INTERNATIONAL DESIGN CONFERENCE – DESIGN 2024
https://doi.org/10.1017/pds.2024.88

6 degree of freedom positional object tracking for physical prototype digitisation

Michael Wyrley-Birch, Aman Kukreja, James Gopsill, Christopher Michael Jason Cox and Chris Snider ✉

University of Bristol, United Kingdom

✉ chris.snider@bristol.ac.uk

Abstract

Underpinning much work on the use of Virtual Reality technologies in design prototyping, is the need to reliably track the 3D position of a physical object in real space, then allowing synchronisation with a digital counterpart. With many tracking methods requiring changes to object geometry, this work develops and benchmarks four minimally invasiveness 6 DoF tracking approaches, before discussing their use in a prototyping context. Results show that using AI and point cloud methods, accuracies of 20mm at 20Hz are achievable on low-end hardware with no alterations to the prototype needed.

Keywords: prototyping, virtual prototyping, virtual reality (VR)

1. Introduction

Prototyping has been used throughout engineering to improve the efficacy of different design processes (Sass and Oxman, 2006). Many different prototyping methods within industries help people better characterise the product being produced (Camburn et al., 2017). With the advent of digital technologies, new and exciting prototyping methods have been unlocked, allowing virtual models to be created, bringing many advantages but lacking the benefits of physical objects. These include digital prototypes lacking physical tactility but far cheaper to produce (Liker and Pereira, 2018). Combining these two domains into a more cohesive system could unlock new and interesting ways for users to interact with the product during the design phase. One way to achieve this is to use a series of low-fidelity physical prototypes that the user interacts with as a one-to-one map for a high-fidelity digital prototype allowing for quick and easy refinement of the final ergonomics and possible use cases of the product. For example, if different drill handles needs to be tested to find the most comfortable one, several physical prototypes with different grips could be tested by giving them to a user in virtual reality, allowing them to interact with the virtual world using the physical drill.

To do this, the real-world position and orientation of the prototype must be captured in real-time and with enough accuracy and precision that, to the human controlling the system, there is no dissonance between the virtual and real worlds (Caserman et al., 2019). The tracking must also not be invasive; geometries or weight characteristics of the prototype must not change dramatically, as this would affect how people interact with the prototype (Desanghere and Marotta, 2015). While there are many systems for capturing the 6D pose estimation of an object, there is little to no research into the invasiveness of solutions for their applicability in prototype digitalisation. Therefore, this paper proposes and explores different generic 6D pose estimation methods, measuring their tracking potential against invasiveness. The main aim of the developed system is to estimate the position and rotation of a generic prototype in real-time, measure the difficulty to implement it, and determine its level of invasiveness.

For the purpose of this study, four different pose estimation methods were produced and benchmarked against the points discussed above on a variety of other objects. Additionally, the shortcomings and improvements to the methods specified are discussed. The paper continues with a summary of what generic 6D pose estimation means and related work (Section 2), followed by the methodology for the implementation and benchmarking (Section 3), the results produced by the benchmarks (Section 4), a discussion with future work (Section 5) and finally a conclusion (Section 6).

2. Background

Seamless integration of Computer-aided design (CAD) modelling software and 3D printing has created a new way to quickly generate low-fidelity prototypes from virtual models. These prototypes provide tactility needed for basic user-interactions but lack the potential variability and complex interactions a high-cost prototype could afford such as haptic feedback, geometric variations, aesthetic modifications, etc. One way to remedy this is to accurately track the low-fidelity prototype in 3D space, then using that model as a controller linked to a Virtual Reality or other computer simulation then examine product variations. Tracking objects in 3D space is a complex problem with many technical challenges related to occlusions, computational speed, ambient lighting conditions, etc.

2.1. Pose estimation

Six Degrees of freedom (6D) pose estimation is the process of calculating the position and rotation of an object in 3D space. There are many different methods for object tracking using a variety of technologies. These include hardware-type solutions, where specific trackers are attached to an object and software-type solutions, where Computer Vision techniques are used to estimate the location and rotation of an object. Some examples of these are shown below (Table 1).

Table 1. Different methods for object tracking

Fiducial markers (i.e. ArUco makers)	Markers of known size and shape are placed on the object allow 6 DoF pose estimation. (Boonbrahm et al., 2020)
MoCap	Retroreflective markers placed on the object are tracked using an array of infrared cameras. (Chatzitofis et al., 2019)
Deep learning-based approaches	These methods aiming to give the pose of an object from a video sequence. (Castro and Kim, 2023; Nguyen et al., 2022; Tuscher et al., 2021)
VR hardware trackers	These aim to give reliable pose estimation based on internal hardware1.

The strengths of each approach vary depending on usecase. In robotics, deep learning-based approaches have been used to track the location of objects for robotic arms to grasp (Tuscher et al., 2021). In Augmented reality, ArUco markers anchor virtual objects in the real-world (Boonbrahm et al., 2020). The latencies, accuracy, and jitter characteristics of these methods have been looked at thoroughly. However, not much consideration has been given to the level of modification of shape, the invasiveness, and the total difficulty of integrating a potential solution into any workflow. If the geometry or visuals of the objects are modified too much, the user may focus more on the modification instead of the underlying prototype, potentially breaking the engagement of the user with the prototype (Desanghere and Marotta, 2015). In addition, the object tracking methods should be as accessible as possible, and easy and low-cost to implement. For example, adding a Vive tracker to a prototype dramatically changes its geometry, with a large amount of setup time and cost. Therefore, this paper aims to explore different tracking solutions and their performance, invasiveness, and implementation difficulty.

3. Method

To assess the suitability of prototype tracking, four methods of 6-DoF pose estimation were developed and benchmarked on a variety of different object geometries[2]. Each method was benchmarked to characterise advantages and disadvantages.

---

[1] https://www.vive.com/uk/accessory/tracker3/

[2] GitHub link: https://github.com/DMFDML/GenericObjectTracking

3.1. Metrics

To make the tracking of prototypes as convincing as possible, it must track in real-time accurately with little jitter in the estimation (system performance), not alter the geometry or visuals of an object to the point that it breaks the user's immersion (invasiveness) and be easy to integrate into any workflow so anyone can use it (difficulty to implement). Table 2 shows the measured metrics, each was chosen as a proxy for the points needing to be tested. The best case for positional and rotational estimations is based on hand and controller tracking as the tracking should be as accurate (Abdlkarim et al., 2023; Batmaz et al., 2021). The invasiveness of a solution is broken down into: Non-Invasive (no alterations to the object needed); Visually Invasive (visual look of the object affected); Semi-Invasive (geometry and visuals affected a small amount); Very invasive (object geometry, visuals, and weight affected); Extremely invasive (object geometry, visuals and weight massively affected).

Table 2. Table of different metrics used to evaluate the pose estimation solutions

Metric	**Reason for capturing**	**What it captures**	**Best case**
Latency of system	System performance	Time taken to output position in an operational system.	0.058 s (Caserman et al., 2019)
Position estimation	System performance	The accuracy of the position estimation and the amount of jitter.	Accuracy: 0.011 m Jitter: 0.00375 m
Rotation estimation	System performance	The accuracy of the rotation estimation and the amount of jitter.	Accuracy: 0.377 rad Jitter: 0.00873 rad
Level of invasiveness	Invasiveness	Alterations to object geometry required by the method.	Non-Invasive
Cost to implement	Difficulty of implementation	Cost of implementation	Vive tracker: £358
Skill level to implement	Difficulty of implementation	Difficulty of setup and workflow integration	All steps are trivial

3.2. System development

Based on the different tracking methods specified in Section 2, four different tracking methods were created, each with varying characteristics of invasiveness (Table 3).

Table 3. Different tracking methods development

Method	**Reason for development**
ArUco Markers	Industry standard, used in many augmented reality applications so is easy to implement, low cost and quick but be quite invasive.
Reflectors with Plane Calculations	MoCap reflectors are used for easy object recognition. This method estimates the pose faster than the other methods, is visually invasive but gives less accuracy .
Reflectors and ICP	Generic object tracker, using MoCap reflectors to capture position as an input to the Generic tracker. This approach is computationally less intensive than the generic tracker, but is semi-invasive.
Generic	This method uses deep learning-based approaches and was developed to be completely non-invasive and give good pose estimations.

3.2.1. ArUco markers

ArUco markers are widely supported and easy-to-use pose estimation method that quickly gives reasonable 6D estimations from one camera. It uses fiducial markers of a specified size and the intrinsic distortion coefficients of the camera to estimate the 3D location and pose of the marker. The markers must be flat and readily visible, so alteration of object geometry was required. (Figure 1).

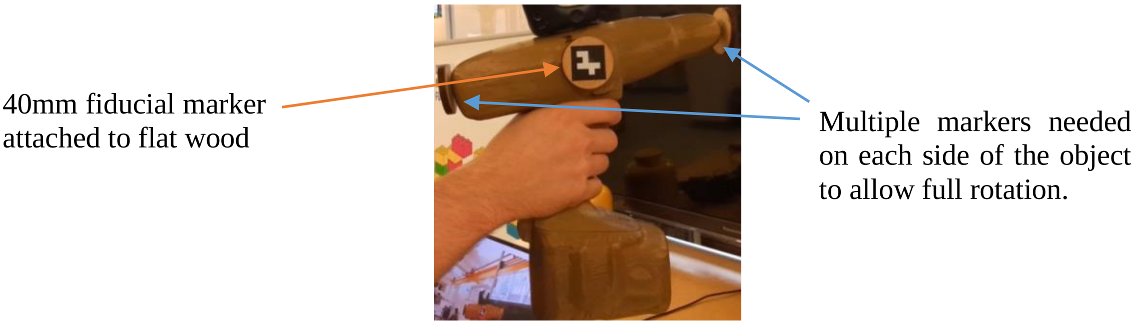

Figure 1. ArUco marker on drill prototype

3.2.2. Reflectors and plane calculations

In this method, an Infrared (IR) camera is used to get a clear image of the reflector, which is then detected and clustered. These points and clusters generate planes and their normals using a least-squares solution. A weighted average is then applied to the normals, using the residual error, to get the average direction the object faces and, therefore, its pitch and yaw. The 3D location of the object is obtained from the average depth of all the points. This results in this method giving 5D-pose estimation. Nonetheless, it was determined to be worth assessing due to its latency advantage over other methods (Figure 2).

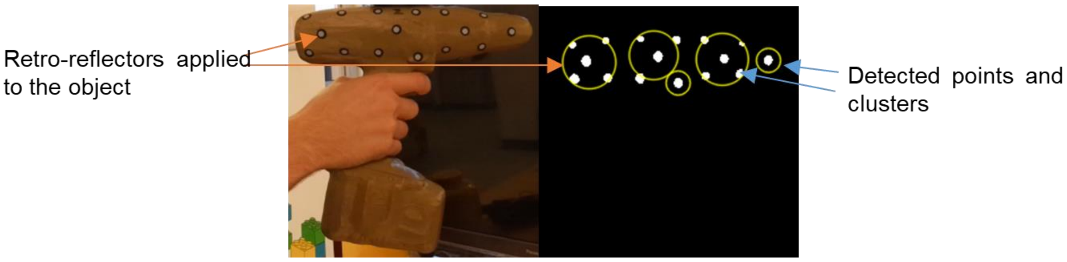

Figure 2. Reflectors and plane calculations on drill prototype

3.2.3. Reflectors and ICP

This method uses an Infrared (IR) camera to get a clear image of reflectors on the object which are detected and, using an alpha shape algorithm, generate a polygon of just the object in the image. This polygon is then used to generate the mask and bounding box needed for the Iterative Closest Point (ICP) registration method described in the Generic Tracker section (Figure 3). This was produced due to the latency advantage over the Generic tracker and the potential for more accurate pose estimations, at the cost of invasiveness, due to the mask created covering the object better.

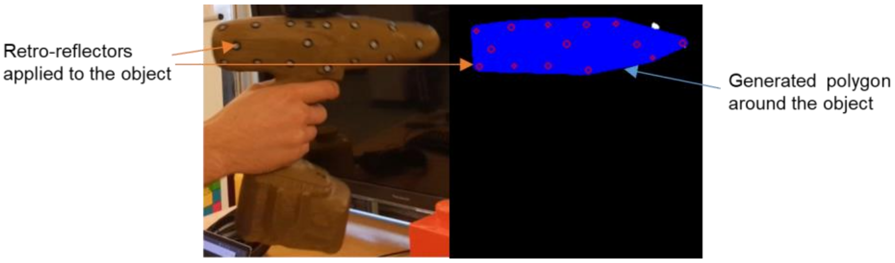

Figure 3. Reflectors and ICP on drill prototype

3.2.4. Generic tracker

The generic tracker developed in this work takes the inspiration from the method proposed by Tuscher et al., 2021. It uses the 2D tracking algorithm SiamMask (Wang et al., 2019) and ICP (Zhang, 2014). The method was implemented and modified to work better for our workflow. It consists of three stages, creation of the reference point cloud for the object, processing of the next frame of the video using SiamMask to generate a bounding box and mask of the object, this is used to segment the point cloud

of the new frame, and finally determine the the difference in rotation and translation between the mask of the object and the reference using ICP to generate the 6D pose estimation of the object.
Two main improvements were made to the method:

a) Better point cloud filtering to speed up the registration: A progressive threshold filters the point cloud points outside n standard deviations of the mean, and n is increased by 0.1 until there are enough points for ICP to work well, eliminating many erroneous points.
b) Progressively increasing point cloud size: When the object is rotated to a position in which there is insufficient point cloud data, the reference point cloud is updated. This occurs when the Inlier Root Mean Squared Error (RMSE) is small enough for a suitable match but high enough to show a lack of information. The overlap can be further refined by running ICP with more iterations and using Robust Kernels, then merging and voxelised to reduce size (Figure 4). This has the additional benefit of generating a high resolution point cloud during the tracking process.

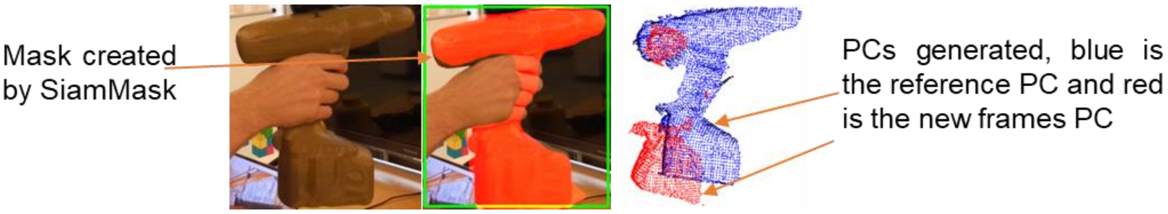

Figure 4. Generic tracking on drill object

3.3. Experimental procedure

Three objects were chosen to find out how the different tracking methods perform on various geometries: a coin, a cylinder, and a drill (Figure 5). They were chosen for their unique properties. The coin was relatively small and flat with high symmetry, characteristics that are typically hard to track. The cylinder had rotational symmetry along one of its axes, making it hard to estimate its rotation. The drill had complex geometry to simulate the performance hit of a larger PC. As such, the coin and cylinder present challenging cases for tracking with high potential uncertainty, while the drill presents a realistic case of a product.

Two test cases were derived for the objects to test the performance of each tracking method. In both cases, the tracked object was placed on a turntable at a known position, with a known rotational speed. As such, in each test the object was subject to translation and rotation, with motions then representing simple motions as may be made during the prototyping process. The camera was then positioned above the table so that it was in the centre of the frame, and a series of images of the object moving around the table was captured at a fixed framerate (30FPS). For each test videos were processed to track positions depending on hardware used, using a laptop (i5 processor, 4GB RAM).

In Test 1, the position of the object was calculated from the video using the radius of the object on the table, the rotational speed, and the frame rate of the camera, the ground truth roll (Figure 6). Test 2 involved using a Vive tracker, which, as specified in Section 2.1, is a widely used solution for pose estimation in VR applications due to its low latency, high accuracy and precision. However, it is very invasive. Its estimation was used as ground truth values to compare the solutions against a known, suitable solution and benchmark the metrics not captured in the previous test (pitch, yaw, and depth). A Vive tracker was placed on an object (Figure 6), and its values were read simultaneously as the object tracking solution specified.

Figure 5. Picture of the chosen objects

Figure 6. Experiment 1 and 2 setups

4. Results

This section outlines the results obtained from the test cases specified in Section 3. Table 4 shows the average results from benchmarking across all tests, with the best results in each category being italicised. As all tests were run on the same hardware, the Average Time was be used as a relative measure between tracking methods. These results are therefore representative of latency on typical hardware, but should be considered only as a relative measure. Although the Vive tracker wasn't benchmarked, it has been added to show the difference between the methods produced and a common high-accuracy but highly invasive solution (Kuhlmann de Canaviri et al., 2023).

Table 4. Benchmarking results for different trackers

Tracker	**Average time (s)**	**Rotational Accuracy (rad)**	**Rotational SD (rad)**	**Positional accuracy (m)**	**Positional SD (m)**	**Invasiveness**	**Price (£)**
ArUco	0.0078	0.8052	0.5964	0.1046	0.0545	Semi-Invasive	***0.00***
Reflectors and plane calculations	***0.0041***	1.4215	0.5269	0.0853	0.0662	Visually Invasive	374.00
Reflector-ICP	0.0470	***0.3637***	0.4731	0.1115	0.0622	Visually Invasive	374.00
Generic	0.0648	0.3957	***0.4277***	***0.0265***	***0.0309***	***Non-Invasive***	355.00
Vive tracker	>0.01	0.0016	0.0080	0.0002	0.0004	Very Invasive	577

4.1. System performance

4.1.1. *Latency for frames*

Figure 7 shows the time taken to generate pose estimation for each video sequence frame. Red graph areas indicate performance worse than the best case scenario (0.058s specified in Section 3.1). The graph shows that for objects of a small enough size, all trackers can be considered real-time. However, if the object is big enough, the methods that use ICP take longer to generate a pose estimation due to the increased point cloud data throughput. The area 'A' in Figure 8 (Drill) shows a significant increase in the latency. This is most likely due to the reference point cloud being added as new sides of the object get revealed. The updating procedure takes time, causing massive spikes. The latencies after the 'A' are also higher due to the larger final point cloud, but stabilise when the entire object has been captured.

4.1.2. *How accurate and how much variation was in the estimation*

Figure 8 shows the root mean squared error between the pose estimation and the ground truth for each video frame. Having a value closer to 0 is the best. Like Figure 7, the graph is shaded with red above the best case accuracy (0.011m 0.377rad as specified in Section 3.1).

Table 4 shows that the average rotation of the reflector tracker is within the threshold at 0.3637 rad, and for the generic tracker method it is almost within the threshold at 0.0265m, mainly for the coin and cylinder objects as shown in the graph below. The jitter for both position and rotation, seen as rapid spikes on the graph, was relatively stable for all methods except the ArUco tracker.

The objects with rotational symmetry (coin and cylinder) are both tough to track consistently due to the tests being performed while rotating it around the axis of symmetry. The ArUco and reflector trackers could not reliably track the coin object due to the angle of the ArUco marker being too extreme and the size of the detected mask being too small. The reflector tracker with no ICP does not track well for the object benchmarks due to its lack of ability to track the axis of rotation for the benchmark.

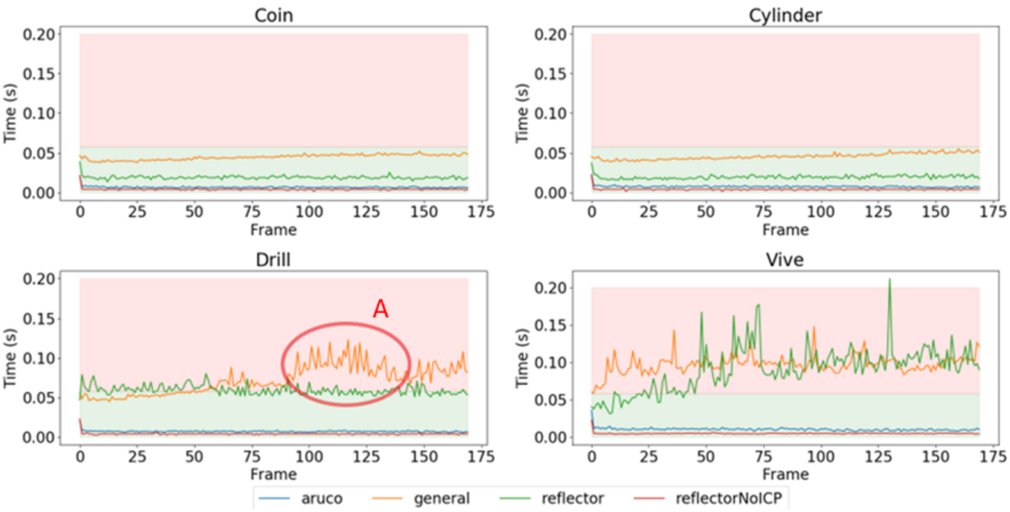

Figure 7. Time to process each frame of the video

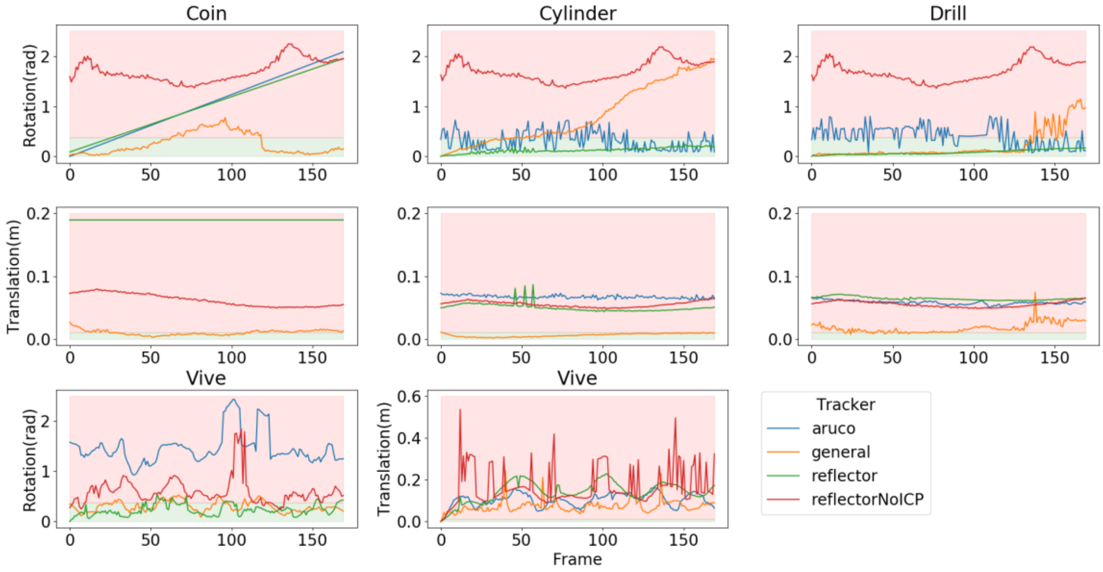

Figure 8. RMSE of the difference between pose estimation and ground truth for each frame

4.2. Invasiveness on all objects

Table 5 shows the level of invasiveness observed for different methods for the chosen objects. Evidently, the Generic Tracker was completely non-invasive and it thus meets the best case requirement (Table 2). The two trackers that use reflectors are visually invasive. However, the whole object doesn't need to have reflectors placed on it. For example, the drill object just had reflectors placed on the top part (Figure 3), suggesting it does not have to be invasive if the object is large enough. The level of invasiveness, however, does not give the complete picture of how intrusive the tracking method is due to the wide variety of possible object geometries. For example, ArUco markers have to be placed on a flat surface and be a specific size to track well, which meant that for the coin, they were only visually invasive, but for the drill, they were semi-invasive (Figure 1, Figure 5).

Table 5. Invasiveness of each tracker type, when located on each tested object

	ArUco	Reflectors and plane calculations	Reflector-ICP	Generic	Vive tracker
Coin	Visually	Visually	Visually	None	Extremely
Cylinder	Very Invasive	Visually	Visually	None	Extremely
Drill	Semi Invasive	Visually	Visually	None	Very

4.3. Difficulty to implement

Table 6 shows the cost and difficulty of implementing the solution to characterise how difficult it would be to integrate it into a workflow. The skill level to set up refers to the difficulty needed to integrate the solution into a workflow, and the skill level to implement refers to the difficulty required to implement the solution from scratch. The Generic Tracker was the easiest and cheapest to set up, but it was the hardest to implement due to the complexity of the code and the need for fine-tuning parameters. The ArUco Tracker was the easiest to implement due to its prevalence but hard to set up due to the need to calibrate the camera. If the base stations are already affixed to the walls, then the Vive tracker is straightforward due to its prevalence in the VR space. The implementation cost could be reduced by using cheap cameras in a multi-camera set up to calculate the depth, but this would dramatically increase the setup work needed. Importantly, reflector and Generic methods all require depth cameras rather than standard webcams greatly increasing cost, while the ArUco method operates with any optical camera.

Table 6. Cost breakdown of each solution

Solution	Implemented Price	Setup Work as Implemented
ArUco	Printing ArUco: negligible, ArUco Backing: negligible, Camera: standard webcam.	Printing ArUco: negligible, Setting up camera: installing Azure SDK, calibrating camera
Reflector with plane	Camera cost: £355.00 Reflector dots: £19.00	Adding Reflectors to object: negligible, Setting up camera: installing Azure SDK
Reflector-ICP	Camera cost: £355.00 Reflector dots: £19.00	Adding Reflectors to object: negligible, Setting up camera: installing Azure SDK
Generic	Camera cost: £355.00	Setting up camera: installing Azure SDK
Vive tracker	Base station x2: £438.00 Tracker: £139.00	Base station: attaching base stations to walls Setting up tracker: Installing SteamVR

5. Discussion

This section discusses the system's performance characteristics, the study's limitations as it is implemented, the potential value the system could bring and future work.

5.1. System performance

The developed system of object tracking was tested in a variety of scenarios with the coin and cylinder being the extreme cases in terms of complexity for tracking, and the drill object more realistic. Evidently, the results show that the average pose estimation performance is acceptable for the methods using ICP, giving the best rotational and positional accuracy of 0.3637rad and 0.0265m respectively. As seen in the Fig. 8, most of the RMSE values for Generic and Reflector ICP are in the green zone for position and rotational accuracies. However, the hardware used to test these methods is too slow for large geometries at 0.0719s and 0.0593s, respectively. The reflector tracking with plane calculation, on the other hand, worked the fastest at 0.0041s and gave relatively stable position estimation at 0.0853m accuracy with 0.0662m SD. However, it only provides 5D pose estimation. Besides, its rotational precision and accuracy are highly sensitive to the placement of the reflectors on the object, which can be seen by the high average 1.4215rad. The ArUco tracking was the second fastest at 0.0078s, which is real-time. However, the pose estimation generated is very sensitive to the camera's calibration, resulting in less accurate and less precise results than is possible with this method.

The level of invasiveness is also variable depending on the object's geometry, but all the methods are visually invasive if the object has flat points in its geometry (Table 5). If the solution needs to be for Virtual Reality, then visual invasiveness isn't a problem because it does not require object to be seen. Whereas, for augmented reality the method must be entirely non-invasive, suggesting that only the Generic Tracker would work for such applications.

When tracking objects, occlusion (where some obstacle prevents measurement of the object of interest) is often a major issue. For all systems using fiducial markers and IR reflectors this remains an issue when grasping the object in the hand, as the hand may cover the markers. While increasing number of markers (i.e. for the IR system) ensures some are always visible, further studies should consider optimisation with respect to occlusion. The generic tracker however minimises this issue via it's approach; when scanning with a consistent hand position, the hand itself becomes part of the tracked system. When the hand is removed, the system then updates the point cloud with the new geometry and continues. Accordingly, this presents an opportunity for robust tracking when hand occlusion is a risk.

5.2. Study implementation and limitations

All the solutions are straightforward to use as they only need a camera, printer, and reflective dots, which are all readily available. They do not need specific training and runs on relatively low-end hardware. This study ran all benchmarks on a processor with eight cores and a GPU with 4GB of usable memory, and as such all computationally-dependent results should be considered as relative to the hardware used. Improvements to hardware massively decrease latencies. The benchmarking techniques used in this paper, while they were attempted to be as accurate as possible, could have problems in the data collection. The potential errors could have occurred while capturing the video sequence due to: the variable speed of the turn table, the frame rate not being constant, or a possible imperfect alignment of the camera with the centre of the turn table, etc. This would cause an error in the ground truth calculation for the turn table benchmark. The same data collection problems could have happened when using the Vive tracker resulting in lower performance.

While the geometries used were chosen to be either worst-case or representative of typical products, they are not broad in scope. Further studies should consider a broader geometry set and effect of their form on detection accuracy. Further, while tests included translation and rotation, tracked motions were limited in scope and controlled. Further studies should also test precision on broader motion sets.

Furthermore, all the code for this paper was written in Python, which is slower compared to other languages, like C++, and thus hampering performance of all solutions (Fourment and Gillings, 2008). With code optimisation and parameter tuning, the time taken to generate frames for the methods using ICP could be improved to real-time with far better performance characteristics on different objects.

5.3. Potential value and future work

The tracking of prototypes could be used in a variety of different use cases. These include the potential to easily compare prototypes with different ergonomic characteristics by giving the participants low fidelity 3D printed models and observing them in Virtual Reality environment. Depending on the design process used, the Generic object tracking solution could help synchronise the physical and digital prototypes by scanning the object and enhance the exploration. As the reference point cloud being updated every time new information is found, the proposed method can also be used to create a low-cost point cloud scanner.

6. Conclusion

This paper has presented four different methods of 6D pose estimation in the context of tracking generic physical prototypes and drive its corresponding digital model. The methods were benchmarked and rigorously tested for their performance, invasiveness, and difficulty in implementation. Overall, the Generic tracker was found to give the best performance as it achieved the rotational and positional accuracies of 0.3957rad and 0.0265m with the potential for massive latency gains if run on better hardware. The ArUco marker based method were better suited for flat surfaces, whereas, ICP and reflector based methods could be used for 5D pose estimation of smaller objects. The observations drawn from this work on object tracking methods can be used to select the best method for user specific applications. While new generic object tracking methods are being developed rapidly, the design research community's next objective could be to investigate the process of generating a digital prototype using the estimated pose, with the aim of supporting the design refinement and exploration by accessing real time modification of the object.

Acknowledgements

The work reported in this paper was conducted at the University of Bristol in the Design and Manufacturing Futures Laboratory (www.dmf-lab.co.uk) as part of EPSRC grant ref: EP/W024152/1.

References

Abdlkarim, D., Di Luca, M., Aves, P., Maaroufi, M., Yeo, S.-H., Miall, R.C., Holland, P., Galea, J.M., 2023. A methodological framework to assess the accuracy of virtual reality hand-tracking systems: A case study with the Meta Quest 2. Behav. Res. Methods. https://doi.org/10.3758/s13428-022-02051-8

Batmaz, A.U., Seraji, M.R., Kneifel, J., Stuerzlinger, W., 2021. No Jitter Please: Effects of Rotational and Positional Jitter on 3D Mid-Air Interaction, in: Arai, K., Kapoor, S., Bhatia, R. (Eds.), Proceedings of the Future Technologies Conference (FTC) 2020, Volume 2, Advances in Intelligent Systems and Computing. Springer International Publishing, Cham, pp. 792–808. https://doi.org/10.1007/978-3-030-63089-8_52

Boonbrahm, S., Boonbrahm, P., Kaewrat, C., 2020. The Use of Marker-Based Augmented Reality in Space Measurement. Procedia Manuf., International Conference on Industry 4.0 and Smart Manufacturing (ISM 2019) 42, 337–343. https://doi.org/10.1016/j.promfg.2020.02.081

Camburn, B., Viswanathan, V., Linsey, J., Anderson, D., Jensen, D., Crawford, R., Otto, K., Wood, K., 2017. Design prototyping methods: state of the art in strategies, techniques, and guidelines. Des. Sci. 3, e13. https://doi.org/10.1017/dsj.2017.10

Caserman, P., Martinussen, M., Göbel, S., 2019. Effects of End-to-end Latency on User Experience and Performance in Immersive Virtual Reality Applications, in: van der Spek, E., Göbel, S., Do, E.Y.-L., Clua, E., Baalsrud Hauge, J. (Eds.), Entertainment Computing and Serious Games, Lecture Notes in Computer Science. Springer International Publishing, Cham, pp. 57–69. https://doi.org/10.1007/978-3-030-34644-7_5

Castro, P., Kim, T.-K., 2023. PoseMatcher: One-shot 6D Object Pose Estimation by Deep Feature Matching. https://doi.org/10.48550/arXiv.2304.01382

Chatzitofis, A., Zarpalas, D., Kollias, S., Daras, P., 2019. DeepMoCap: Deep Optical Motion Capture Using Multiple Depth Sensors and Retro-Reflectors. Sensors 19, 282. https://doi.org/10.3390/s19020282

Desanghere, L., Marotta, J.J., 2015. The influence of object shape and center of mass on grasp and gaze. Front. Psychol. 6.

Fourment, M., Gillings, M.R., 2008. A comparison of common programming languages used in bioinformatics. BMC Bioinformatics 9, 82. https://doi.org/10.1186/1471-2105-9-82

Kuhlmann de Canaviri, L., Meiszl, K., Hussein, V., Abbassi, P., Mirraziroudsari, S.D., Hake, L., Potthast, T., Ratert, F., Schulten, T., Silberbach, M., Warnecke, Y., Wiswede, D., Schiprowski, W., Heß, D., Brüngel, R., Friedrich, C.M., 2023. Static and Dynamic Accuracy and Occlusion Robustness of SteamVR Tracking 2.0 in Multi-Base Station Setups. Sensors 23, 725. https://doi.org/10.3390/s23020725

Liker, J.K., Pereira, R.M., 2018. Virtual and Physical Prototyping Practices: Finding the Right Fidelity Starts With Understanding the Product. IEEE Eng. Manag. Rev. 46, 71–85. https://doi.org/10.1109/EMR.2018.2873792

Nguyen, V.N., Du, Y., Xiao, Y., Ramamonjisoa, M., Lepetit, V., 2022. PIZZA: A Powerful Image-only Zero-Shot Zero-CAD Approach to 6 DoF Tracking. https://doi.org/10.48550/arXiv.2209.07589

Sass, L., Oxman, R., 2006. Materializing design: the implications of rapid prototyping in digital design. Des. Stud. 27, 325–355. https://doi.org/10.1016/j.destud.2005.11.009

Tundra Labs [WWW Document], n.d. . Tundra Labs. URL https://tundra-labs.com/en-gb (accessed 8.3.23).

Tuscher, M., Hörz, J., Driess, D., Toussaint, M., 2021. Deep 6-DoF Tracking of Unknown Objects for Reactive Grasping.

Wang, Q., Zhang, L., Bertinetto, L., Hu, W., Torr, P.H.S., 2019. Fast Online Object Tracking and Segmentation: A Unifying Approach. https://doi.org/10.48550/arXiv.1812.05050

Zhang, Z., 2014. Iterative Closest Point (ICP), in: Ikeuchi, K. (Ed.), Computer Vision: A Reference Guide. Springer US, Boston, MA, pp. 433–434. https://doi.org/10.1007/978-0-387-31439-6_179