





Article

Improving Avatar Accuracy with Gaussian Process Regression Method in Mirror Metaverses

Mai Cong Huong ¹, Artem Volkov ^{1,2}, Ammar Muthanna ^{1,2}, Andrey Koucheryavy ^{1,2}, Dmitry Kozyrev ³ and János Sztrik ^{4,*}

¹ Department of Telecommunication Networks and Data Transmission, The Bonch-Bruевич Saint-Petersburg State University of Telecommunications, 193232 Saint Petersburg, Russia; mai.kkh@sut.ru (M.C.H.); artem.nv@sut.ru (A.V.); muthanna.asa@sut.ru (A.M.); akouch@mail.ru (A.K.)

² The M.I. Krivosheev National Research Centre for Telecommunication (NTRC), 105064 Moscow, Russia

³ Department of Probability Theory and Cybersecurity, Peoples' Friendship University of Russia named after Patrice Lumumba (RUDN University), 117198 Moscow, Russia; kozyrev-dv@rudn.ru

⁴ Department of Informatics Systems and Networks, Faculty of Informatics, University of Debrecen, Egyetem ter 1, 4032 Debrecen, Hungary

* Correspondence: sztrik.janos@inf.unideb.hu

Abstract

This paper deals with unwanted spatial distortion in virtual environments and its impact on the construction of metaverse environments that require high precision, especially in fields with specific requirements, such as medicine. At the same time, it presents the main technical factors leading to this phenomenon. The paper also emphasizes that data reliability is the first factor that needs to be analyzed and evaluated. Through a comprehensive analysis of the limitations of traditional methods and the development trend of techniques based on Artificial Intelligence (AI), a data processing method based on the Gaussian process regression method is proposed. Through experiments and result analysis, this method significantly improves data reliability, thereby enhancing the accuracy of avatar motion simulation in the virtual environment of the metaverse. Future research trends include further improvement of processing accuracy and speed; deploying on real devices; expanding the research into other factors contributing to unintended spatial distortions; exploring and applying appropriate processing techniques and technologies to enhance simulation reliability in virtual metaverse environments.

Keywords: spatial distortions; simulation; digital reconstruction; metaverse; Gaussian process regression



Academic Editor: Saïd Mahmoudi

Received: 10 November 2025

Revised: 26 November 2025

Accepted: 7 December 2025

Published: 11 December 2025

Citation: Huong, M.C.; Volkov, A.; Muthanna, A.; Koucheryavy, A.; Kozyrev, D.; Sztrik, J. Improving Avatar Accuracy with Gaussian Process Regression Method in Mirror Metaverses. *Information* **2025**, *16*, 1099. <https://doi.org/10.3390/info16121099>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, the “Metaverse” has become a rapidly emerging field of technology and a topic of great interest within the scientific community, due to its vast development potential and significant benefits. The emergence and extremely strong development of a series of new technologies, such as AI, blockchain, Virtual Reality (VR), Augmented Reality (AR), Internet of Things (IoT) systems, have provided the means to research, develop, and perfect the metaverse. Step by step, the metaverse is being integrated into various aspects of life, giving us new ways to communicate with the environment and life around us, as well as technological solutions to solve difficult problems.

The concept of “Metaverse” was first mentioned in the novel “Snow Crash” by writer Neal Stephenson, with the meaning of a virtual space where people can interact with each

other through avatars [1]. Nowadays, from their perspective, scientific researchers have given many different definitions of the metaverse. To summarize, Metaverse is an expanded virtual space where reality and virtuality coexist; where people can participate in many activities such as communication, connection, learning, working, shopping, exchanging, business, entertainment, etc.; where users connect through digital avatars, which recreate the user's true self in the real world; and where individuals can move and interact freely within the metaverse.

The term "virtual" in the traditional sense means something that does not physically exist but is represented by software and therefore is not real. We can see it, but we cannot touch, hold, or interact with it directly. However, in today's computer technology era, this concept does not fully reflect the term "virtual". "Virtual" in the field of information technology and telecommunications also refers to things created from the digital environment, simulating "real" objects or functions, and can operate effectively like their "real" versions. For example, virtual currency, virtual servers, virtual memory, virtual meeting rooms, etc. Reconstruction and simulation are two main aspects that are of particular interest to developers, researchers, and scientists in the field of "virtualization". They are related but have distinct differences. Simulations are virtual versions of real-world entities, which have cognitive or functional similarities to real-world entities, but do not have the same practical value or impact as their real-world counterparts. Also, virtual versions, generated by computers, but reconstruction results in copies of real-world entities that have (or nearly) the same practical value or impact as their real-world counterparts. Therefore, they not only exist in the virtual environment but also have meaning in the real world with an impact close to that of the physical entities they simulate [2]. From a certain perspective, the reconstruction of entities in the virtual world can be seen as a higher-level process compared to simulation. However, simulation plays an important role in predicting or analyzing events that may happen in the future. Therefore, in the fields of information technology and telecommunications, they are often researched and developed together. Metaverse is one of them.

Compared to individual fields such as simulation or reconstruction, creating a virtual space of the metaverse requires a much higher level of virtualization, which is much more rigorous and complex. The virtual world of the metaverse must be formed from a combination of different underlying technologies, in which reconstruction and simulation play a core role in giving users the feeling of "immersing themselves in another world" that actually exists alongside the real world. There are still many challenges and scientific problems to solve to build such metaverses. One of them is the optimization of synchronization accuracy between the real world and the virtual environment of the metaverse through simulation and digital reconstruction. The most serious and immediate consequence of poor synchronization is unintended spatial distortion, which breaks the user's sense of immersion. However, not all spatial distortions have negative effects. Eike Langbehn and Frank Steinicke, in their research, gave users the feeling of walking in a virtual space larger than the limited space in the real world [3]. Similarly, Satoshi Saga and Kotaro Sakae proposed pseudo-haptics that create virtual tactile sensations by manipulating the control-display (C/D) ratio between the user's movements and their visual feedback [4].

This research aims to provide an architecture for virtualizing user behavior in the metaverse; to present, analyze, and evaluate the main technical factors that lead to the unwanted spatial distortion phenomenon in the virtual world that we encountered during our experimental research. At the same time, a method to improve the reliability of data for the avatar motion simulation process based on the Gaussian process regression method is proposed.

2. Related Works

2.1. Simulation and Digital Reconstruction in the Virtual World

The simulation and digital reconstruction of objects from the real world to virtual environments, requiring high precision, has been of interest and research focus in recent years and has had impressive achievements at different scales of implementation. At the small scale, or the level of the object, the 1.8 m high statue of Donatello “Maddalena” was modeled in 3D by researchers G. Guidi, J. A. Beraldin, and C. Atzeni with a maximum vertical deviation of less than 0.5 mm [5]. On a larger scale, we can easily find them in role-playing video games or social networks. Second life is a typical example of 3D simulation models with extremely great detail. Especially at the level of large spaces, the Digital Twin (DT) is the most typical example. Building 3D City models or Virtual Cities is nothing strange. For example, Virtual London (UK), Digital Twin of Gothenburg (Sweden), Digital Twin of Helsinki (Finland), etc. According to the data published by the authors Dušan Jovanović, Stevan Milovanov, Igor Ruskovski, Miro Govedarica, Dubravka Sladić, Aleksandra Radulović, and Vladimir Pajić in their work [6], when building a 3D city model, the Root Mean Square Error (RMSE) index is determined to be less than 3 cm vertically and, less than 10 cm horizontally. In particular, the Virtual Singapore project is implemented on a national scale—an initiative to build a 3D digital twin model of Singapore, to support the development of a “Smart Nation” enhanced by 3D technology—the 3D modeling range is up to 5500 km along the road network with an error of 0.3 m [7]. This is considered an important milestone marking the expansion of the scale from the city model to the national model in the field of DT, in particular, and the field of reconstruction and simulation in general. However, this simulated object is often static.

2.2. Unintended Spatial Distortion

Unintended spatial distortion in virtual environments is a common issue when simulating objects for mobile applications such as VR, AR, or the metaverse. Analyzing the factors that lead to this phenomenon is a prerequisite for developing methods to minimize its impact on the user experience. Manuela Chessa and Fabio Solari show that distortions in the spatial layout of virtual scenes, specifically the underestimation of egocentric distance, arise from users viewing the world through an uncalibrated camera, leading to a distorted virtual view of space [8]. Matthew D. Ryan and Paul M. Sharkey demonstrate the influence of network latency on distortion in distributed virtual environments [9]. Zhenping Xia and co-authors focused on objectively quantifying dynamic spatial distortions in virtual environments to enhance realism and minimize visually induced motion sickness—a serious consequence of spatial distortion in virtual worlds [10]. Many factors lead to unintended spatial distortion in virtual worlds. One of the first factors to consider and address is the data provided for the simulation and digital reconstruction of the virtual world.

2.3. Data Processing Based on the Gaussian Process Regression Method

The application of AI to data processing is gradually becoming popular and brings superior results compared to traditional methods. Among them is the application of the Gaussian process regression method and its developed versions to data processing. It provides the ability to handle large data with complex nonlinearities. James Hensman, Nicolo Fusi, and Neil D. Lawrence focused on extending Gaussian Processes (GP) to handle large datasets (big data)—something that traditional GP has difficulty with due to high computational costs [11]. Ching-An Cheng and Byron Boots proposed an extended Gaussian Process (GP) regression method that is capable of incremental real-time updates and is computationally efficient using variational and representative points [12]. Additionally, Armin Lederer and colleagues concentrated on developing a real-time machine

learning method based on Gaussian Processes [13]. In summary, integrating machine learning technologies for data processing in real-time applications such as the Metaverse offers promising improvements in accuracy, stability, and performance. Further research in this area is to find a balance between processing speed and model accuracy for real-time data processing.

3. Virtualization Architecture of Human Movement in the Virtual World of the Metaverse, Spatial Distortion in the Virtual World of the Metaverse and the Factors That Lead to It

The rapid development of platform technologies provides a solid basis for the emergence and development of complete metaverses—vast 3D digital spaces where users can interact with each other, with virtual objects, and the surrounding environment through their digital avatars. Going beyond the limits of the 2D concept, the metaverse promises to bring a living world where work, entertainment, and social interaction are approached comprehensively. The core element that needs to be promoted in the metaverse is the creation of an “immersive” experience for users. Users do not just “look” at the virtual world; they need to feel that they are actually “present” in it. This presence relies heavily on the ability to accurately and instantly reflect every action and gesture of the user in the real world into the virtual environment through the avatar. From a wave, a nod, to complex steps, everything needs to be reproduced smoothly and naturally.

Therefore, coming up with standards and norms for building an architecture that can effectively simulate user movements has become one of the top technological challenges. Such architecture is responsible not only for collecting user movement data, processing it, and transmitting it over the network with the lowest latency, but also for ensuring that the avatar in the virtual world can accurately reproduce those movements.

Many organizations and communities in the field of telecommunications development around the world are developing standards to make “Metaverse” concepts a reality. Typical examples include: the International Telecommunication Union (ITU) Metaverse Focus Group 4 with standards FGMV-28 [14], FGMV-29 [15]; the Institute of Electrical and Electronics Engineers (IEEE) with standards IEEE 2888.3 [16], IEEE 2888.4 [17]; and others. In their research, Kyoungro Yoon, Sang-Kyun Kim, and their colleagues summarize the main content of the IEEE 2888 standard, which is designed to connect the physical world with virtual environments. They also visualized this standard with Draft Architecture for Virtual Reality Disaster Response Training System with Six Degrees of Freedom (6 DoF) [18].

Figure 1 shows the architecture of user behavior virtualization in the metaverse virtual environment according to the IEEE 2888.4 standard.

Sensors, cameras, touch gloves, or telepresence suits collect data about the user and the environment. Aggregated data from the motion tracking system is sent to the motion tracking server. The motion tracking server processes the data and sends analysis requests to the computing system (Data Center). Here, the analysis and processing system receives the input data, analyzes the movement, and sends commands for user actions. The situation assessment system analyzes the received data, issues commands in response to the user’s actions within the virtual environment, and sends them to the simulation and reconstruction system. The situation assessment system analyzes the received data, issues commands in response to the user’s actions from the virtual environment, and sends them to the modeling and reconstruction system. Here, the system predicts and simulates the user’s movements and the environment based on the collected data. Simulation models, integrated together and displayed in a “virtual” form in the HMD, simulate the feedback of the virtual object/environment with the user’s actions. Internet of Things (IoT) devices convey to users the sensation of feedback from virtual objects/environments, while simultaneously

collecting and transmitting data to the system for modeling assessment and calculation of the user's next actions.

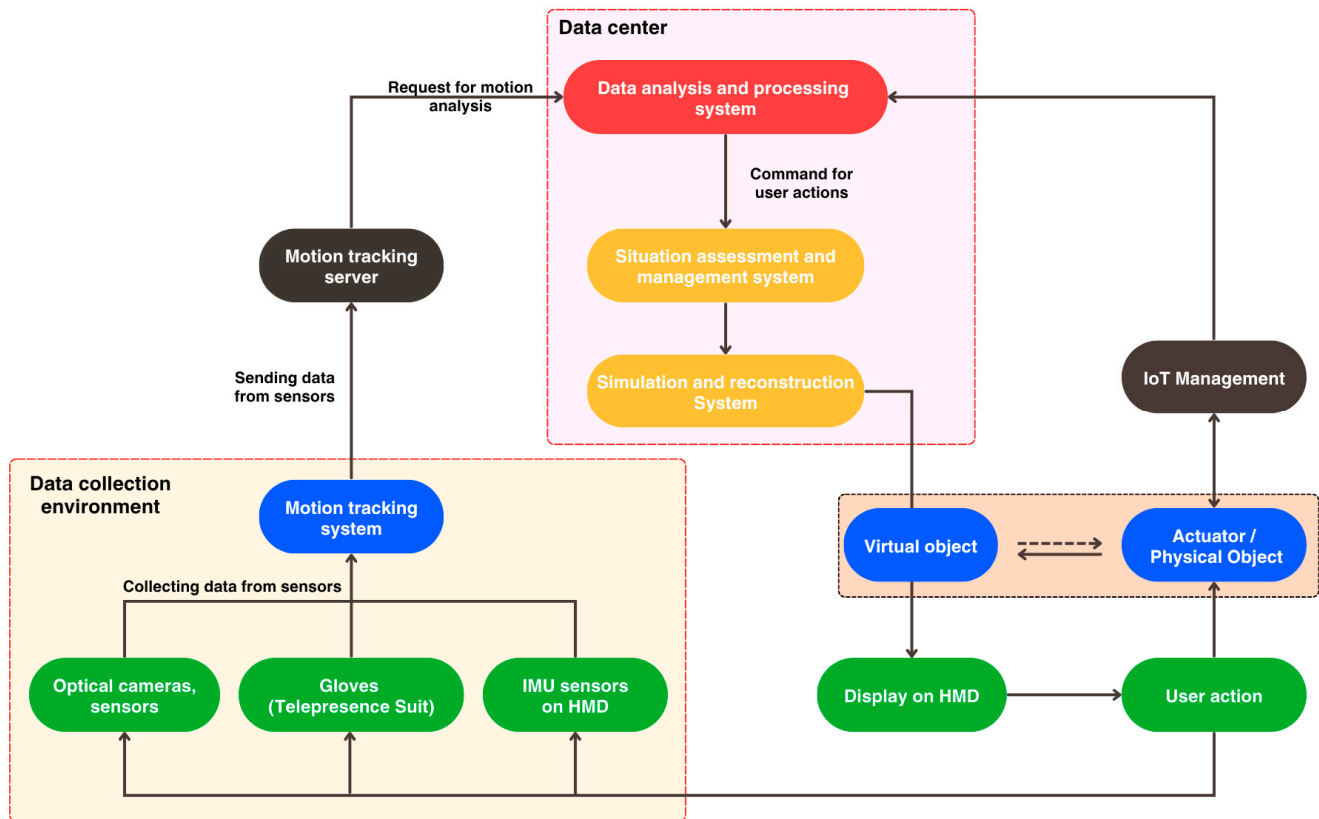


Figure 1. User behavior virtualization architecture for the metaverse according to IEEE 2888.4 standard.

The metaverse is not only a technological trend but also opens opportunities for breakthroughs in many fields. Typical applications include virtual classrooms or experimental modeling in education; psychotherapy, telemedicine in the medical field; virtual office, virtual store in commerce and business; and others.

Currently, one of our key research projects is being conducted in Mega Lab 6G of the Saint Petersburg State University of Telecommunications named after Prof. M.A. Bonch-Bruевич is aimed at applying the metaverse in the field of medicine. We were fortunate to meet and exchange experiences with leading doctors. They shared with us a method of treatment based on psychotherapy for patients who, after severe injuries, are almost unable to move or move. When a patient unfortunately suffers a serious injury to a limb or other body part, muscle rehabilitation is very important to avoid muscle atrophy and loss of mobility. Treatment is usually performed using massage in combination with physiotherapy. Recent medical research has shown that muscle stimulation through sensory memory in areas of the brain responsible for motor control or associated with sensation and pain has a more positive effect in the rehabilitation treatment of patients after severe injury [19].

With our efforts, we have obtained the first results of this research. However, there are difficulties that we encounter. The most typical is the virtualization of real-world objects into a virtual environment. The size and coordinates of the object, as well as the simulation of the user's actions affecting the environment, have relatively large errors compared to the real version. This phenomenon in the field of augmented and virtual reality is also being researched and discussed, and can be called unintended "unintended spatial distortion" in the virtual world [10]. This is a key factor in determining the feasibility of this research. The metaverse, especially in the medical field, should provide users with the most realistic

sensation, both visual, auditory, and tactile, like what they experience in the real world. Therefore, this issue needs to be studied and considered deeply and comprehensively.

There are many causes that can lead to this phenomenon, which are divided into two main groups: human factors and technical factors.

In the course of our research, we present technical factors that may lead to spatial distortion in virtual worlds within the Metaverse, which are systematized in Figure 2.

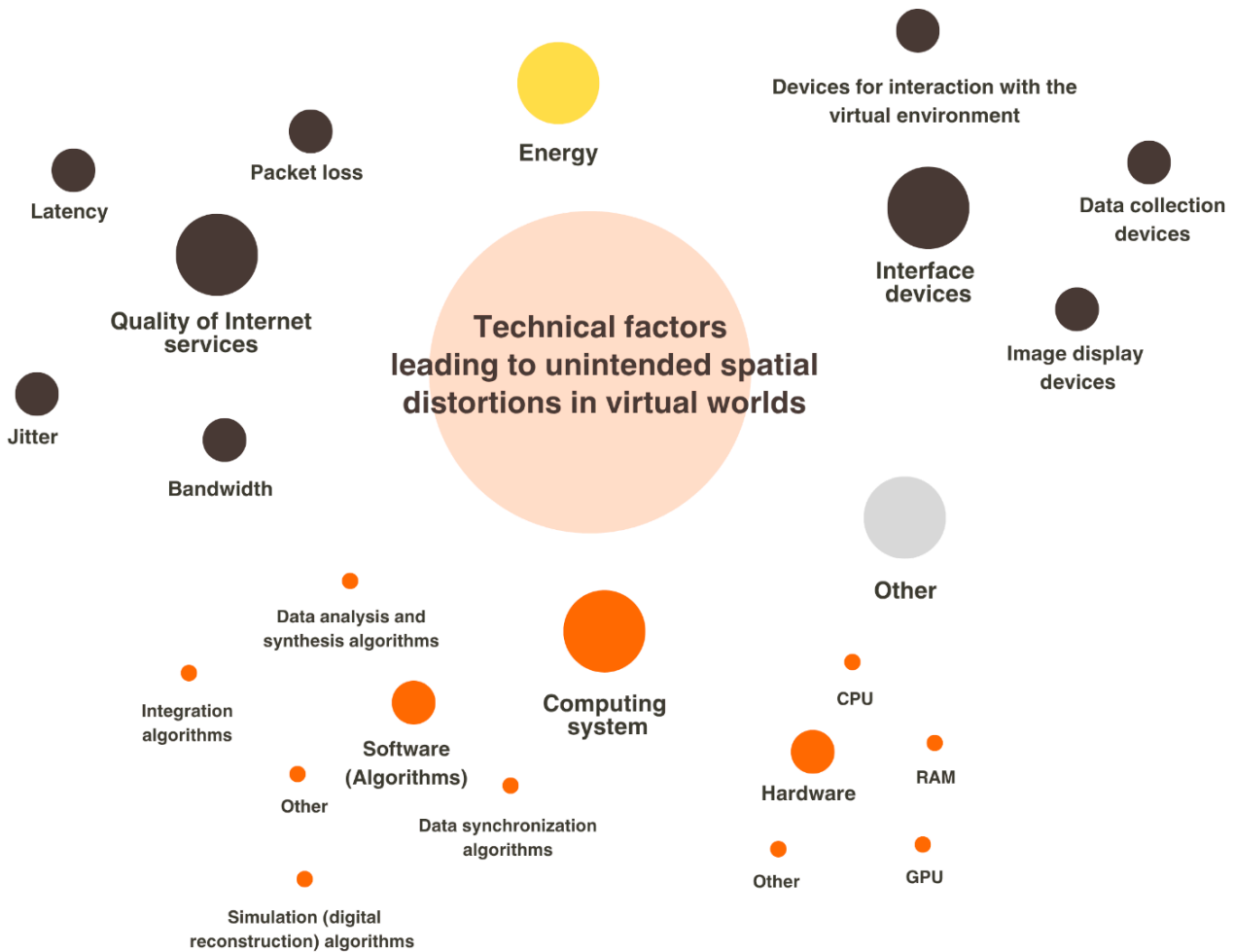


Figure 2. Technical factors leading to unintended spatial distortions in virtual worlds.

Energy

The energy factor does not directly lead to spatial distortion in the virtual world, but is an important fundamental factor that ensures a reliable infrastructure for the entire system, from data processing performance, images, to simulation models, etc. Lack of Energy for processing can lead to errors and omissions in data collection, processing, and operations in the virtual world, which in turn leads to errors in the virtualization process.

Quality of Internet services

Like energy, the Internet is an important component of the Metaverse infrastructure and a foundation for carrying out other functions. There are four aspects of Quality of Service (QoS) on the Internet that need to be considered: latency, bandwidth, jitter, and packet loss [20].

Like the Internet quality of service (QoS) requirements for virtual or augmented reality application domains, the metaverse requires an Internet with extremely low latency and high bandwidth. High latency will lead to a situation where data cannot be updated in a

timely manner, especially when simulating the movement of objects in the real world. This can lead to positioning errors in the virtual space, which may even cause motion sickness in users. Low bandwidth prevents data from being processed and transmitted on time, causing it to be rendered in the virtual environment as incomplete or simplified models, losing the necessary detail. Latency fluctuations cause instability in the transmission and rendering of graphics and other data of virtualized objects. Packet loss is an extremely important factor when considering the realism and high precision of virtualization in virtual environments. When data files are not transmitted or are lost during transmission, the object is not simulated, or parts of the object are not fully simulated in the virtual space.

In general, the quality of Internet service directly affects the stability, authenticity, and integrity of space and real-world objects that are modeled and reconstructed digitally in the virtual environment. The four elements of QoS on the Internet interact with each other and affect the quality of the virtual world. Therefore, they must be considered simultaneously and comprehensively.

Interface devices

There are three main groups of devices considered here: a group of data collection devices, a group of image display devices, and a group of devices that help users interact with the virtual environment.

Data acquisition devices include sensor systems, cameras, and more. As the name suggests, their role is to collect data according to the function of each type of device provided to the system, to analyze and process the information for simulation in a virtual environment.

Visual display devices include head-mounted displays (HMDs), virtual reality glasses, or projection systems. They help users and those around them observe what is happening in the virtual world. Visual display devices include head-mounted displays (HMDs), virtual reality glasses, or projection systems. They help users and those around them observe what is happening in the virtual world.

Devices that help interact with virtual environments include touch gloves, portable controllers, telepresence suits, and others. They allow users to interact and perform actions in a virtual environment. Based on the data transmitted from these devices, the system models the user's behavior in the virtual environment.

Currently, the quality of these optimized devices is extremely high. However, technical inaccuracies still exist, even if they are very minor. When using a metaverse system, many data collection devices are required. Therefore, if the permissible error of all these devices is not properly controlled, the collected error will have a clear difference compared to reality. This will lead to errors in the entire system, and the consequence will be spatial distortions in the virtual world.

Computing System

This is an extremely important fundamental component of the entire system. Two main aspects are considered: hardware and software (algorithms).

Hardware includes the central processing unit (CPU), graphics processing unit (GPU), random access memory (RAM), and others. When the amount of data being processed exceeds the hardware limit, the data is not fully updated in real time, which leads to distortion effects such as spatial scaling or frame stuttering.

Algorithms

The computing system processes data incorrectly or uses inefficient algorithms, such as simulating a collision between two spherical solid objects on a straight line in the real world, causing them to pass through each other or move in an unnatural manner instead of moving in the opposite direction of their original motion.

Solving the problem of virtualization in the metaverse requires the use of many different algorithms. These include data analysis and synthesis algorithms, data synchronization algorithms, simulation (digital reconstruction) algorithms, integration algorithms, and other algorithms.

Data analysis and synthesis algorithms help analyze and synthesize data transmitted from sensors, cameras, user devices, etc., in order to filter out noise, detect errors, and format the data in a way that is suitable for a computing system.

Data is collected from multiple device sources and distributed sensors. Each device has its own characteristics. Data synchronization algorithms help maintain consistency, reduce errors, prevent information conflicts, minimize latency, and optimize data transmission between system components.

Simulation (digital reconstruction) algorithms include user avatar simulation, environment simulation, object/person simulation, physics simulation algorithms, behavior simulation algorithms, and other algorithms. These algorithms are required to ensure high accuracy to create a realistic user experience and a sense of presence in the real world. However, this remains a challenge in metaverse environments.

Integrating simulation models and digital reconstruction into one virtual environment is also a serious challenge. The data from many different models is very large and complex; the models overlap, so the integration algorithms must provide very high accuracy.

These algorithms play an important role in solving each stage of the virtualization process from reality to the virtual world, as well as in managing the virtual environment. Incorrect processing of these algorithms will directly lead to spatial distortions in the virtual environment.

In summary, the development of modern technologies provides a foundational platform for the emergence and evolution of metaverses in the near future, where users not only “look” into the virtual world but must truly “feel” present within it. This presence relies heavily on the ability to accurately and instantly reflect every action and gesture of the user in the real world into the virtual environment through the avatar. To achieve that, one of the important issues that needs to be carefully considered is the unintended spatial distortion in the virtual world. There are many causes that can lead to this phenomenon, which are divided into two main groups: human factors and technical factors. In our research, we have identified technical and hardware-related factors that contribute to this phenomenon, including energy issues, internet QoS, device limitations, as well as processing and computing systems. Various approaches are being researched and implemented to overcome the above challenges. In particular, optimized algorithms are improved to enhance system performance, while AI is utilized to support device management, data processing, and more efficient computational operations. This is also the next research and development trend of the metaverse, as well as other related technologies.

4. Improving the Accuracy of Avatar Motion Based on the Gaussian Process Regression Method

4.1. The Importance of Data Filtering: Limitations of Traditional Methods

In today’s digital age, data plays an extremely important role in any technical system or application. In applications that interact with users between the real world and virtual environments, such as the metaverse or virtual reality, augmented reality, data is not just input information but the core foundation, directly determining the performance, realism, and appeal of any simulated experience. The level of detail and quality of data deeply affects every aspect, from the reconstruction of a physical environment to the way avatars interact with each other.

For the metaverse, data provided to the system is often collected primarily from the surrounding environment and the user through a multi-sensor system. However, these are raw data and often have limitations in reliability, such as errors, integrity, outliers, and many others. Therefore, data filtering is necessary before using it.

Advanced filtering methods can reduce position drift by more than 50% and reduce positioning errors by up to 50%, while multi-sensor data fusion improves positioning accuracy by up to 80% and improves map accuracy by 35% [21]. An improved Kalman filter has significantly reduced the Mean Squared Error (MSE), for example, by 87.5% for oxygen sensors [22]. Multi-sensor fusion frameworks demonstrate strong anti-interference capabilities, maintaining high Intersection over Union (IoU) reaching 0.852 with less than 5% degradation even in the presence of significant interference [23].

Traditional and basic filters, such as Kalman, are designed for linear [24] or near-linear systems with slowly varying parameters. However, most real-time systems, such as the metaverse, are nonlinear. This often leads to poor simulation adaptability in the case of complex nonlinear data. Because of these limitations, more modern methods, such as machine learning and artificial intelligence (AI) based filters, are increasingly being developed to be able to handle complex systems and better adapt to real-world conditions.

4.2. Real-Time Data Filtering Using Gaussian Process Regression

For continuous data, such as results from sensor measurements, in recent years, the Gaussian process regression method has emerged as a solution for models where the relationships between the data are complex and nonlinear. Gaussian Process Regression (GPR) is a non-parametric machine learning method based on probability theory that uses Gaussian processes to model data. Unlike many traditional regression methods that aim to estimate the parameters of a predefined function, GPR works by defining a probability distribution directly over the space of functions that can describe the observed data [25]. This approach has two main advantages: first, GPR can model complex and nonlinear relationships in data without requiring strict assumptions about the functional form; second, it provides not only predicted values but also a quantitative assessment of the uncertainty associated with each prediction. It is this capability that makes GPR such a powerful and flexible tool [26]. A Gaussian process is defined as a collection of random variables, any finite subset of which follows a Gaussian distribution. However, the biggest problem this method faces is the complexity of the algorithm and the memory requirements for large datasets. Therefore, it is not suitable for real-time processing tasks.

To overcome the limitations mentioned above, variations in this method are being developed and enhanced accordingly. In this research, the Sparse Variational Gaussian Process (SVGP) is used. SVGP uses a small set of inducing points to provide an approximate representation of the posterior distribution instead of working directly with the entire dataset. This allows for a significant reduction in computation time, resource consumption, and memory usage, while simultaneously speeding up the processing. The implementation of SVGP is shown in Figure 3.

The basic working principle of SVGP is to use variational methods to approximate a complex real posterior distribution $p(f|y)$ by computing a more easily manipulated variational distribution $q(f)$, which serves as an approximate distribution. Instead of calculating the exact posterior distribution, which is often difficult to determine, we seek the “best” approximation in a family of simpler distributions, which is defined as follows [27]:

$$q(f) = \int p(f|u)q(u)du, \quad (1)$$

where

- u —inducing points selected from the full range of values;
- $q(u)$ —variational distribution over the variables u ;
- $p(f|u)$ —conditional distribution of the latent function f at any point, given the function values at the inducing points u are known.

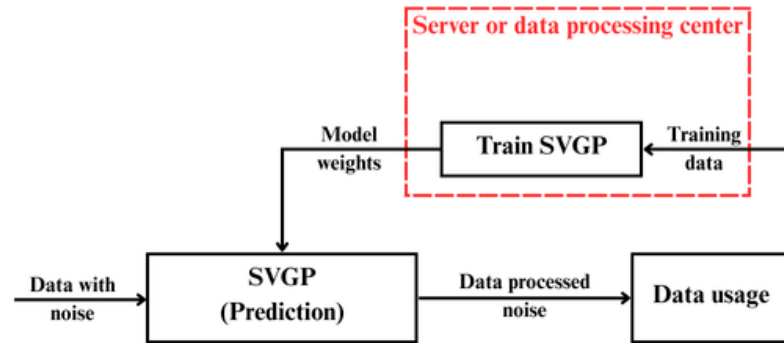


Figure 3. Description of the SVGP implementation process.

The variational distribution $q(u)$ is often chosen to be a simple and easy-to-use distribution, the most common of which is the multivariate Gaussian distribution:

$$q(u) = \mathcal{N}(u|m, S), \tag{2}$$

where

- m —expected vector (mean);
- S —covariance matrix.

Step 1: Train and save the SVGP model weights

This step is performed on a server or on computing systems with sufficiently powerful hardware.

For the model to “learn” the characteristics of the prior values, the following components of the SVGP model need to be defined [28]:

- Mean function $\mu(u)$ defines the prior mathematical expectation of the Gaussian process. It serves as the “baseline” for the model’s predictions. A simple and common choice for this function is a constant $\mu(u) = C$, where C is a constant optimized during the training process.
- The covariance function, or kernel function, defines the prior covariance structure of the Gaussian process, describes the degree of similarity between data points, and influences the shape of the modeled function. Various algorithms act as covariance functions. In this research, Scalekernel is used as a covariance function, which is defined as follows:

$$K_{scaled} = \theta_{scale} k_{Matern}. \tag{3}$$

$$k_{Matern} = \frac{2^{1-\nu}}{\Gamma(\nu)} (\sqrt{2\nu} \cdot d)^\nu K_\nu(\sqrt{2\nu} \cdot d), \tag{4}$$

$$d = (u_i - u_j)^T \Theta^{-2} (u_i - u_j), \tag{5}$$

where

- θ_{scale} —output scale parameter;
- k_{Matern} —Matern kernel matrix;
- ν —smoothness parameter;
- $\Gamma(\cdot)$ —gamma function;
- d —distance between two arbitrary induced points u_i and u_j ;

Θ —lengthscale parameter, which controls the range of influence between points;

$K_\nu(\cdot)$ —modified Bessel function of the second kind.

$K_\nu(z)$ —The modified Bessel function of the second kind is one of two linearly independent solutions of the modified Bessel differential equation:

$$z^2 \frac{d^2 y}{dz^2} + z \frac{dy}{dz} - (z^2 + \nu^2)y = 0. \tag{6}$$

- The Likelihood Function defines the relationship between the values of the latent function f and the observed labels y . It describes how noise is added to the values of the latent function to obtain the observed data. Gaussian Likelihood is the standard likelihood function and is defined as follows:

$$P(y|f) = f + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma^2), \tag{7}$$

where

ϵ —random noise sampled from a normal distribution;

σ^2 —noise variance parameter.

The main goal of training the SVGP model is to optimize the Evidence Lower Bound (ELBO)—the variational lower bound or loss function. It is important to note that optimizing the ELBO is mathematically equivalent to minimizing the Kullback–Leibler divergence between the approximate (predicted) posterior distribution and the actual posterior distribution. Many versions of ELBO have been successfully studied, including Variational ELBO [29] (Hensman et al., 2014), Predictive Log Likelihood [30] (Jankowiak et al., 2020), Gamma Robust Variational ELBO [31] (Knoblauch, 2019). They have advantages and disadvantages compared to each other. In this research, considering the nature of the data and possible technical limitations of the equipment, the Variational ELBO was used as the loss function, which is defined as follows:

$$\mathcal{L}_{ELBO} = \sum_{i=1}^N E_{q(f_i)} [\log p(y_i|f_i)] - \beta.KL[q(u)||p(u)], \tag{8}$$

where

N —the number of data points.

There are 2 main components in formula (8):

- The first component is the expected log-likelihood under the variational distribution $q(f)$, which measures how well the model fits the observed data:

$$\sum_{i=1}^N E_{q(f_i)} [\log p(y_i|f_i)]. \tag{9}$$

- The second component is the Kullback–Leibler (KL) divergence between the variational distribution $q(u)$ and the prior distribution $p(u)$:

$$\beta.KL[q(u)||p(u)], \tag{10}$$

where β —proportionality constant that reduces the adjustment effect of the KL divergence, $\beta = 1$ leads to the true variational ELBO.

This component acts as a regularization mechanism, encouraging $q(u)$ not to deviate too far from the prior distribution. This helps prevent overfitting and maintain the plausibility of the probabilistic model.

In the field of machine learning in general, algorithms developed for model optimization aim to minimize the ‘loss function’ by adjusting the model’s weight. The general mechanism of these algorithms operates by moving in the opposite direction of the increasing gradient (the derivative of the loss function). The gradient points in the direction of greatest increase, and going in the opposite direction means “going down the hill” to the point of minimum error.

Adam (Adaptive Moment Estimation) is one of the most popular, widely used, and effective adaptive learning rate optimization algorithms in machine learning models [32,33]. This optimization method is also used in this research. It combines ideas from Momentum and RMSProp to achieve fast and stable convergence. Momentum aims to accelerate the gradient in a consistent direction by adding a portion of the previous gradient to the current gradient. In other words, the gradient accumulates “momentum” over time, allowing the model to ignore ‘flat’ regions and reduce oscillations in areas with rapidly changing “slopes”. Meanwhile, RMSProp adjusts the learning rate for each individual parameter based on its variance. Parameters with large gradients receive smaller update steps, while parameters with small gradients receive larger ones.

The process of updating the parameters of the Adam optimizer model θ at time t includes the following steps:

- Gradient calculation:

$$g_t = \nabla_{\theta} \mathcal{L}(\theta_{t-1}) \quad (11)$$

- Gradient moving average update:

$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) g_t \quad (12)$$

- Update the moving average squared gradient:

$$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) g_t^2 \quad (13)$$

- Perform bias correction:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (14)$$

- Update the parameter:

$$\theta_t = \theta_{t-1} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \quad (15)$$

where

∇_{θ} —gradient operator with respect to θ ;

β_1 —decay rate for the first moment estimate;

β_2 —decay rate for the second moment estimate;

α —learning rate (step size);

ϵ —small constant to prevent division by zero when \hat{v}_t it is very small.

This process is performed continuously for all values of the model’s training data and repeated many times to find the optimal parameters. At each training cycle (epoch), the data is split into random mini-batches that are fed to the model. ELBO is computed for each mini-batch. Gradients are computed using the backward pass. Then, the model parameters are updated using the Adam optimizer.

Once the model has been trained, the best-performing weight sets are saved and passed to the SVGP, where they are used to process the data obtained from the potentiometers.

Step 2: Prediction (Use) with SVGP

In this step, the model can be deployed on edge data collection devices (without strict hardware requirements). Here, the SVGP model loads the set of weights transmitted from the server after training for further use.

To predict a new value x^* , SVGP computes the predictive distribution $q(f^*)$ [34]:

$$q(f^*) = \int p(f^*|u) q(u) du = \mathcal{N}(\mu^*, \sigma^{*2}) \tag{16}$$

In which:

$$\mu^* = K_{*u}K_{uu}^{-1}m \tag{17}$$

$$\sigma^{*2} = k(x^*, x^*) - K_{*u}K_{uu}^{-1}K_{u*} + K_{*u}K_{uu}^{-1}SK_{uu}^{-1}K_{u*} \tag{18}$$

where

- K_{*u} —the covariance vector between the test point x^* and the inducing points;
- K_{uu} —the covariance matrix of the inducing points;
- m —the mean vector of the variational distribution $q(u)$;
- $k(x^*, x^*)$ —the prior variance at the test point x^* ;
- $K_{u*} = K_{*u}^T$ —the transposed matrix of K_{*u} ;
- S —the covariance matrix of the variational distribution $q(u)$.

4.3. Simulate and Evaluate the Results Obtained by the SVGP Method with Traditional Methods

In this research, we use synthetic data. The dataset is simulated based on the real technical specifications of the Potentiometer 503 (Figure 4), taking into account the positional relationships between the potentiometers during the process of capturing the user’s motion parameters. From this ideal dataset, we further consider and compute various types of noise and device errors that occur during measurement, thereby obtaining a noise-contaminated dataset. The types of noise and errors considered include:

- Thermal noise: random noise caused by the thermal motion of electrons in the resistor.
- Contact noise [35]: caused by imperfect contact between the wiper and the resistive layer;
- Electromagnetic interference (EMI) [36]: environmental effects causing;
- Temperature drift: resistance changes due to ambient temperature variation.
- Hysteresis: caused by mechanical friction and material elasticity;
- Missing data: simulating communication failures or ADC read errors;
- Analog-to-digital converter noise [37];
- Burst errors and outliers: caused by strong impulse noise or temporary disconnections.

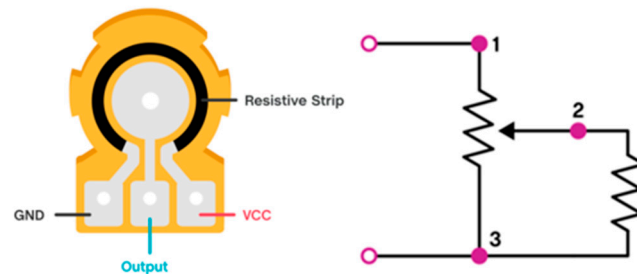


Figure 4. Potentiometer 503.

To collect the user’s motion data, four potentiometers are set up in the following positions:

- A potentiometer mounted on the lateral side of the shoulder, with its axis of rotation parallel to the body, partially attached to the torso and partially to the arm, measuring the shoulder elevation angle, set as the origin of the coordinate system;

- A potentiometer mounted on the upper part of the shoulder, with its axis of rotation parallel to the longitudinal axis of the humerus, measures the arm rotation angle.
- A potentiometer mounted on the elbow, the axis of rotation parallel to the axis of flexion of the elbow, at a distance of about 30 cm from the shoulder potentiometers.
- A potentiometer mounted on the wrist, the axis of rotation parallel to the axis of flexion/extension, at a distance of about 25 cm from the elbow potentiometer.

In this research, we used the 503 potentiometer model (WH148) manufactured by Chengdu Guosheng Technology Co., Ltd., Chengdu City, Sichuan Province, China. The main parameters of the 503 potentiometers are as follows:

- Input voltage: $V_{cc} = 5\text{ V}$;
- Total resistance: $R_{total} = 50\text{ k}\Omega$.

These potentiometers are configured to have the same time markers for data recording.

We simulate 60 different types of user movements, with approximately 1000 samples for each type. These movements are based on six main categories, including circle, Lissajous, figure eight, write S, write O, and pick and place (Figure 5). From each type of movement, one dataset is selected for training the model.

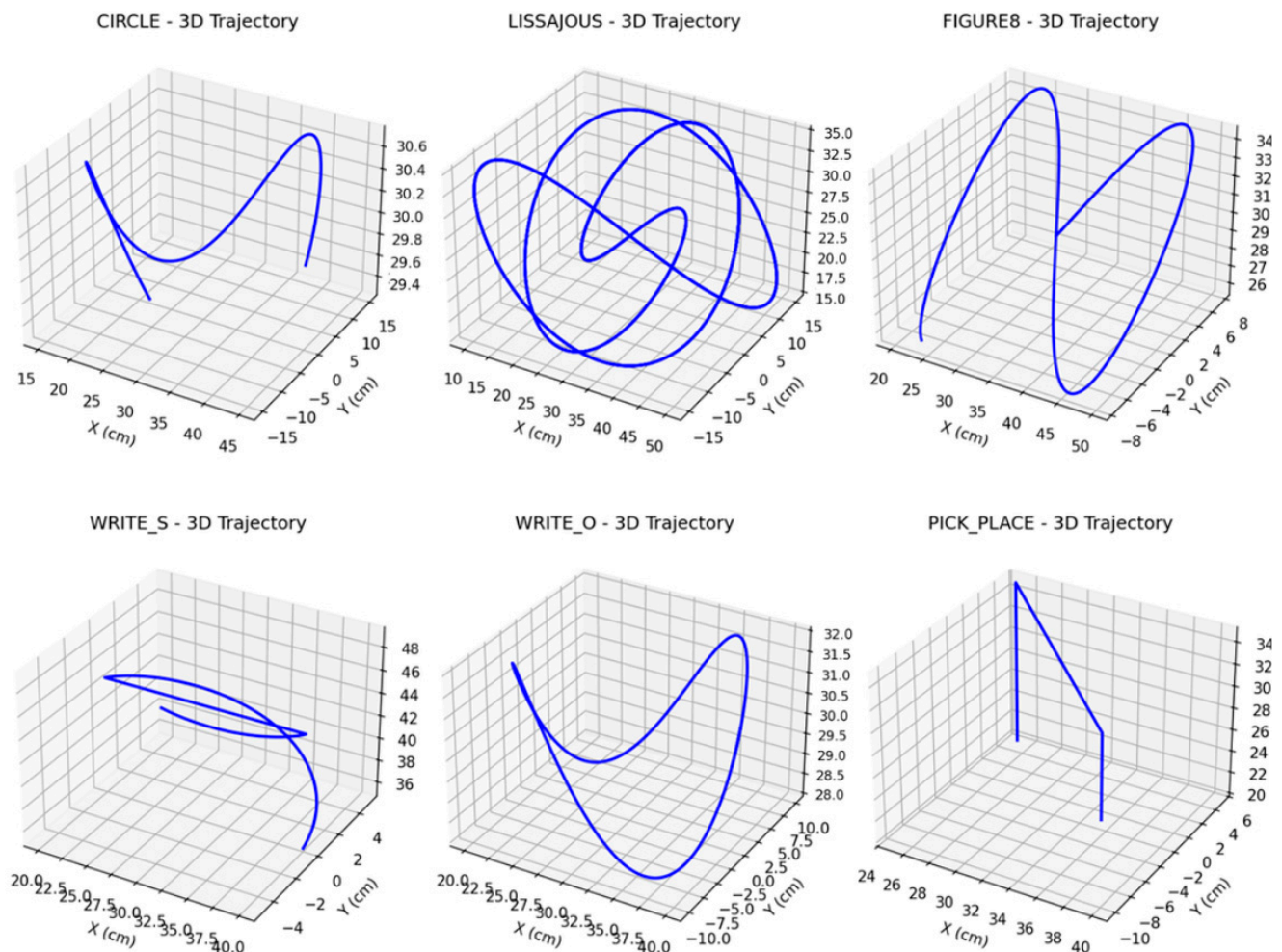


Figure 5. The types of user arm motion trajectories that are simulated.

The simulated data of the user's arm motion is collected by a potentiometer placed at the shoulder to measure the shoulder pitch angle, for the motion trajectory types shown in Figure 6. It can be seen that the collected data has complex nonlinearity, so traditional

methods such as the standard Kalman filter are not effective due to the requirement of data linearity.

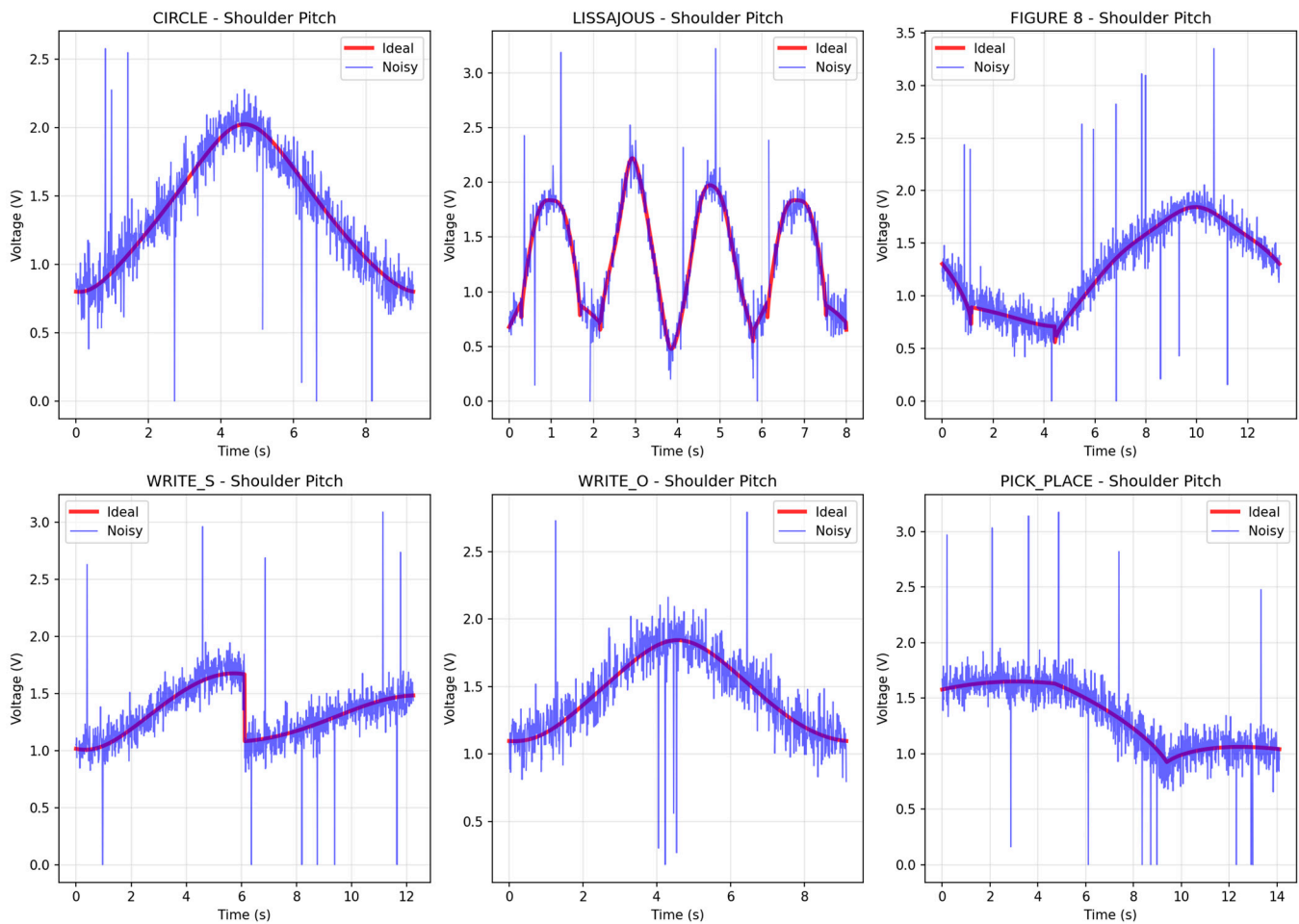


Figure 6. The output data of the 503 potentiometers for the six types of motion trajectories.

To train the SVGP model, we used hardware from an NVIDIA GeForce RTX 3050 laptop GPU and utilized the built-in Gpytorch library [28] to inherit and modify the model with the following parameters:

- Window sliding: 10 (helps the model learn the relationships between consecutive samples)
- Kernel hyperparameters: MaternKernel with smoothness parameter ($\nu = 2.5$) and Automatic Relevance Determination equal to the size of the window sliding
- Number of inducing points: 300 randomly selected; however, their positions are learned and automatically optimized during training to minimize the loss at each epoch
- Mini-batch size: 128
- Number of epochs: 150
- Optimizer parameters: Adam with a learning rate of 0.01

Evaluation of the deployability of SVGP on edge devices, where the hardware is not powerful enough. The pre-trained SVGP was executed on the device's CPU, and its filtering results were compared with two other filtering methods on the same type of device for nonlinear data, including: Unscented Kalman Filter (UKF) and an adaptive moving average filter, using data from the potentiometer placed at the shoulder to measure the shoulder pitch angle. The results for the motion trajectory "write-S" are shown in Figure 7. Table 1 presents the signal processing performance for different motion trajectories.

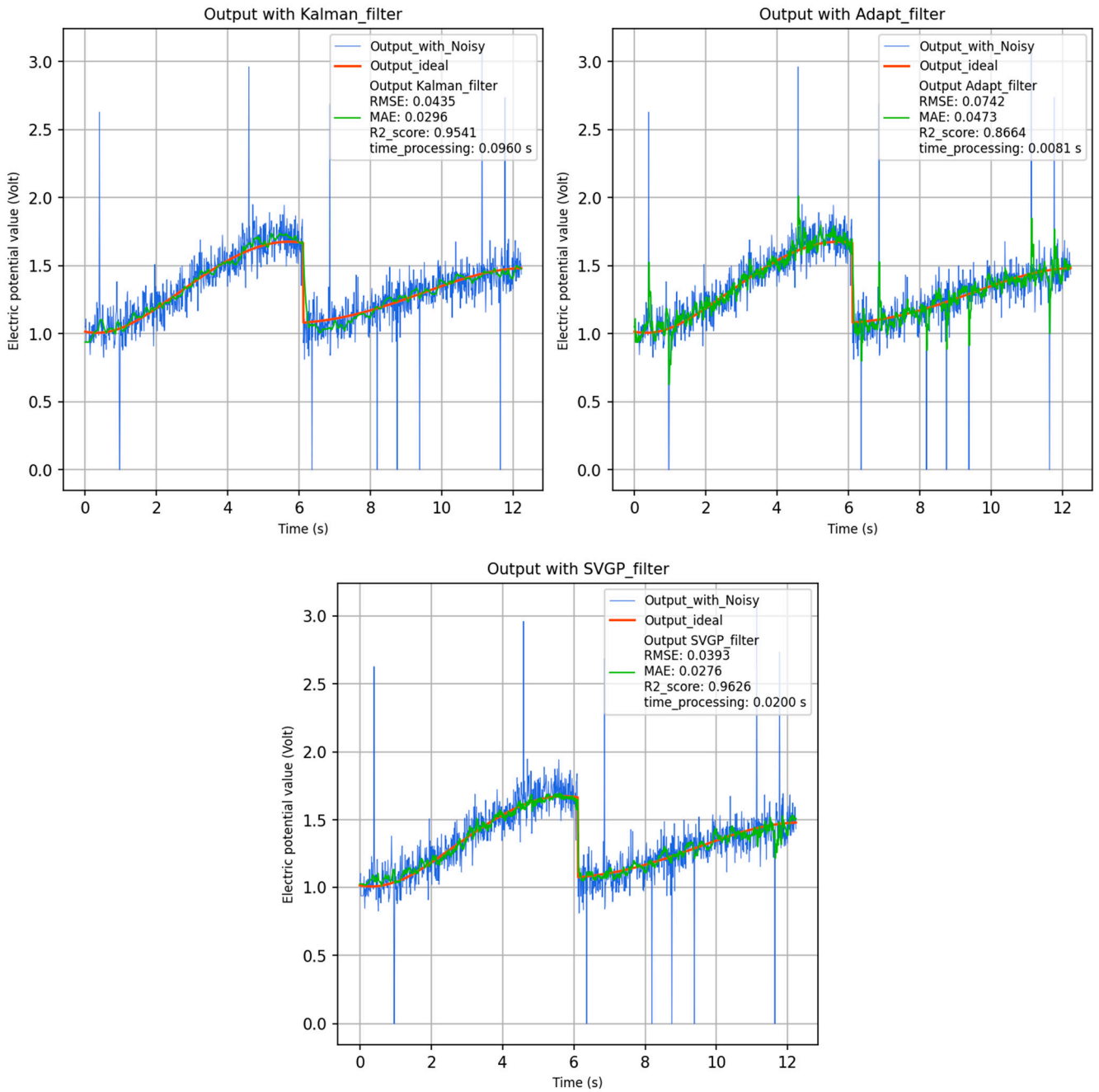


Figure 7. Filter performance on shoulder pitch angle data from the “write S” motion.

In addition to considering processing speed, the performance of the filters is evaluated based on the following three metrics, which are defined as follows:

- Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \tag{19}$$

- Mean Absolute Error (MAE):

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \tag{20}$$

– R-squared score (R^2):

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2} \tag{21}$$

where

y —data recorded by the potentiometer under ideal conditions;

\hat{y}_i —data after processing by the filter;

\bar{y} —average value of data recorded by the potentiometer under ideal conditions.

Table 1. Filter performance on simulated user motion trajectories.

	Circle				Write O			
	RMSE	MAE	R^2 score	Time (s)	RMSE	MAE	R^2 score	Time (s)
UKF	0.0406	0.0324	0.9902	0.105	0.0401	0.032	0.9762	0.0844
Adaptive filter	0.0782	0.0521	0.9635	0.013	0.0717	0.0496	0.9239	0.005
SVGP	0.0571	0.0385	0.984	0.026	0.0426	0.0343	0.9732	0.018
	Lissajous				Figure 8			
	RMSE	MAE	R^2 score	Time (s)	RMSE	MAE	R^2 score	Time (s)
UKF	0.114	0.097	0.9463	0.0684	0.0358	0.0289	0.9919	0.1105
Adaptive filter	0.1003	0.0665	0.9584	0.0064	0.0707	0.0459	0.9683	0.009
SVGP	0.0919	0.0635	0.9652	0.02	0.0319	0.0244	0.9935	0.0281
	Write S				Pick place			
	RMSE	MAE	R^2 score	Time (s)	RMSE	MAE	R^2 score	Time (s)
UKF	0.0435	0.0296	0.9541	0.096	0.0794	0.0337	0.9136	0.1357
Adaptive filter	0.0742	0.0473	0.8664	0.0081	0.1089	0.0547	0.8376	0.0112
SVGP	0.0393	0.0276	0.9626	0.02	0.0359	0.0277	0.9824	0.0234

It should be noted that the time values reported in the table represent the total processing time for all samples (excluding data transmission between devices and the handling of outliers). Therefore, the actual processing time per sample can be calculated by dividing the table values by approximately 1000, which is the number of samples per trajectory type.

From the collected data, it can be observed that the fastest data processing is achieved by the adaptive filter. For relatively simple or moderately complex motions such as Circle or Write S, the UKF demonstrates the highest performance, although the difference compared to SVGP is negligible; however, its processing time is 4–5 times longer. For more complex motions, such as Lissajous, Figure 8, Write S, and Pick and Place, SVGP demonstrates significantly better processing performance compared to the other two methods, while its processing speed shows only a slight difference relative to the adaptive filter. This indicates that, in simulated motion scenarios, SVGP provides a balanced trade-off between performance and data processing speed.

Compared to methods such as Cubature Kalman filters (CKF) or Particle Filter (PF). CKF is considered to perform better than UKF in many cases [38,39]. Similarly to UKF, it provides effective signal processing for motions ranging from simple to moderately complex, with faster speed. However, a common limitation of Kalman filtering methods is that for highly nonlinear and complex motions, their processing capability is restricted. This limitation is better addressed by PF, but the main drawback of PF is its algorithmic complexity, which results in significantly slower processing speed compared to Kalman-filtering methods [40].

Furthermore, when evaluating the feasibility of SVGP compared to other processing methods, in addition to performance and processing speed, another critical factor to consider is the cost of deployment and operation. For the goal of building a system capable of processing data directly on edge devices to reduce the load on central systems, such as in the Metaverse, the algorithm must strike a balance among all three factors. Other modern AI-based methods, such as Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN) multi-frame fusion filtering, require training datasets with tens of thousands of samples and a large number of hyperparameters. In contrast, SVGP only requires around 1000 to a few thousand samples for training and a small number of hyperparameters to learn the data characteristics. This results in significant differences in training time, model size, memory requirements, and computation speed, making SVGP more suitable for deployment on edge devices.

In summary, the SVGP-based filtering method provides a balanced signal processing solution, optimizing processing speed, performance, and operational and deployment costs. It has the potential to become a promising approach for data processing on edge devices with limited hardware resources in real-time systems, such as the Metaverse. The main challenges of this method lie in the following two aspects:

- Determining the number of inducing points: if the number of inducing points is too small, the model may underfit; if the number is too large, it increases memory usage and computation time, which slows down data processing speed.
- Ensuring training data: In practice, it is very difficult to provide the model with highly accurate data for training. Deploying high-precision measurement devices is complex and costly. In addition, a diverse range of data covering different motion trajectories is required. However, once the model is successfully trained, it can significantly reduce deployment costs and be applied on a large scale. This is particularly useful for wide-area distributed systems such as the Metaverse.

Using the data filtered by the SVGP method for the rendering process of the data collected by the four aforementioned potentiometers and comparing it with unfiltered data or data processed by an inefficient method, the results are shown in Figure 8.

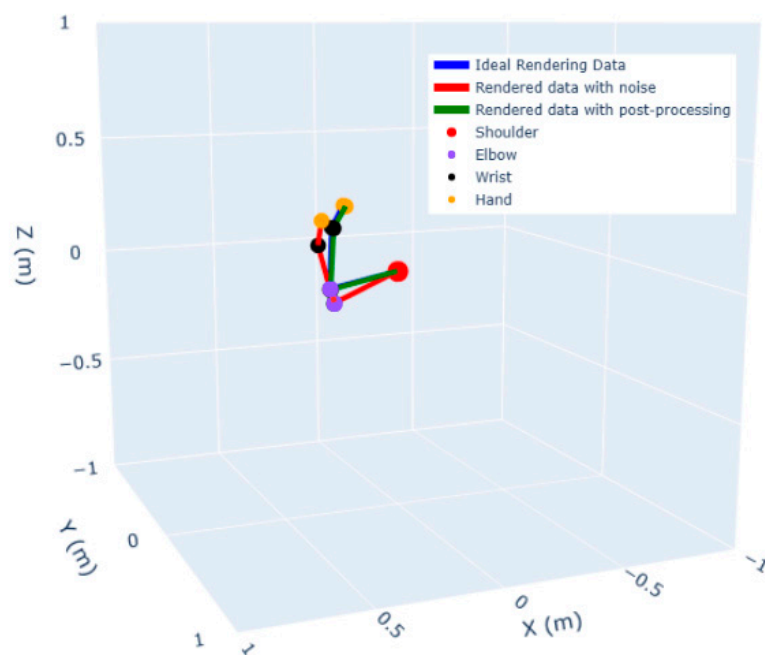


Figure 8. Evaluation of data rendering quality from the potentiometers.

From the results obtained, it is easy to see that the data filtered by the SVGP method brings high accuracy, and insignificant deviation compared to data collected under ideal conditions. Conversely, data that is not filtered for noise or filtered using methods that require linearity introduces significant errors.

In summary, human motion data collected from sensors in real-time systems such as the metaverse are often nonlinear in nature. Meanwhile, traditional filters, e.g., Kalman, are designed for linear or near-linear systems with slowly varying parameters. This makes traditional filters ineffective, leading to significant errors in motion reconstruction. Applying AI to data processing provides effective methods to deal with this problem. For continuous data such as sensor measurements, SVGP is a powerful, efficient, and suitable solution for the noise filtering problem in nonlinear motion tracking systems.

5. Conclusions and Future Research Directions

In this research, we presented our assessments regarding the phenomenon of spatial distortion in virtual environments. However, not all spatial distortions lead to negative consequences. Developers sometimes intentionally create spatial distortions to build virtual spaces tailored to their own requirements. In contrast, unintended spatial distortion is a common and challenging issue for applications interacting with virtual spaces. It leads to serious consequences for user experience, including visual motion sickness and incorrect assessment of object coordinates and dimensions in the virtual environment. Many factors lead to this phenomenon, which are divided into two main groups: human-related factors and technical factors. On the technical side, our research identifies factors that lead to this phenomenon, including energy, Internet quality, interface devices, device hardware limitations, data processing, and usage algorithms.

Data is an extremely important factor. In applications that interact with users between the real world and virtual environments, such as the metaverse or virtual reality, augmented reality, data is not just input information but the core foundation, directly determining the performance, realism, and appeal of any simulated experience. Therefore, algorithms for processing data recorded from sensors must ensure accuracy and reliability. However, traditional noise filters for data processing are often designed for linear or near-linear systems with slowly varying parameters. In reality, most real-time systems, such as those in the metaverse, are nonlinear. This often leads to poor adaptability or significant errors when using such data for simulation. Integrating AI into data filtering provides an effective solution to this problem.

In this research, to evaluate the efficiency of data filtering, we presented a method of implementing a filter based on the Gaussian process regression method, specifically SVGP. The obtained results are compared with the results of two other filtering methods, including: a variant of the Kalman filter suitable for complex nonlinear data—UKF, and an adaptive filter. The results show that the filter based on the Gaussian process regression method is a powerful, efficient, and suitable solution for the noise filtering problem in nonlinear motion tracking systems. Compared to other filters, SVGP achieves a good balance among three factors: processing speed, performance, and operational and deployment costs. It has the potential to become a promising approach for data processing on edge devices with limited hardware resources in real-time systems, such as the Metaverse. The main challenges of this method lie in the following two aspects: determining the number of inducing points and ensuring training data.

In the future, the research development trend will include improving the accuracy and speed of data processing; collecting real-world data and training the SVGP model; and deploying the model on devices for real-time applications. Additionally, the research will be expanded to explore in greater depth other factors contributing to unintended spatial

distortion as mentioned above. At the same time, explore and apply appropriate processing techniques and technologies to enhance simulation reliability in the virtual environment of the metaverse.

Author Contributions: Conceptualization, M.C.H., A.K. and A.M.; methodology, M.C.H., A.K. and A.M.; formal analysis, M.C.H., A.V., A.M. and J.S.; investigation, M.C.H., A.V., A.K. and A.M.; writing—original draft preparation, M.C.H., A.V. and A.M.; writing—review and editing, M.C.H., A.V., A.M., D.K. and J.S.; supervision, A.M. and D.K.; project administration, A.M. and A.K.; funding acquisition, A.M., D.K. and J.S. All authors have read and agreed to the published version of the manuscript.

Funding: Supported by the University of Debrecen Program for Scientific Publication.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, and further inquiries can be directed to the corresponding authors.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Stephenson, N. *Snow Crash*; Bantam Books: New York, NY, USA, 1992.
- Brey, P.A.E. Virtual Reality and Computer Simulation. In *The Handbook of Information and Computer Ethics*; Himma, K.E., Tavani, H.T., Eds.; John Wiley & Sons: Hoboken, NJ, USA, 2008; pp. 361–384. [CrossRef]
- Langbehn, E.; Steinicke, F. Redirected Walking in Virtual Reality. In *Encyclopedia of Computer Graphics and Games*; Lee, N., Ed.; Springer: Cham, Switzerland, 2018; pp. 1–11. [CrossRef]
- Saga, S.; Sakae, K. Sensory Perception During Partial Pseudo-Haptics Applied to Adjacent Fingers. *Multimodal Technol. Interact.* **2025**, *9*, 19. [CrossRef]
- Guidi, G.; Beraldin, J.-A.; Atzeni, C. High-accuracy 3D modeling of cultural heritage: The digitizing of Donatello's "Maddalena". *IEEE Trans. Image Process.* **2004**, *13*, 370–380. [CrossRef] [PubMed]
- Jovanović, D.; Milovanov, S.; Ruskovski, I.; Govedarica, M.; Sladić, D.; Radulovic, A.; Pajić, V. Building a Virtual 3D City Model for Smart Cities Applications: A Case Study on Campus Area of the University of Novi Sad. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 476. [CrossRef]
- Geospatial World. Virtual Singapore—Building a 3D-Empowered Smart Nation. Available online: <https://geospatialworld.net/prime/case-study/national-mapping/virtual-singapore-building-a-3d-empowered-smart-nation/> (accessed on 20 October 2025).
- Chessa, M.; Solari, F. A Geometric Model of Spatial Distortions in Virtual and Augmented Environments. In Proceedings of the 2018 IEEE International Conference on Image Processing, Applications and Systems (IPAS), Nice, France, 12–14 December 2018. [CrossRef]
- Ryan, M.D.; Sharkey, P.M. Distortion in Distributed Virtual Environments. In *Virtual Worlds*; Heudin, J.C., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 1998; Volume 1434, pp. 42–48. [CrossRef]
- Xia, Z.; Han, Q.; Zhang, Y.; Zhang, Y.; Hu, F. Objective Quantification of Dynamic Spatial Distortions for Enhanced Realism in Virtual Environments. *Virtual Real.* **2025**, *29*, 30. [CrossRef]
- Hensman, J.; Fusi, N.; Lawrence, N.D. Gaussian Processes for Big Data. *arXiv* **2013**, arXiv:1309.6835. [CrossRef]
- Cheng, C.-A.; Boots, B. Incremental Variational Sparse Gaussian Process Regression. NIPS, December 2016. Available online: https://papers.nips.cc/paper_files/paper/2016/hash/2596a54cdbb555cfd09cd5d991da0f55-Abstract.html (accessed on 20 October 2025).
- Lederer, A.; Ordóñez Conejo, A.J.; Maier, K.A.; Xiao, W.; Umlauf, J.; Hirche, S. Gaussian Process-Based Real-Time Learning for Safety Critical Applications. In Proceedings of the 38th International Conference on Machine Learning, Online, 18–24 July 2021; PMLR 139:6055–6064. Available online: <https://proceedings.mlr.press/v139/lederer21a.html> (accessed on 20 October 2025).
- FGMV-28; Requirements for the Metaverse Based on Digital Twins Enabling Integration of Virtual and Physical Worlds. In Proceedings of the 5th meeting of ITU FG-MV on metaverse, Queretaro, Mexico, 5–8 March 2024. Available online: <https://www.itu.int/en/ITU-T/focusgroups/mv/Documents/List%20of%20FG-MV%20deliverables/FGMV-28.pdf> (accessed on 20 October 2025).

15. FGMV-29; Reference Model for the Metaverse Based on a Digital Twin Enabling Integration of Virtual and Physical Worlds. In Proceedings of the 5th Meeting of ITU FG-MV on Metaverse, Queretaro, Mexico, 5–8 March 2024. Available online: <https://www.itu.int/en/ITU-T/focusgroups/mv/Documents/List%20of%20FG-MV%20deliverables/FGMV-29.pdf> (accessed on 20 October 2025).
16. 2888.3-2024; IEEE Standard for Orchestration of Digital Synchronization Between Cyber and Physical Worlds. IEEE: Piscataway, NJ, USA, 2025.
17. 2888.4-2023; IEEE Standard for Architecture for Virtual Reality Disaster Response Training System with Six Degrees of Freedom (6 DoF). IEEE: Piscataway, NJ, USA, 2024.
18. Yoon, K.; Kim, S.-K.; Choi, J.-H.; Jeong, S.P. Interfacing Cyber and Physical Worlds: Introduction to IEEE 2888 Standards. In Proceedings of the 2021 IEEE International Conference on Intelligent Reality (ICIR), Piscataway, NJ, USA, 12–13 May 2021. [[CrossRef](#)]
19. Perrey, S. A New Way to Treat Central Nervous System Dysfunction Caused by Musculoskeletal Injuries Using Transcranial Direct Current Stimulation: A Narrative Review. *Brain Sci.* **2025**, *15*, 101. [[CrossRef](#)] [[PubMed](#)]
20. Hossain, M.F.; Jamalipour, A.; Munasinghe, K. A Survey on Virtual Reality over Wireless Networks: Fundamentals, QoE, Enabling Technologies, Research Trends and Open Issues. *TechRxiv* **2023**, techRxiv:24585387.v1.
21. Vasile Crudu & MoldStud Research Team. The Role of Sensor Fusion in Enhancing SLAM for AR Applications. Available online: <https://moldstud.com/articles/p-the-role-of-sensor-fusion-in-enhancing-slam-for-ar-applications> (accessed on 20 October 2025).
22. Eom, K.H.; Lee, S.J.; Kyung, Y.S.; Lee, C.W.; Kim, M.C.; Jung, K.K. Improved Kalman Filter Method for Measurement Noise Reduction in Multi-Sensor RFID Systems. *Sensors* **2011**, *11*, 10266–10282. [[CrossRef](#)] [[PubMed](#)]
23. Zhou, Q.; Zuo, J.; Kang, W.; Ren, M. High-Precision 3D Reconstruction in Complex Scenes via Implicit Surface Reconstruction Enhanced by Multi-Sensor Data Fusion. *Sensors* **2025**, *25*, 2820. [[CrossRef](#)] [[PubMed](#)]
24. Julier, S.J.; Uhlmann, J.K. New Extension of the Kalman Filter to Nonlinear Systems. In Proceedings of the Signal Processing, Sensor Fusion, and Target Recognition VI, Orlando, FL, USA, 21–24 April 1997; Volume 3068. [[CrossRef](#)]
25. Wang, J. An Intuitive Tutorial to Gaussian Process Regression. *Comput. Sci. Eng.* **2023**, *25*, 4–11. [[CrossRef](#)]
26. Rasmussen, C.E.; Williams, C.K.I. *Gaussian Processes for Machine Learning*; The MIT Press: Cambridge, MA, USA, 2006; ISBN 026218253X. Available online: <https://gaussianprocess.org/gpml/chapters/RW.pdf> (accessed on 20 October 2025).
27. Cunningham, H.J.; de Souza, D.A.; Takao, S.; van der Wilk, M.; Deisenroth, M.P. Actually Sparse Variational Gaussian Processes. *arXiv* **2023**, arXiv:2304.05091. [[CrossRef](#)]
28. GPyTorch’s Documentation. Available online: <https://docs.gpytorch.ai/en/v1.8.1/> (accessed on 20 October 2025).
29. Hensman, J.; Matthews, A.G.G.; Ghahramani, Z. Scalable Variational Gaussian Process Classification. *arXiv* **2014**, arXiv:1411.2005. [[CrossRef](#)]
30. Jankowiak, M.; Pleiss, G.; Gardner, J.R. Parametric Gaussian Process Regressors. International Conference on Machine Learning. *arXiv* **2020**, arXiv:1910.07123. [[CrossRef](#)]
31. Knoblauch, J. Robust Deep Gaussian Processes. *arXiv* **2019**, arXiv:1904.02303. [[CrossRef](#)]
32. Agarwal, R. Complete Guide to the Adam Optimization Algorithm. Available online: <https://builtin.com/machine-learning/adam-optimization> (accessed on 20 October 2025).
33. Kingma, D.P.; Ba, J.L. Adam: A Method for Stochastic Optimization. *arXiv* **2015**, arXiv:1412.6980. [[CrossRef](#)]
34. Cortinovis, S.; Aitchison, L.; Hensman, J.; Eleftheriadis, S.; van der Wilk, M. Inverse-Free Sparse Variational Gaussian Processes. NeurIPS 2024 Workshop on Bayesian Decision-Making and Uncertainty. Available online: <https://openreview.net/pdf?id=nFHpVRvOgH> (accessed on 20 October 2025).
35. Fernández-Chimeno, M.; García González, M.Á.; Vargas-Drechsler, M.; Ramos Castro, J.J. Noise in Instrumentation. In *Wiley Encyclopedia of Biomedical Engineering*; Wiley: Hoboken, NJ, USA, 2006. [[CrossRef](#)]
36. Richelli, A. EMI Susceptibility Issue in Analog Front-End for Sensor Applications. *J. Sens.* **2015**, *2016*, 1082454. [[CrossRef](#)]
37. Lizon, B. Fundamentals of Precision ADC Noise Analysis. Texas Instruments, 2025. Available online: <https://www.ti.com/lit/eb/slyy192a/slyy192a.pdf?ts=1761033212461> (accessed on 20 October 2025).
38. Arasaratnam, I.; Haykin, S. Cubature Kalman Filters. *IEEE Trans. Autom. Control.* **2009**, *54*, 1254–1269. [[CrossRef](#)]
39. Liu, H.; Wang, X.; Zhang, Y.; Chen, Y.; Li, X.; Liu, S. Mode-Matching Universal Kalman Filter Based on Closed Skew-Normal Distribution. *Adv. Space Res.* **2025**, ahead of print. [[CrossRef](#)]
40. Stelzer, I.V.; Kager, J.; Herwig, C. Comparison of Particle Filter and Extended Kalman Filter Algorithms for Monitoring of Bioprocesses. In *Computer Aided Chemical Engineering*; Espuña, A., Graells, M., Puigjaner, L., Eds.; Elsevier: Amsterdam, The Netherlands, 2017; Volume 40, pp. 1483–1488. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.