sqrt{a^2+b^2}} \quad (1)$$

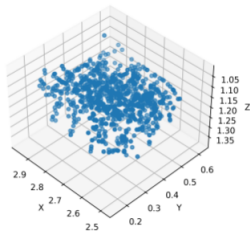
Where d is the distanced a , b and c are the coefficients of the line and x_0 and y_0 is the point away from the line. The smallest difference from the point and the path lines was taken as the error.

5. Results

A total of 12,947 data points were captured during the calibration step (15 min of moving the HMD around the scene of interest). Table 2 shows an excerpt of the captured data. 44% (5,673) of the captured data featured x , y pixel positions from all three cameras. Given that neither camera was occluded during the calibration step, there exists an opportunity to improve the processing of the charuco marker. On inspection of the videos captured, motion blur from moving the HMD during the calibration step was deemed to be a main contributing factor to the data loss.

Table 3 presents the result from the ML training statistics for the 7 camera combinations – input data size and RMSE score. The input data sizes reveal the impact of the data loss with fewer points for the two and three camera combinations. Nonetheless, the RMSE score shows a significant performance improvement when two or more cameras are used to predict spatial positions. This was to be expected as a single camera lacks sufficient data to determine depth therefore relying on the probability distribution of the historic data to give an indication of where the object is likely to be. The addition of more than two cameras offers little increase in performance and thus, multiple cameras only need to be considered for situations where occlusion or data loss may be an issue. The accompanying figure shows how the accuracy improves with the size of dataset for the 3 camera ML model with the model quickly converging to an accuracy of 0.00725. Additional data aids in reducing the variance in prediction. The results indicate that individuals do not need to spend considerable time collecting and training the pipeline.

Table 2. An excerpt of data collected during the calibration phase



#	Cameras						HMD		
	x_1	y_1	x_2	y_2	x_3	y_3	x	y	z
1	3.35E+02	2.07E+02	4.24E+02	2.21E+02	NaN	NaN	2.84E+00	1.21E+00	3.21E-01
2	3.36E+02	2.07E+02	NaN	NaN	4.86E+02	1.57E+02	2.84E+00	1.21E+00	3.23E-01
3	NaN	NaN	4.26E+02	2.24E+02	NaN	NaN	2.84E+00	1.20E+00	3.24E-01
4	3.38E+02	2.15E+02	NaN	NaN	NaN	NaN	2.84E+00	1.20E+00	3.26E-01
5	3.39E+02	2.18E+02	4.27E+02	2.32E+02	4.88E+02	1.68E+02	2.85E+00	1.20E+00	3.29E-01
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
-5	3.40E+02	2.21E+02	NaN	NaN	NaN	NaN	2.85E+00	1.19E+00	3.31E-01
-4	3.41E+02	2.23E+02	4.29E+02	2.37E+02	4.89E+02	1.73E+02	2.85E+00	1.19E+00	3.33E-01
-3	3.41E+02	2.26E+02	NaN	NaN	NaN	NaN	2.85E+00	1.19E+00	3.36E-01
-2	3.41E+02	2.28E+02	4.30E+02	2.43E+02	NaN	NaN	2.85E+00	1.18E+00	3.40E-01
-1	3.41E+02	2.31E+02	NaN	NaN	4.89E+02	1.82E+02	2.85E+00	1.18E+00	3.45E-01

Table 4 details the results from the path tracing study. A total of 2,556 data points were captured during the exercise. In addition, to the pipeline was able to identify the hand using all three cameras for the

entire task, which enabled the three camera ML model to be used to predict the spatial location of the index finger. The first path analysis metric was the variance in the z value and is shown in Figure 5a as a histogram. The minimum, maximum, mean and standard deviation were 0.988m, 1.084m, 1.067m, and 0.004m, respectively. This shows there is very little movement of the hand in the z-direction showing that the depth can be accurately captured through the pipeline.

Table 3. Training results for different camera combinations (12,947 data points total)

Camera Combination	Input Data Size	RMSE
0	9473	0.042
0,1	7309	0.008
0,1,2	5673	0.007
0,2	7167	0.007
1	9585	0.046
1,2	7344	0.008
2	9352	0.050

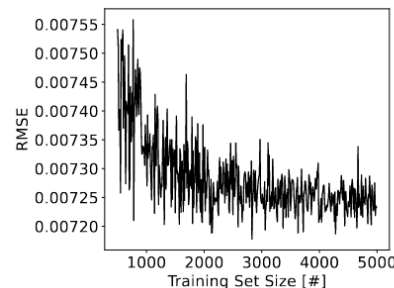
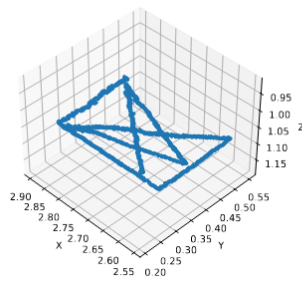
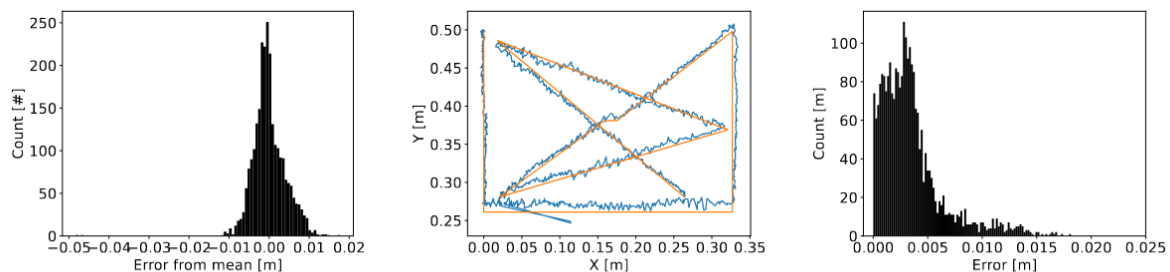


Table 4. Captured points and the number of camera that captured them



Camera Combination	MediaPipe Detection
0	0
0,1	0
0,1,2	2556
0,2	0
1	0
1,2	0
2	0



(a) Variance in the z-direction. (b) Path captured compared to ideal path. (c) Distance away from ideal path. Normalised w.r.t. the mean.

Figure 5. An individual configuring and using the pipeline to detect hand position

Figure 5 also shows the variance of the x, y values reported with the idealised path overlaid in Figure 5b and a histogram of the error against the path in Figure 5c. The minimum, maximum, mean and standard deviation are 0.000m, 0.018m, 0.003m, and 0.003m, respectively. The results again show a high degree of accuracy from the pipeline featuring cameras situated approximately a metre away from the working area.

6. Discussion

The results of the pipeline have shown promise that a user's spatial hand interaction of a product-process within a design and/or manufacturing context captured. The approximate accuracy of 3mm offers the opportunity to capture product-process interactions. Further, there are very few constraints in terms of

configuring and deploying the pipeline and the relative low-cost hardware used compared to other solutions on the market could support wider adoption.

The study reported remains preliminary and further evaluations performing a variety of product-process interactions are required to provide a more complete picture of the utility of the pipeline. Improvements and future work include:

- Providing guidelines in terms of camera positioning or adjustment in order to provide the highest resolution of the intended working area.
- Post-processing the path data to detect, flag and remove anomalous readings.
- refinement of the pipeline codebase to increase data capture and processing rates.
- Post-processing the paths and classifying ours of activity within the path.
- Testing and validating the tracking accuracy with a larger group of participants.
- Testing the usability of the pipeline for designers as a design support tool.
- Examining the ethical and privacy implications of capturing hand tracking data through the pipeline.

The pipeline was tested with web cameras operating at 720p 30FPS resulting in a data rate of 1260 landmark readings per second. A 10 min session would give a 756,000 data points. The pipeline can therefore amass a large amount of data on a product-process interaction and future work is required to interpret this data to provide valuable insights and meaning for designers to act on. Examples include post-processing the data to identify common activities and/or interactions that can be classified through language, such as grasping, twisting, and turning. Post-processing the data in this manner would product product-process interaction narratives that designers can interpret and act upon. The data could also be replayed through 3D and immersive VR environments providing Spatial Video Replays. It would be interesting to compare the insights generated by designers using the spatial video replays compared to contemporary video recordings for the product-process interaction. One hypothesis is that they could enhance the learning of a process as individuals can view the interaction episode from different angles and even overlay their hands with the hands of the replay to help them emulate the experience. The replays could also form the input into soft robotic applications where they wish to emulate an individual in performing a process.

The pipeline can also provide spatial hand positions in real-time. The data could then be fed into the VR environment that it has been trained for to provide controller free user interaction in VR systems. This could be useful for designers when interacting with geometry in VR and enable emulation of experiences such as clay modelling.

7. Conclusion

There has and continues to be a desire to capture product-process interaction across design, manufacturing, and prototyping activities. This paper contributes a pipeline that captures spatial hand positions. The preliminary study showed the pipeline can attain an accuracy of ≈ 3 mm. Further advantages of the pipeline are its minimal constraints on set-up, ability to handle occlusion through multiple cameras, and low-cost relative to commercial solutions. These solutions are opening a whole new stream of data regarding product-process interaction (30-60FPS of 42 hand landmarks) and the next step is to learn how these solutions can provide information and insights that could support design.

Funding Statement

The work has been undertaken as part of the Engineering and Physical Sciences Research Council (EPSRC) grants - EP/W024152/1.

Data Availability Statement

The data that support the findings of this study are openly available at <https://dmf-lab.co.uk/21st-century-prototyping-technologies/>.

Disclosure Statement

The authors report there are no competing interests to declare.

References

- Buckingham, G. (2021). Hand tracking for immersive virtual reality: Opportunities and challenges. *Frontiers in Virtual Reality*, 2. <https://doi.org/10.3389/frvir.2021.728461>
- Caeiro-Rodríguez, M., Otero-González, I., Mikic-Fonte, F. A., & Llamas-Nistal, M. (2021). A systematic review of commercial smart gloves: Current status and applications. *Sensors*, 21(8). <https://doi.org/10.3390/s21082667>
- Chien, C. - F., Kerh, R., Lin, K. - Y., & Yu, A. P.- I. (2016). Data-driven innovation to capture user-experience product design: An empirical study for notebook visual aesthetics design. *Computers & Industrial Engineering*, 99, 162–173. <https://doi.org/https://doi.org/10.1016/j.cie.2016.07.006>
- Cox, C. M. J., Hicks, B., Gopsill, J., & Snider, C. (2023). From haptic interaction to design insight: An empirical comparison of commercial hand-tracking technology. *Proceedings of the Design Society*, 3, 1965–1974. <https://doi.org/10.1017/pds.2023.197>
- Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., ... Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825), 357–362. <https://doi.org/10.1038/s41586-020-2649-2>
- Helminen, P., Hamalainen, M. M., & Makinen, S. (2010, August). Redefining User Perception: A Method for Fully Capturing the User Perspective of a Product Concept (Vol. Volume 5: 22nd International Conference on Design Theory and Methodology; Special Conference on Mechanical Vibration and Noise). <https://doi.org/10.1115/DETC2010-28698>
- Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3), 90–95. <https://doi.org/10.1109/MCSE.2007.55>
- Itseez. (2015). Open source computer vision library.
- Johnston, S. H., Berg, M. F., Eikevåg, S. W., Ege, D. N., Kohtala, S., & Steinert, M. (2022). Pure vision-based motion tracking for data-driven design – a simple, flexible, and cost-effective approach for capturing static and dynamic interactions. *Proceedings of the Design Society*, 2, 485–494. <https://doi.org/10.1017/pds.2022.50>
- Joong Hee Lee, W. K., & Yun, M. H. (2023). Development of a therblig-based evaluation methodology for accessible product: A case study of spinal-cord impaired users [PMID: 37477263]. *Disability and Rehabilitation: Assistive Technology*, 0(0), 1–11. <https://doi.org/10.1080/17483107.2023.2235378>
- Karras, O., Unger-Windeler, C., Glauer, L., & Schneider, K. (2017). Video as a by-product of digital prototyping: Capturing the dynamic aspect of interaction. 2017 IEEE 25th International Requirements Engineering Conference Workshops (REW), 118–124. <https://doi.org/10.1109/REW.2017.16>
- Lederman, S. J., & Klatzky, R. L. (2009). Haptic perception: A tutorial. *Attention, Perception, & Psychophysics*, 71(7), 1439–1459. <https://doi.org/10.3758/APP.71.7.1439>
- Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C. - L., Yong, M. G., Lee, J., Chang, W.- T., Hua, W., Georg, M., & Grundmann, M. (2019). Mediapipe: A framework for building perception pipelines.
- Merriau, P., Dupuis, Y., Boutteau, R., Vasseur, P., & Savatier, X. (2017). A study of vicon system positionin performance. *Sensors*, 17(7). <https://doi.org/10.3390/s17071591>
- Openshaw, S., & Taylor, E. (2006). Ergonomics and design a reference guide. Allsteel Inc., Muscatine, Iowa.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011).
- Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- Schiffstein, H. N., & Cleiren, M. P. (2005). Capturing product experiences: A split-modality approach. *Acta Psychologica*, 118(3), 293–318. <https://doi.org/https://doi.org/10.1016/j.actpsy.2004.10.009>
- Shi, J., Tang, W., Li, N., Zhou, Y., Zhou, T., Chen, Z., & Yin, K. (2021). User cognitive abilities-human computer interaction tasks model. In D. Russo, T. Ahrum, W. Karwowski, G. Di Bucchianico, & R. Taiar (Eds.), *Intelligent human systems integration 2021* (pp. 194–199). Springer International Publishing.
- Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C. - L., & Grundmann, M. (2020). Mediapipe hands: On-device real-time hand tracking.

