A Quantitative Comparison of Position Trackers for the Development of a Touch-less Musical Interface

Gabriel Vigliensoni
Input Devices and Music Interaction Laboratory
Centre for Interdisciplinary Research on Music Media and Technology (CIRMMT)
McGill University, Montréal, QC, Canada
gabriel@music.mcgill.ca

Marcelo M. Wanderley
Input Devices and Music Interaction Laboratory
Centre for Interdisciplinary Research on Music Media and Technology (CIRMMT)
McGill University, Montréal, QC, Canada
marcelo.wanderley@mcgill.ca

ABSTRACT

This paper presents a comparison of three-dimensional (3D) position tracking systems in terms of some of their performance parameters such as static accuracy and precision, update rate, and shape of the space they sense. The underlying concepts and characteristics of position tracking technologies are reviewed, and four position tracking systems (Vicon, Polhemus, Kinect, and Gametrak), based on different technologies, are empirically compared according to their performance parameters and technical specifications. Our results show that, overall, the Vicon was the position tracker with the best performance.

Keywords

Position tracker, comparison, touch-less, gestural control

1. INTRODUCTION

A touch-less gestural interface is a type of *alternate controller*, which neither resembles nor is inspired by any acoustic instrument, and falls under the sub-category of *expanded-range controllers*, which require little or only limited physical contact to play them [9]. This kind of non-contact musical instrument must be tailored to the performer's position, orientation, and movement. These variables need to be measured in a non-intrusive manner, i.e., without restricting the performer's movements, in order to do not limit the performance. Hence, when designing an open-air musical interface, several considerations must be issued for sampling the performance space according to the performance needs, but without limiting it.

1.1 Touch-less gestural control: sampling the space

A number of different techniques and technologies can be used to sense and measure the position of points in space, so selecting the proper sensing technology to fulfill the needs of a specific project is a critical step in the development of a musical interface. The devices that implement these technologies are commonly known as *position trackers*, and share some common characteristics that describe their behaviour beyond their specific sensing method. These characteristics are know as the *performance parameters* of the trackers and are key-factors in their response. Some of these performance parameters are their *accuracy*, *precision*, *jitter*, *operating range*, *drift*, *latency*, and *tracker update rate*, and are reviewed in [1]. Although these parameters can provide a good picture about the response of a position tracker and its particular features, in musical practice some of these characteristics are not so relevant in comparison with other fields of study. Hence, another suite of parameters, inherent to the context of live-music, appear, e.g., the system should be portable but robust, easy to set up and calibrate, cheap or reparable, immune to environmental on-stage conditions, and so on [11]. Still with these constraints, the system must measure the performer's musical gestures with accuracy and let the user map these gestures with flexibility to the synthesis engine. Furthermore, to develop expert performance with a new musical instrument, a deterministic behaviour is decisive, so to develop a reliable interface we need to know in advance how the system will respond to the same stimuli in diverse contexts.

2. TRACKER COMPARISON

In this section we provide a summary of the technical specifications of four position trackers based on different technologies. We also review some of their intrinsic constraints and present an experimental comparison of their performance parameters.

2.1 Trackers' technical specifications

Knowing that during the last 10 years or so many controllers with tracking technology have come from the game industry and have had large impact in the community of musicians looking for new ways of interact with music, we decided to compare some of their performance parameters with other systems used in professional motion capture or 3D modelling applications.

The compared trackers were the Vicon 460[1] motion capture system, the Polhemus Liberty 240/8[2] magnetic-based motion tracking system, the Microsoft Kinect[3] computer-vision based system, and the In2Games Gametrak, a mechanical, tethered-based position tracker [10] that is not precisely touch-less, but provides 3D position tracking. While the former two devices are considered professional position trackers, the latter are consumer-oriented systems used in game consoles.

NIME'12, May 21 – 23, 2012, University of Michigan, Ann Arbor.

[1]http://www.vicon.com/, accessed January 23, 2012
[2]http://www.polhemus.com/?page=MotionLiberty/, accessed January 23, 2012
[3]http://www.microsoft.com/en-us/kinectforwindows/, accessed January 23, 2012

2.1.1 Vicon 460

The Vicon 460 is an optical tracker. It uses several cameras to calculate and track the spatial position of beacons attached to body parts or objects using triangulation. The system we tested comprises six Vicon M2 cameras, the Vicon Datastation hardware module, and the Vicon iQ2.5 and Eclipse workstation software packages. The Vicon Datastation handles all the camera synchronization and coordinates the capture and generation of video data. Although the software packages allows to record, organize, analyze, and present the data, we did not use it (except for storing the calibration set up) because our experiment considered to extract the position data from the system in real-time.

2.1.2 Polhemus Liberty 240/8

The Polhemus Liberty 240/8 is an electromagnetic-based tracker system. It features an electronic control unit with the hardware and software to generate and sense magnetic fields, and to measure the position and orientation of up to 8 sensors 240 times per seconds. For our experiment, we used the Polhemus Long Ranger source, which extends the sensing area to a diameter of about three meters. Furthermore, being aware of the distortion that ferromagnetic and metallic surfaces create in the measurements of magnetic-based position trackers, before testing the Polhemus we removed all metallic devices and elements in the room.

2.1.3 Microsoft Kinect

Although there was not official information from Microsoft about the technology behind the Kinect, PrimeSense, the company that provides the raw tracking technology to Microsoft, has released reference design information about the PrimeSensor, the Kinect device, and its technology [4]. The Kinect system comprises the camera and software that processes all data acquired by the camera. The system extracts a two-dimension (2D) image from a video camera, but it is able to extract a third dimension by illuminating the scene with patterns of infrared light. The reflected patterns change depending on the distance from the device to the object were the light is reflected. The system analyses the deformation of the patterns and reconstructs a depth map of the image. In this moving image, the system looks for shapes that resemble the human body and tracks 15 predefined points in it. When these points are detected, the Kinect tracks them and calculates their position.

2.1.4 In2Games Gametrak

The Gametrak is a two-joystick, tethered-based system capable of measuring 3D position of up to two points in space. Two analog potentiometers-based joysticks measure the x and y position for two points in space. To calculate the z position, another two potentiometers, rolled in a retractable spring-loaded drum, quantify the number of turns each potentiometer does when the nylon tethers are extended [10]. The implementation of the system seems very simple, but a complex mechanical system is used to guide the nylon tethers in order to have a clean path [3]. The housing of all potentiometers, nylon tethers, interface, and joysticks is a weighted box that allows the user to pull out the chords without moving the box. The Gametrak system converts the acquired spatial data using the USB-HID (Universal Serial Bus - Human Interface Device) protocol.

2.2 Workflow Pipeline

Since the systems reported their data in different ways and different scales, to do a proper comparison with similar characteristics, a common workflow pipeline was designed for each one of the trackers. The pipeline considered the following stages: data acquisition, data parsing, data normalization and mapping, and data recording and processing. Hence, the results of the experimental comparison will be related to the whole workflow pipeline, including the tracker and all the data processing across the different applications.

2.2.1 Data Acquisition

The data acquisition stage refers to the process of setting up and calibration of the systems. Because each one of the position trackers has its own sensing method and characteristics, we took different approaches for all of them.

The Vicon system has the most detailed and complex calibration method of the trackers we tested. The cameras of the system have to be properly positioned to cover all the sensed space, and all major light reflections (e.g., from shiny metal objects) must be removed from the camera view before starting the process of calibration.

The Polhemus ideally requires a "benign" environment, free of ferromagnetic materials or metallic surfaces, otherwise, the measurements will suffer from distortion. As we did not have this ideal space we reduced the amount of metal structures and objects in the room as much as possible. However, we were aware on how much metal elements exist in the floor, ceiling, and walls of the laboratory. Moreover, as we wanted to test these systems for performance situations, we could not use *PiMgr*, the Polhemus proprietary software, to extract and send the reported measurements in real-time. *PiMgr* can compensate the distortion from metal objects according to a compensation map, but it does not provide a way for extracting its data in real-time.

Since its release in 2010, different groups of people have been working on ways to extract the tracking data from the Kinect. We tested several systems and ended up using the OpenNI framework library[4] because it provided the smallest latency. The Kinect, in companion with the OpenNI library, has a calibration procedure that needs a specific user's pose. Once calibrated, the system tracks up to 15 points located in the pre-defined places of the human body. However, it was not possible to know exactly where the system was exactly at, during the calibration process or the data acquisition stage, because it uses a proprietary signal processing stage to determine the location of these points.

The Gametrak does not have a calibration method, and to extract its data we modified it.[5] It requires the user to grab their plastic tethers, and though the device is heavy, for a proper measuring it must be solidly attached to a surface to avoid changes in its position.

2.2.2 Data Parsing

To record and handle all tracker data we used a different computer from the one for the data acquisition. Therefore, we needed a way to extract, parse, and send all data in real-time from the trackers to the recording computer.

To extract the data from the Vicon system, we followed previous research [2] and used the *QVicon2OSC* application.[6] This application bridges the Vicon motion capture data to OSC, and is capable of sending the position of user-defined points, and the rotation of objects created by the user. Once structured and sent as an OSC message, the data was received in Max/MSP with a customized version of the *OSCeletontoQC*[7] object.

[4]`http://www.openni.org/`, accessed January 23, 2012

[5]`http://x37v.com/x37v/post/labels/sensors.html/`, accessed January 23, 2012

[6]`http://www.sonenvir.at/downloads/qvicon2osc/`, accessed January 23, 2012

[7]`http://mansteri.com/download/software/osceletontoqc/`, accessed January 23, 2012

To acquire and parse the data from the Polhemus Liberty tracker to OSC messages, we used the library and command-line front-end *plhm*.[8] This application is capable to request data from the Polhemus Liberty and send it through a network as OSC. Once parsed, we received the Polhemus data in Max/MSP.

To retrieve the data from the Kinect system, we used *OSCeleton*,[9] an OSC proxy for Kinect Skeleton data. This software establishes communication with the OpenNI framework, and allows a user to scale and offset the data. We opted for not using these last features because we would perform a data post-processing in a later stage of the workflow pipeline. The OSC data was also received in Max/MSP.

Once the Gametrak was modified, its data measurements could be opened directly in Max/MSP because the device is natively recognized as an HID object. Its raw data was formatted as an OSC message for further processing.

2.2.3 Data Normalization and Mapping

As the data reported by each of the trackers has its own scale, we normalized it to a common scale, so that a comparison could be made. In addition, as we were handling 3D position data from several trackers, and we were using different recording objects, a tool for flexible mapping was required. We used the *libmapper*[10] library to map and scale all signals from the trackers to the recording computer. *libmapper* is capable of discovering devices in a network and showing their previously declared inputs and outputs. Also, a user can arbitrary map those input and output ports with any kind of scaling of the signals going through the mapper [8]. In our experiment, we measured the reported maximum and minimum data by each one of the trackers, and used the normalization capabilities of *libmapper* to send values between -1.0 and 1.0. The normalized output of all trackers was routed again to Max/MSP.

2.2.4 Data Recording and Processing

To record the normalized data from all trackers we used the *Digital Orchestra Toolbox's* `dot.recordabsolute` Max/MSP object [7]. This object allows to record arbitrary number of data streams with absolute time-stamping, thus making it possible to compare the update rate of the trackers. In terms of data processing we first parsed all data to a common structure, and then we performed a *change of basis*. This process allowed us to virtually locate each tracker's origin at the same point by translating and rotating the vectors measured in one basis (the tracker basis) to another one (the normalized-space basis). We could then measure the vectors for each point in the space with the same, normalized reference space.

2.3 Experiment design and set up

To compare the reported data by the trackers and their data pipeline, we used the experimental approach described by Kindratenko [6] to compare the accuracy of two tracking systems. He collected the reported data by each tracker at known, nominal, two-dimensional locations, and calculated their accuracy in those points. Although his results are well documented and clear, his experiment only dealt with measurements taken on a 2D plane. Hagedorn et al. [5] designed another experiment to correct the data measured by an electromagnetic motion tracking system. They developed a 3D grid by means of crates, and collected the position tracker reported data at known locations. Because of the nature of the magnetic-tracking system, they used plastic crates so as not to distort the measurements of the tracker.

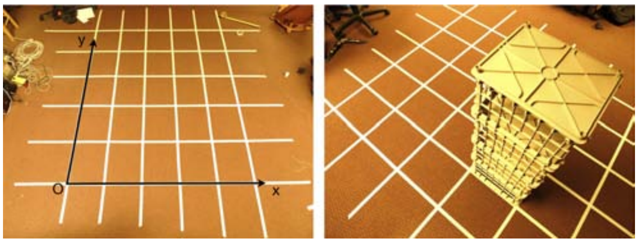

Figure 1: 2D grid on the floor and the plastic crates used to have the same grid at different heights.

For our experiment, we used a combination of the aforementioned approaches to design our experiment. We created a evenly-spaced grid on the floor and used lockable, plastic crates to have the same grid at different heights. The two-dimensional grid on the floor and crates on the grid are shown in Figure 1.

The dimensions of the room, and the systems' minimum and maximum tracking distance, made us to work with a sensed space of 0.95m * 1.28m * 1.06m, which corresponds to the points located at the center of the grid. To cover this space, we did 80 measurements where the markers or sensors for each system were located on each vertex of the plastic crates,[11] as can be seen in Figure 2.

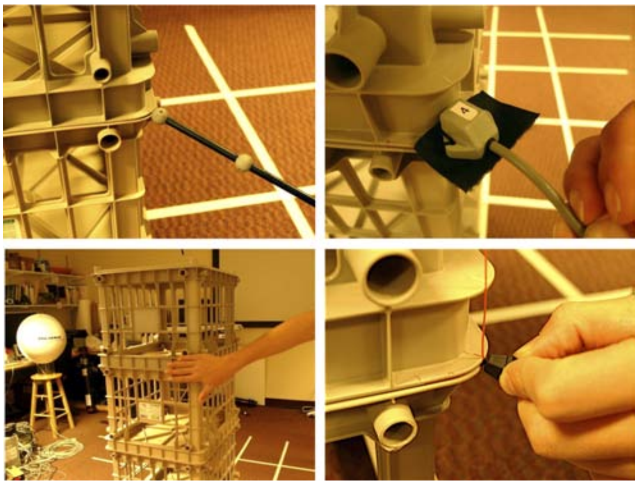

Figure 2: Measurement of the same point in space with the four tracking systems. From upper left to bottom right: Vicon, Polhemus, Kinect and Gametrak.

In order to measure the accuracy, precision, and update rate of the systems, we recorded the tracker's reported data during the lapse of one second. We then took the mean of the reported values and obtained a more representative point to interpolate lines along each axis and plane.

[8]`http://idmil.org/software/plhm/`, accessed January 24, 2012

[9]`https://github.com/Sensebloom/OSCeleton/`, accessed January 23, 2012

[10]`http://idmil.org/software/libmapper/`, accessed January 24, 2012

[11]It should be noticed that error can affect the data acquired by the trackers. The error types can range from non-uniform crate manufacture, to human-error in the creation of the grid, in the placement of the crates on the grid, or in the positioning the sensors at the measured points in the crates. The actual error amount was not measured, though.

3. RESULTS

In this section we present a summary of the experimental results. They are grouped by shape of the reported space, accuracy and precision, and tracker update rate for each one of the four systems.

3.1 Reported space

As mentioned before, we measured points at discrete places in the space, over a lapse of one second. The amount of reported values depended on the update rate of each one of the trackers. To visualize the shape of the space reported by the trackers, we interpolated lines between points on each axis per plane of measurements. Figure 3 shows a 3D, isometric view of the reported space by all trackers. Red dots in the plots represent the measurement for each point. The closer the points are to a nominal, integer value, the more accurate the system is; and the smaller of the red zones are, the larger the precision is. Furthermore, the straighter the blue lines are, the less distorted the space representation is. The scale of the plots is normalized, (the value for each axis is equivalent to each side of a plastic crate, our calibration object). It can be seen that the reported space by the Vicon is very close to the original. The blue lines are mostly straight, making clear that the Vicon system senses the space evenly along its axes.

The shape of the space measured by the Polhemus in our test-room, with its unique conditions, was particular. It can be seen that the middle-line along the x and z axis are relatively straight. Also, lines along y, close to the magnetic source, start straight but bend at the end. Furthermore, the most distant-to-the-source points were reported far away from their nominal position. Equally, red points closer to the magnetic source show less variability than those located at a larger distance. These issues can be explained by the presence of ferromagnetic elements in the floor and ceiling that could not be removed. Also, the magnetic field decreases with the distance, and so there was more distortion in measurements taken in points further away from the source, shifting their reported position. Although we were expecting a larger linear space with the Polhemus Long Ranger source, our results show that even with a powerful electromagnetic field, the condition of common rooms affect the measurements of magnetic trackers, leaving only a small zone, close to the source, with an acceptable accuracy. It should be noticed, though, that rooms like ours' are frequent environments for music performance, and so this should be considered to determine the tracked space in an actual performance context.

We can see that the measured points for the Kinect are not located in the nominal positions along the three axis, making the reported space be curved, but also showing that the tracker is not accurate. Furthermore, there is a large variability in the position of the red points, and so this system is far less precise than the Vicon. However, the overall shape of the sensed space is still a deformed, wavy cube.

For the Gametrak, the red points are not located in the nominal position of the measurement, but in a "radial" cube. This rectangular shape was scaled-dependant on the distance from the tracker to the grid level which was measured, and so the values became larger for longer distances. This unexpected behaviour was rather consistent for all points in the measured space. Moreover, the red points were more sparse than in the Vicon or Polhemus, indicating that the system is less precise than those trackers, as expected.

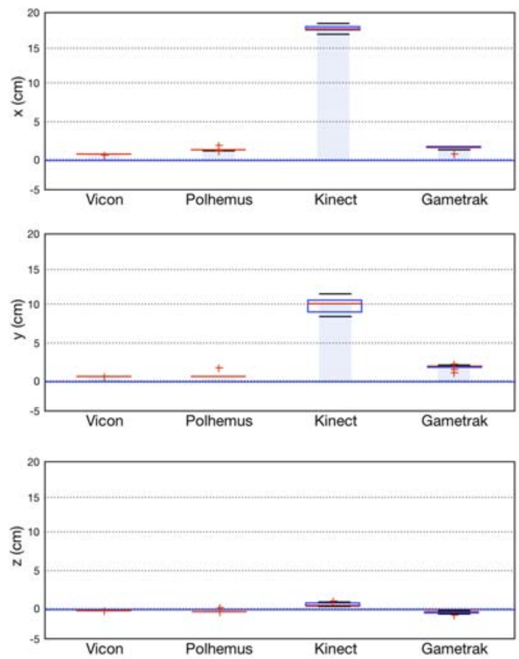

Figure 4: Accuracy and precision of the reported data by the four trackers at the origin.

3.2 Accuracy and precision

In our experiment we did measurements at discrete points inside a normalized space. To evaluate the accuracy of each system, we calculated the mean of all measurements, and compared it with the nominal, actual value of the position we measured. The closer the mean value to the actual integer point meant a higher accuracy of the system. At the same time, small deviations of the measurements compared with the nominal position, implied a higher precision.

Figure 4 shows the data reported by the four trackers at the origin of the system on the x, y, and z axis. The blue line represents the nominal, actual position of the measured point. Overall, the system with the best performance in terms of accuracy and precision is the Vicon, with median values along the axes close to 0, and little variability of its measurements. This high precision is shown by the small size of the boxes and whiskers for the Vicon data. The Polhemus also shows good performance in terms of accuracy, but less than the Vicon. Also, a closer look to the box and whisker plot shows that the distribution of the reported data is more sparse, meaning that it is less precise, in this setup and environment. The Kinect is the least accurate of all trackers, reporting a median value for x and y with offsets close to 17 cm and 10 cm, respectively, and it is also the less precise, with the largest variability. The plot for the Gametrak shows median values very close to the Polhemus, but with less precision.

3.3 Data measurement rate

For comparing the data measurement rate of the trackers, we subtracted the time-stamps on two consecutive measurements, over all measurements. By doing so, we could observe how constant in time each tracker reported its data, as well as to calculate the average rate of the measurements. As mentioned before, it is important to keep in mind that these values did not represent the update rate of each tracker alone, but the whole chain of processes and tools we used. Figure 5 shows box and whisker plots for the update rate for each system.

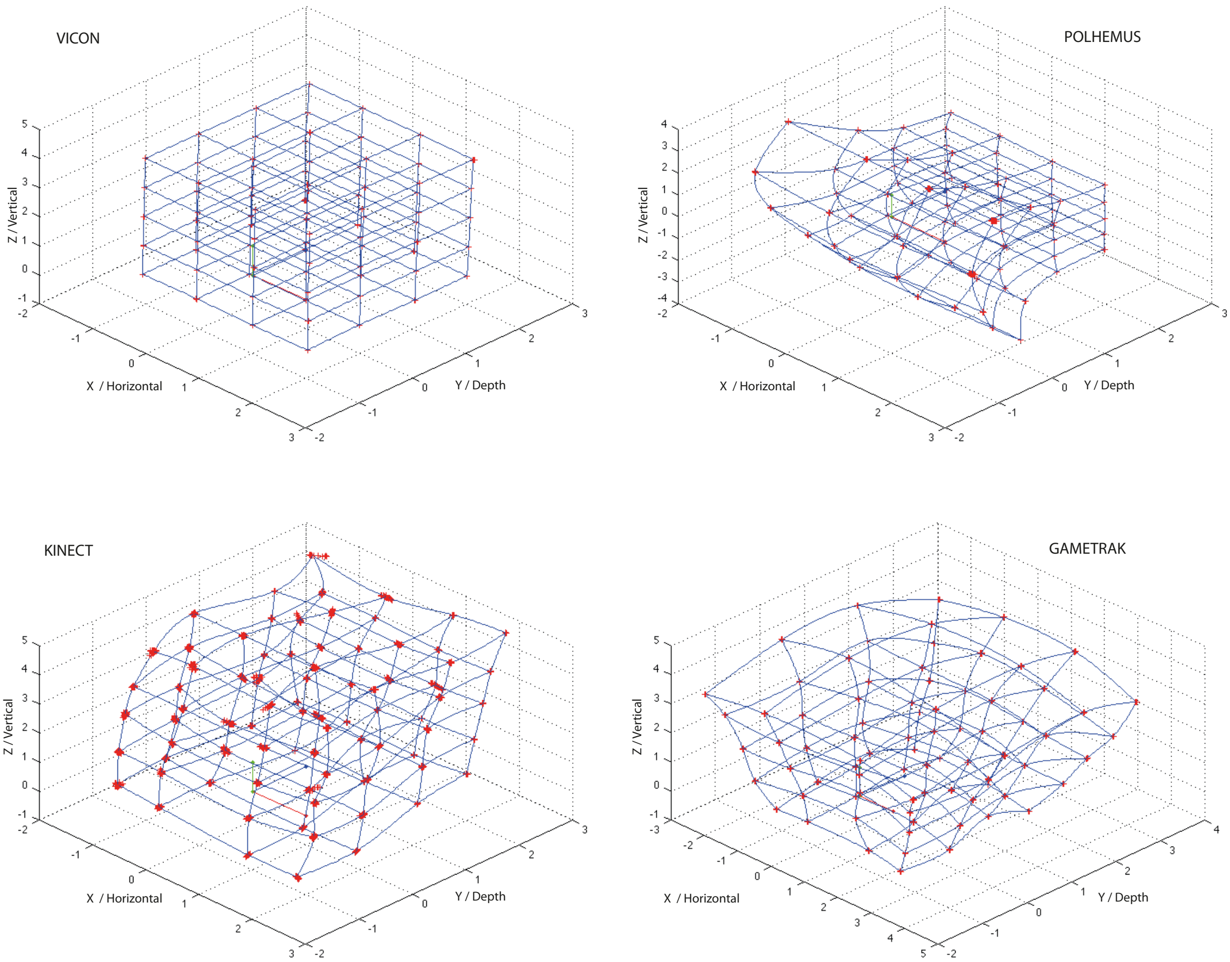

Figure 3: Shape of the space reported in a normalized scale by the four position trackers. Accuracy can be seen as how close the red points are to their actual nominal position in the grid, and precision to how spread the points are in a certain place. From upper-left to bottom-right: Vicon, Polhemus, Kinect, and Gametrak.

The Vicon system reported spatial position with a median rate of 100Hz, but with variability between 71Hz and 125Hz. Furthermore, there is a large number of outlier data points which are not part of the boxer and whisker because they are outside of the distribution curve. Analyzing the raw data, we realized that some measurements were sent after longer times than the average update rate, but the next ones came very close to the previous one. Also, in the Vicon software we selected an update rate of 240Hz, which is clearly faster than what we obtained in our experiment. Because of these two issues, we speculated that there was a data bottleneck in some part of the data flow for the Vicon. As we used almost the same pipeline for all systems, and we did not find this kind of bottleneck with the other trackers, the problem should be located in the unique stages of the workflow pipelines. For the Vicon, this stage corresponded to the QVicon2OSC application, which does not offer a way to monitor the signals it parses and processes.

The Kinect output median rate is 31Hz. It has the lowest sample rate of the trackers we tested, but it is the more stable, though. The Polhemus reported an average of 236 points per second, which is close to its actual specification. However, when we analyzed the data arrival time, we realized that its distribution was extremely sparse. To isolate and analyze this issue, we recorded the data internally with *PiMgr*, the Polhemus' proprietary application, and obtained similar results to those when we sent OSC messages to another computer. Figure 6 shows the difference in the arrival time between points over a lapse of 15 seconds. The median of the time differences is 4ms. However, every 256ms, there is delay of 20ms in the next reported point which is compensated later. Furthermore, a much larger correction, in the order of 260ms is performed by the system every 3500ms. There are also three peaks that are not corrected (circa 4200ms, 8500ms, and 12000ms). This behaviour of the Polhemus is interesting. The system is reporting the number of points per second that it should report, but unevenly. We reviewed again the documentation and were intrigued by the fact that some documents stated that the system provided 240 *measurements in one second*, per sensor, but other documents stated that the *update rate* of the system is 240Hz. Both statements are different, while we confirmed that the former is true, we proved that the latter is false. Figure 5 shows a median update rate for the Gametrak close to 100Hz. It can be seen a large deviation of the data distribution, meaning that the Gametrak reported its data in a very unsteady way over time.

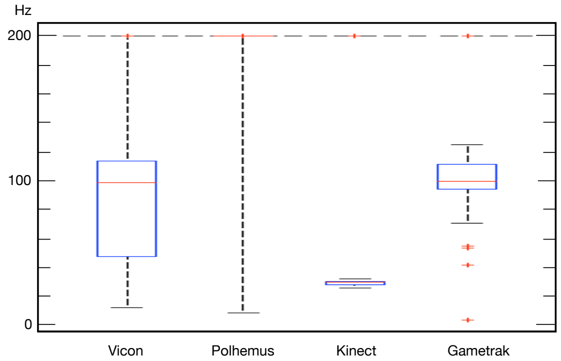

Figure 5: Update rate for the four position trackers.

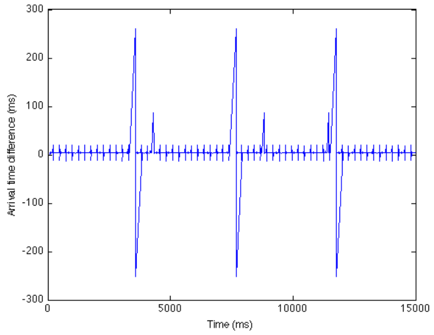

Figure 6: Difference in the arrival time between points over a lapse of 15 seconds for the Polhemus

4. CONCLUSIONS AND FINAL REMARKS

We have presented an experimental comparison of four position tracker systems. The accuracy and precision of the systems, the shape of the space they sense, and their update rate have been measured in the same environment with similar conditions, and a common workflow pipeline has been used to compare their reported data.

Overall, the tracker with the best performance is the Vicon 460 Motion Capture system. The shape of the space it senses is the closest to the nominal space, and is the most precise and accurate of all the position trackers we tested. In terms of its update rate, however, the workflow pipeline we used with the Vicon provides less than half of the rate we expected according to our settings of the system. The measurements of the Polhemus are biased by the presence of metallic surfaces and objects in the room that we could not remove. Because of these conditions, the space it reports is close to the actual space, but only in a limited region, close to the source. Beyond a limit, the system reports the position of the points very biased toward the ceiling, floor, and walls. In terms of its output rate, the Polhemus reports values at an uneven rate, but it compensates this shifts in time to meet its declared specifications. The Kinect lacks of precision and accuracy in all zones of the space. However, the shape of the space it senses is close to the nominal space. The workflow pipeline we used in companion with the Kinect has the most stable output rate of all systems, but at the same time is the slowest one. It is, however, the system with the simplest set up and calibration processes, and it is immune to changes in lighting conditions. Finally, the Gametrak mechanical tracker reports a curved space. Points in the vicinity of the device are reported close to their actual position, but distant ones are gradually deviated from their nominal position, following the path of curved, concentric lines with their origin in the device. The arrival time of the Gametrak measurements is reasonably constant and fast for the requirements of our system.

During this research, we have found that the design of a touch-less musical interfaces can be delineated by the performance parameters of the tracker we choose. At the same time, however, it can also be said that in musical practice, some of these characteristics could, or could not, be important, it depends on the gesture to track. For example, when tracking percussion playing gestures it seems sensible to have a very fast update rate, but a relatively small sensed space. On the other hand, when tracking hand's gestures to modulate processes, a slower update rate might be used but in larger sensed space. Finally, choosing the most appropriate system for tracking musical gestures in performance contexts requires a compromise between the technical parameters of the trackers, but also their practical considerations of use.

5. ACKNOWLEDGEMENTS

This research has been funded by BecasChile Bicentenario, CONICYT (Comisión Nacional de Ciencia y Tecnología), Gobierno de Chile, and NSERC Discovery Grant (Natural Sciences and Engineering Research Council of Canada), Canadian Foundation for Innovation.

6. REFERENCES

[1] G. Burdea and P. Coiffet. *Virtual reality technology.* John Wiley and Sons Inc., Hoboken, NJ, USA, 2003.

[2] G. Eckel, D. Pirro, and G. Sharma. Motion-enabled live electronics. In *Proceedings of the 6th Sound and Music Computing Conference*, Porto, Portugal, 2009.

[3] A. Freed, D. McCutchen, A. Schmeder, A. Skriver, D. Hansen, W. Burleson, C. Nørgaard, and A. Mesker. Musical applications and design techniques for the gametrak tethered spatial position controller. In *Proceedings of 6th Sound and Music Computing Conference*, pages 189–94, Porto, Portugal, 2009.

[4] B. Freedman, A. Shpunt, M. Machline, and Y. Ariely. Depth mapping using projected patterns. Patent US2010/0118123 A1, 2010.

[5] J. Hagedorn, S. Satterfield, J. Kelso, W. Austin, J. Terrill, and A. Peskin. Correction of location and orientation errors in electromagnetic motion tracking. *Presence: Teleoperators and Virtual Environments*, 16(4):352–66, 2007.

[6] V. Kindratenko. A comparison of the accuracy of an electromagnetic and a hybrid ultrasound-inertia position tracking system. *Presence: Teleoperators & Virtual Environments*, 10(6):657–663, 2001.

[7] J. Malloch, S. Sinclair, and M. Schumacher. Digital orchestra toolbox.

[8] J. Malloch, S. Sinclair, and M. M. Wanderley. From controller to sound: Tools for collaborative development of digital music instruments. In *Proceedings of the 2007 International Computer Music Conference*, pages 65–72, 2007.

[9] A. Mulder. *Design of virtual three-dimensional instruments for sound control.* PhD thesis, Simon Fraser University, Vancouver, BC, Canada, 1998.

[10] E. E. Myers. A transducer for detecting the position of a mobile unit. Patent GB 2373039 A, 2002.

[11] G. Vigliensoni. Touchless gestural control of concatenative sound synthesis. Master's thesis, McGill University, Montréal, QC, Canada, August 2011.