An approach using multiple machine learning algorithms based on sensor modeling for gesture matching and recognition

Pinyu Li

Shenyang Feiyue Experimental High School, Shenyang, China

pinyu.li2025@outlook.com

Abstract. Gesture recognition is crucial for applications such as character modeling and humanoid robots, yet gesture recognition and generation remain underexplored, with most studies relying on camera-based tracking, which is limited by single-finger accuracy and computational requirements. This study proposes a hybrid framework that combines a haptic controller with machine learning (ML) to capture high-resolution finger motion data. The haptic controller enhances accuracy by providing additional evaluation of finger position and pressure metrics. We evaluated three ML algorithms - Random Forest (RF), Xgboost, and Support Vector Regression (SVR) to model 45 Euler angle-based joint rotations from 20 input parameters for each hand. Our results show that XGBoost outperforms RF and SVR in all sample sizes (500-3000 data points), achieving the lowest angle error (4431.4°) and distance error (19.8 cm) at 3000 samples. This study provides an innovative method for gesture recognition and provides valuable empirical experience for the application and development of related fields.

Keywords: Gesture recognition, Machine learning, Xgboost, Sensor.

1. Introduction

The generation of dynamic body movements plays an important role in character modeling and humanoid robots [1]. Accurate gesture recognition is the key to determining the efficiency of humanoid robots or the precision of animation modeling. Generally speaking, there are two methods for modeling hands. One is a physical method based on sensors, and the other is a data-driven method [2]. The physical method mainly simulates the control motion trajectory in the real world and models the strength of muscles and joints involved in each gesture through physical formulas. The data-driven method mainly uses a large number of examples of hand task movements and analyzes the dynamic graphs of human gesture activities to derive parameters and models. The advantages of physics-based methods are their high interpretability and versatility. Since this method simulates the muscle and joint dynamics in the real world, it can accurately reflect the mechanism of hand movement and is usually highly transferable. Once the physical model is established, it can be generalized between different tasks or users. However, the disadvantages of this type of method are high modeling and computational costs, complex parameter settings, and difficulty in handling individual differences or non-standard movements with high degrees of freedom[3]. In contrast, data-driven methods emphasize flexibility and learning ability [4]. Through training with a large amount of hand movement data, complex nonlinear movement patterns and individual characteristics can be automatically learned. The disadvantage of data-driven methods is that they are highly dependent on data quality and diversity. In the absence of sufficient data, they are prone to overfitting or insufficient generalization [5], and the results are difficult to interpret, and the predictability of precise control is poor.

With the development of artificial intelligence, many breakthroughs have been made in a large number of fields that require quantitative research [6, 7]. Among them, deep learning collects and analyzes a large amount of data and can autonomously perform nonlinear fitting and modeling [8]. In recent years, experts have actively studied the use of deep learning technology to explore character activity models. However, research and exploration on hand motion generation have been limited. Mainstream hand gesture detection and tracking are mainly based on image data provided by cameras.

Although the image data provided by the camera is good at capturing hand position and activity dynamic information in real time, camera-based tracking technology often has limitations in the precision of capturing the movement rules of a single finger. Although deep learning algorithms can alleviate this precision problem to a certain extent, the required computing resources have become another limitation. The tactile controller can solve this problem well. The tactile controller can unexpectedly provide another dimension of data about the position and movement rules of a single finger for the hand gesture map captured by the camera. This research proposes an innovative approach and highlights the effectiveness of multiple machine learning algorithms in gesture matching, providing valuable experience for related modeling applications.

2. Literature Review

2.1. Gesture Modeling Elements

A common method for gesture recognition is to determine the joint position of the hand by analyzing the three features of joint position or bone length, and joint angle [9]. The recognition method of joint position requires the formulation of a time-variable coordinate system. Most joint positions are described by simple Cartesian coordinates [10]. However, there are some limitations related to the human body's movement structure. For example, the joint position does not encode the direction information of the surrounding bones, but this coordinate system may cause changes in bone length, so additional constraints are usually required. Given that gestures are generated directly based on the input values of the tactile controller, this study uses Euler angle training to generate the parameters of the position of the finger joints. Euler angles are a general method for representing joint angles, which contain parameters of the position and direction of three different axes of the hand in a three-dimensional coordinate system [11].

2.1 Machine Learning Algorithms

Random forest (RF) is a classic algorithm for machine learning [12], as shown in Figure 1. Random forest is based on the characteristics of ensemble learning and decision trees. It builds multiple decision trees and combines their prediction results according to certain standards. The core idea is that the combination of multiple weak learners can form a strong learner. Its core feature is the sampling method of Bagging [13]. When training each tree, a random data subset with replacement sampling is used to build a sub-model. In addition, when each tree splits a node, only the features of the random subset are considered to avoid certain strong features dominating the model and improve the generalization ability [14].

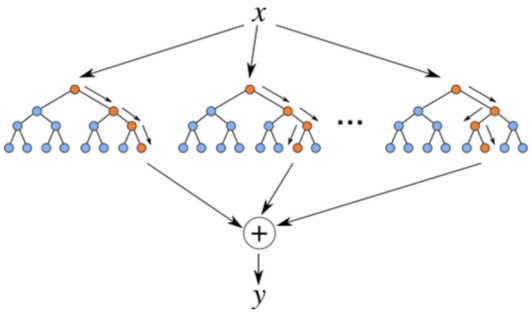

Figure 1. Random Forest Architecture

Xgboost (eXtreme Gradient Boosting) is also an ensemble learning algorithm based on gradient boosting decision trees [15]. Unlike RF, in each iteration, it learns the prediction residuals of the previous model, gradually superimposing rather than averaging weak classifiers to build a strong classifier, thereby effectively improving the overall performance of the model. It introduces L1 and L2 regularization terms in the model learning process to prevent overfitting. In addition, in terms of integration strategy, unlike Bagging used by random forests, XGBoost uses a Boosting architecture, emphasizing that each new tree makes targeted corrections to the errors of the previous model, and

introduces the second-order derivative to enhance the convergence speed of the loss function, thereby improving the accuracy of the model [16].

SVR (Support Vector Regression) is an extended form of support vector machine (SVM) in regression tasks [17]. Different from the traditional least squares regression method, the core idea of SVR is that within a given precision range, the deviation between the predicted value and the actual value will not be regarded as an error, and the loss will only be counted when the prediction deviates from this interval to reduce the sensitivity of outliers. SVR constructs an optimal hyperplane, projects the data into a high-dimensional feature space, and finds a regression function in this space that minimizes the structural risk.

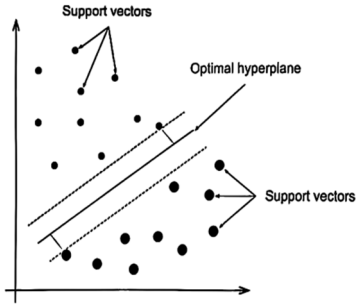

Figure 2. SVR Structure

3. Methodology

3.1 Haptic controller parameters

This experiment uses the input value of a haptic controller to match the modeling gesture. Specifically, this study introduces a special haptic controller as shown in Figure 3. The haptic controller is a rectangular instrument. The instrument includes five acoustic radars and five mechanical feedback buttons. Since each button is connected to a motor and a responsive radar. When pressed, the motor estimates the deformation of the finger based on the pressure intensity to determine the size of the output. When the finger is away from the button, the radar estimates the distance of the finger from the controller to estimate the position and shape of the finger. Therefore, this haptic controller can provide tactile and positional feedback for the finger when simulating virtual objects.

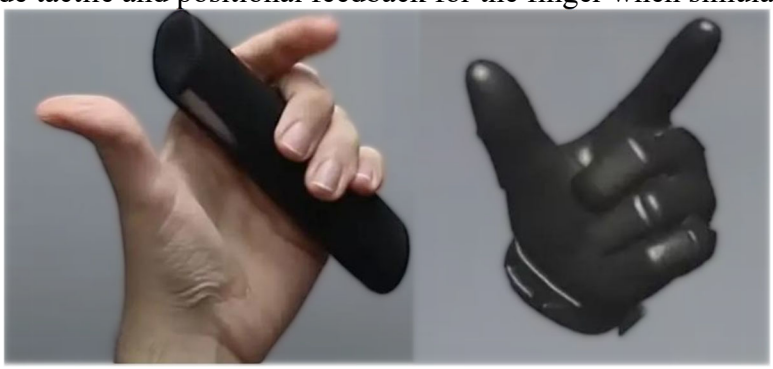

Figure 3. Schematic diagram of a tactile sensor

We first smoothly integrated the Unity game engine with these tactile controllers to facilitate the conversion of input values. We used motion capture technology to measure the process of users using these tactile controllers to perform hand movements. Each time the user operates, the relevant frames are automatically captured to generate corresponding data points. We used a rate of 60 frames per second to generate images and generated data sets of 500, 1000, 2000, and 3000 frames for stability verification. In the model. In the input signal, we use the pressure intensity of the radar as the input source, which includes four types of data: d, reflection intensity I, horizontal offset angle θ and pressure intensity P. There are 20 input parameters for the five fingers. In addition, since each finger has three joints, this study captures the information of the three joints of the five fingers, a total of 45

rotation dimensions based on Euler angles. The information of each joint is stored in the Cartesian coordinate system in the format of Euler angles (x, y, z). Therefore, each training data set contains 20 parameter values as input and 45 Euler angles as output.

This study uses RF, Xgboost and SVR as the trainers for this study the trainer. And used a training ratio of 8:2:1. In order to ensure the optimal performance of random forest, we used the Tree-structured Parzen Estimator (TPE) algorithm [18] to adjust the parameters of random forest. This is a Bayesian Optimization algorithm based on a sequence model for hyperparameter optimization. TPE uses a two-step method to iteratively optimize the objective function. First, the TPE algorithm builds a probability model. Kernel density estimation (KDE) is used to model the excellent parameter group and the poor parameter group, respectively, to form two probability distributions. The modeling formula is shown in Formula 1.$\sum_{i=1}^{n} K(x, x_i)$ is the kernel function，n is the number of samples.

$$p(x) = \frac{1}{n}\sum_{i=1}^{n} K(x, x_i) \tag{1}$$

Afterwards, the TPE algorithm samples from the distribution of new candidate good parameters and selects the parameters with the highest Expected Improvement value for evaluation. The parameters that maximize the EI criterion are selected as the next round of evaluation points. The expression of EI is shown in Formula 2, y^* and y^- represent the resulting values in the distribution of the two parameter groups.

$$EI(x) = \max(0, y^* - y) \cdot \frac{p(x \mid y^*)}{p(x \mid y^-)} \tag{2}$$

The final output of the random forest is the mean of all decision tree predictions:

$$\hat{\boldsymbol{y}} = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{h_t}(\boldsymbol{x}) \tag{3}$$

$\boldsymbol{x}$ is the 4-dimensional input feature, $\boldsymbol{h_t}$ is the prediction function of the t-th tree, and T is the total number of trees

3.2 Evaluation Method

This study evaluates the accuracy of hand motion matching by comparing the actual hand pose with the pose generated by RF. As shown in Figure 4, blue is the actual joint point, and red is the joint point matched by RF.

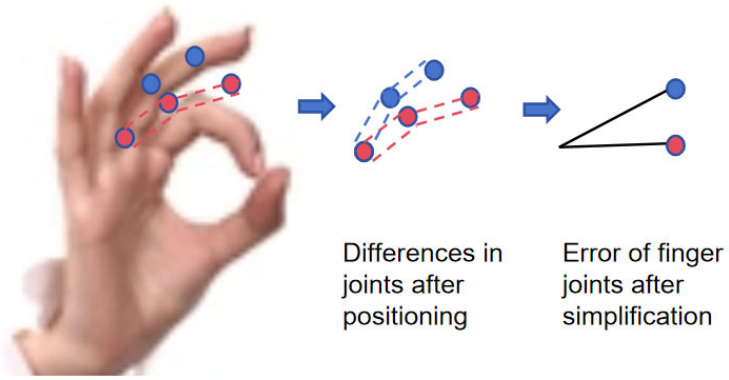

Figure 4. Example of gesture generation error

Therefore, it contains the angle error, that is, the direction deviation between the generated skeleton and the actual skeleton, and the distance error, that is, the position offset between the joints. Angular error (AE) is the average cumulative angle deviation of each finger joint in all frames, reflecting the error of finger bending or rotation:

$$AE = \frac{1}{N_1}\sum_{f=1}^{N_2}\sum_{j=1}^{N_1} \arccos\left(\frac{b_{1,j}^{f} \cdot \boldsymbol{b_{2,j}^{f}}}{|b_{1,j}^{f}|_2 |b_{2,j}^{f}|_2}\right) \tag{4}$$

$\boldsymbol{b_{1,j}^f}$ is the direction vector of the j-th bone in the original pose. $\boldsymbol{b_{2,j}^f}$ the direction vector of the j-th bone in the predicted pose. N1 represents the number of finger joints, while N2 represents the number of frames.

Position error (PE) represents the cumulative Euclidean distance error of each finger joint in all frames, reflecting the global position deviation of the generated pose:

$$PE = \frac{1}{N_1}\sum_{f=1}^{N_2}\sum_{j=1}^{N_1} |\, p_{1,j}^f - p_{2,j}^f|_2 \tag{5}$$

$\boldsymbol{p_{1,j}^f}$ represents the 3D coordinates of the j-th joint in the original pose in the f-th frame. $\boldsymbol{p_{2,j}^f}$ represents the 3D coordinates of the j-th joint in the predicted pose in the f-th frame.

4. Results

This study calculated the joint angle error and position error to measure the accuracy of the gestures generated by the three machine learning algorithms RF, XGBoost and SVR. Table 1 shows the error of the gestures generated using the training dataset used in this study. This study shows the results obtained using 500, 1000, 2000, and 3000 data samples. The unit of the angle error is degrees (°), and the unit of the position is centimeter (corresponding to the measurement value 0.1 Unity in the Unity game engine).

Table 1.Experimental Results

Model	AE	DE
SVR-500	9787.1	56.1
SVR-1000	6177.3	34.2
SVR-2000	5164.6	24.7
SVR-3000	4832.4	21.6
RF-500	10521.5	57.1
RF-1000	7280.1	34.8
RF-2000	6451.2	27.8
RF-3000	5915.2	23.7
Xgboost-500	9128.0	52.4
Xgboost-1000	5967.8	31.2
Xgboost-2000	4982.2	22.1
Xgboost-3000	4431.4	19.8

For the SVR algorithm, the angle error (AE) was 9787.1 with 500 samples, which is significantly better than RF (10521.5) but slightly worse than XGBoost (9128.0). This indicates that SVR is capable of capturing angular features relatively well in small-sample scenarios. As the sample size increased from 500 to 3000, AE decreased by 50.6%; however, the final AE value of 4832.4 remained higher than that of XGBoost (4431.4). In terms of distance error (DE), SVR demonstrated a clear linear modeling advantage: DE dropped from 56.1 to 21.6, a 61.5% reduction, outperforming RF (58.5%). This suggests that SVR is more effective in modeling the more linear nature of positional relationships. However, SVR showed signs of reaching a performance bottleneck with large datasets from 2000 to 3000 samples, DE improved by only 1.9, the smallest gain among all models, possibly indicating that SVR had approached its theoretical performance limit.

For the RF algorithm, AE was the worst among the three models at 500 samples (10521.5). Although AE improved by 43.8% with 3000 samples, the final AE value of 5915.2 was still significantly higher than that of XGBoost, reflecting the relatively low efficiency of the Bagging strategy in learning the nonlinear relationships in angle estimation. DE improved from 57.1 to 23.7 over the same sample range, with the largest improvement (4.1) observed between 2000 and 3000

samples. This suggests that the ensemble of trees partially captured some spatial geometric relationships, though still less effectively than the explicit optimization employed by XGBoost.

XGBoost consistently delivered the best performance in terms of AE across all sample sizes. With 3000 samples, its AE reached 4431.4 , which is 8.3% lower than SVR and 25.1% lower than RF. For DE, XGBoost also demonstrated superior robustness: DE decreased from 52.4 to 19.8, a 62.2% reduction. Notably, even in the small-sample setting (500 samples), XGBoost outperformed both SVR and RF significantly, suggesting that its loss function design (e.g., the use of Huber loss) offers greater resilience to positional deviations.

5. Discussion

The experimental results reveal the performance characteristics of SVR, RF, and XGBoost in gesture modeling, and each algorithm shows its own advantages and limitations. SVR performs well in linear modeling of small sample scenarios and positional relationships, but reaches a performance plateau when the data set is large. RF shows obvious data scale dependence and improves significantly with the increase of sample number, but it still lags behind XGBoost due to its low efficiency in handling nonlinear angle relationships through bagging. Our research found that among the algorithms for post-modeling machine learning, XGBoost is the most robust method. With its advanced gradient boosting framework, second-order optimization, and regularized loss function, it always maintains excellent accuracy at all sample sizes, making it particularly suitable for gesture matching and generation. Our research results also show that the frame rate selection and sample size of image training are crucial. Although the performance has not reached the best state after 3000 points, the actual decrease in its loss function is very limited at 3000 samples, and it is difficult to observe the difference with the naked eye.

One shortcoming of our research is that it only considers dynamic gesture recognition under joint alignment, without considering the integrity and variability of its activities. This requires us to model the time dimension more deeply. Future research will combine other modalities, such as time data to further improve performance. In addition, this article mainly studies modeling, but does not use deep learning, which may be more refined for image processing. In future research, we will continue to use deep learning for in-depth comparison and analysis.

6. Conclusion

This study proposed a hybrid gesture recognition framework that combines tactile controller data with machine learning. This study modeled the joints and positions of gestures, and innovatively calculated the accuracy of gestures by evaluating angle errors and position errors. By evaluating RF, XGBoost, and SVR, we demonstrated that XGBoost performed well in reducing both angle and position errors due to its gradient boosting architecture and regularization mechanism. SVR is effective for linear position modeling but has poor scalability; RF's ensemble method improves with the increase in data volume but lags in nonlinear feature extraction. The results highlight the importance of sample size, with diminishing returns beyond 3000 samples. Its limitations include the exclusion of temporal dynamics and reliance on joint-level alignment that lacks full motion context. Future research will combine time series analysis and deep learning to enhance modeling fidelity. This study advances the development of gesture recognition systems and provides valuable insights for applications such as robotics and animation where accuracy and adaptability are critical.

References

[1] Kim, S., Kim, C., You, B., Oh, S. (2009). "Stable whole-body motion generation for humanoid robots to imitate human motions," in 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, pp. 2518-2524.

[2] Wang, J., Li, Y., Gao, R. X., Zhang, F. (2022). "Hybrid physics-based and data-driven models for smart manufacturing: Modelling, simulation, and explainability," Journal of Manufacturing Systems, vol. 63, pp. 381-391.

[3] Patterson, T. A., Parton, A., Langrock, R., Blackwell, P. G., Thomas, L., King, R. (2017). "Statistical modelling of individual animal movement: an overview of key methods and a discussion of practical challenges," AStA Advances in Statistical Analysis, vol. 101, pp. 399-438.

[4] Xu, J., Wang, Y. (2025). "Enhancing healthcare recommendation systems with multimodal LLMs-based MOE architecture," in 5th International Conference on Signal Processing and Machine Learning (CONF SPML 2025), IET, vol. 2025, pp. 123-129.

[5] Luo, Y., Ye, Z., Lyu, R. (2024). "Detecting student depression on Weibo based on various multimodal fusion methods," in Fourth International Conference on Signal Processing and Machine Learning (CONF-SPML 2024), SPIE, vol. 13077, pp. 202-207.

[6] Luo, Y., Wang, Z. (2024). "Feature mining algorithm for student academic prediction based on interpretable deep neural network," in 2024 12th International Conference on Information and Education Technology (ICIET), IEEE, pp. 1-5.

[7] Xu, J., Wang, Y. (2025). "FMT: A Multimodal Pneumonia Detection Model Based on Stacking MOE Framework," arXiv preprint arXiv:2503.05626.

[8] Pang, P. C.-I., Chang, S., Verspoor, K., Pearce, J. (2016). "Designing health websites based on users' web-based information-seeking behaviors: A mixed-method observational study," Journal of Medical Internet Research, vol. 18, no. 6, p. e145.

[9] Ahad, M. A. R., Ahmed, M., Antar, A. D., Makihara, Y., Yagi, Y. (2021). "Action recognition using kinematics posture feature on 3D skeleton joint locations," Pattern Recognition Letters, vol. 145, pp. 216-224.

[10] Tulli, S. K. C. (2024). "Motion Planning and Robotics: Simplifying Real-World Challenges for Intelligent Systems," International Journal of Modern Computing, vol. 7, no. 1, pp. 57-71.

[11] Ancillao, A. (2022). "The helical axis of anatomical joints: calculation methods, literature review, and software implementation," Medical & Biological Engineering & Computing, vol. 60, no. 7, pp. 1815-1825.

[12] Salman, H. A., Kalakech, A., Steiti, A. (2024). "Random forest algorithm overview," Babylonian Journal of Machine Learning, pp. 69-79.

[13] Luo, Y., Zhang, R., Wang, F., Wei, T. (2023). "Customer segment classification prediction in the Australian retail based on machine learning algorithms," in Proceedings of the 2023 4th International Conference on Machine Learning and Computer Application, pp. 498-503.

[14] Asadi, S., Roshan, S., Kattan, M. W. (2021). "Random forest swarm optimization-based for heart diseases diagnosis," Journal of Biomedical Informatics, vol. 115, p. 103690.

[15] Luo, Y. (2023). "Identifying factors influencing China junior high students' cognitive ability through educational data mining: Utilizing LASSO, random forest, and XGBoost," in Proceedings of the 4th International Conference on Modern Education and Information Management (ICMEIM 2023), Wuhan, China, pp. 202-207.

[16] Liew, X. Y., Hameed, N., Clos, J. (2021). "An investigation of XGBoost-based algorithm for breast cancer classification," Machine Learning with Applications, vol. 6, p. 100154.

[17] Montesinos López, O. A., Montesinos López, A., Crossa, J. (2022). "Support vector machines and support vector regression," in Multivariate Statistical Machine Learning Methods for Genomic Prediction, Springer, pp. 337-378.

[18] Ozaki, Y., Tanigaki, Y., Watanabe, S., Nomura, M., Onishi, M. (2022). "Multiobjective tree-structured parzen estimator," Journal of Artificial Intelligence Research, vol. 73, pp. 1209-1250.