Visual Computing Magazine

A Systematic Procedure for Comparing Template-based Gesture Recognizers

Mehdi Ousmer, LRIM, University Catholic of Louvain La Neuve, Belgium

Abstract
This paper presents a conceptual framework and systematic procedure for conducting comparative testing of template-based gesture recognizers under controlled conditions. The proposed systematic procedure investigates the influence of various factors on recognition accuracy and responsiveness, including the number of templates, sampling points, fingers, and their configuration with other hand parameters, such as the palm. The experimental validation applies 6 selected gesture recognition algorithms tested on 2 established benchmark datasets from the literature and gesture recognition competitions.

1 Introduction

A diverse array of input devices, including game controllers, smartphones, and optical sensors, can be incorporated into three-dimensional gesture-based user interfaces (3D GBUI). These devices typically incorporate multiple motion sensors capable of measuring various physical quantities, thereby facilitating the broad application of gesture recognition algorithms in multiple domains.
This article focuses on designing and applying a systematic procedure for comparative testing. We rely on an adapted version of the AMMPT framework throughout this work for consistency in terminology [2, 17]. A systematic procedure refers to a structured sequence of steps followed in a defined order, while comparative testing evaluates software performance against competitors to improve quality and functionality [25]. While this method offers benefits for software improvement and market positioning, it can face challenges in implementing changes and managing customer perceptions [1].
Despite extensive comparative analyses in gesture recognition research [10,11], a comprehensive evaluation of hand gesture recognition using the Leap motion controller remains unexplored.
Extensive research exists on gesture recognition using advanced Machine Learning (ML) techniques for LMC [8]. These models excel at recognizing complex gestures and handling noisy data [9, 21]. Deep learning-based recognizers have become particularly prevalent [4, 20]. However, these techniques require large gesture datasets for training and validation to achieve accurate results with complex gestures.

Visual Computing Magazine

A Systematic Procedure for Comparing Template-based Gesture Recognizers

Mehdi Ousmer, LRIM, University Catholic of Louvain La Neuve, Belgium

This paper focuses on template matching recognizers. This technique provides a fast, simple, and accurate classification method. A template matching recognizer identifies candidate gestures by comparing them with predefined gesture classes, which are represented by a limited set of training templates. The selection of this technique is justified by several criteria:

1. Real-time interaction requirements for interactive applications demand optimal system performance, characterized by minimal response time (e.g., 0.1 seconds, according to Nielsen [15]) and high accuracy (e.g., a recognition rate of >90% [14, 26]).
2. A gesture dataset that's easy to design, modify, and expand, particularly for supporting user-defined gestures [27].
3. They require minimal AI/ML expertise, as they are easy to train and modify, understand and compute, and interpret, and offer straightforward integration into the development cycle.

2 Methodology

2.1 Gesture Recognition Testing

The gesture recognition testing procedure includes four distinct stages: elicitation, data acquisition, training, and testing. Although these stages constitute standard practice in the literature on gesture recognition evaluation, their implementation varies between works [13, 12, 28].

The initial phase, the Gesture Elicitation Study, enables collaborative gesture creation wherein participants design task-specific gestures reflecting end user preferences. In the subsequent data acquisition stage, multiple devices are used to capture gesture data. The training stage focuses on the development of the recognizer model based on the selected classification technique.

The final testing stage evaluates the trained model's recognition capabilities through performance metrics. To formally represent the gesture recognition testing procedure, the Software & Systems Process Engineering Meta-Model (SPEM) notation was employed [16, 17]) (Figure 1).

Visual Computing Magazine

A Systematic Procedure for Comparing Template-based Gesture Recognizers

Mehdi Ousmer, LRIM, University Catholic of Louvain La Neuve, Belgium

2.2 Comparative Testing

Gesture recognition comparative testing evaluates template-based recognizers' performance in multiple contexts of use. The testing usually involves comparing the accuracy or recognition rate of the recognizers and considering factors such as computational complexity, real-time capability, and feasibility on affordable hardware. Comparative testing involves evaluating recognizers against predefined test cases to identify the most accurate one. This process relies on a structured framework for comparing different template-based gesture recognizers. The literature identifies several key components essential for gesture recognition testing.

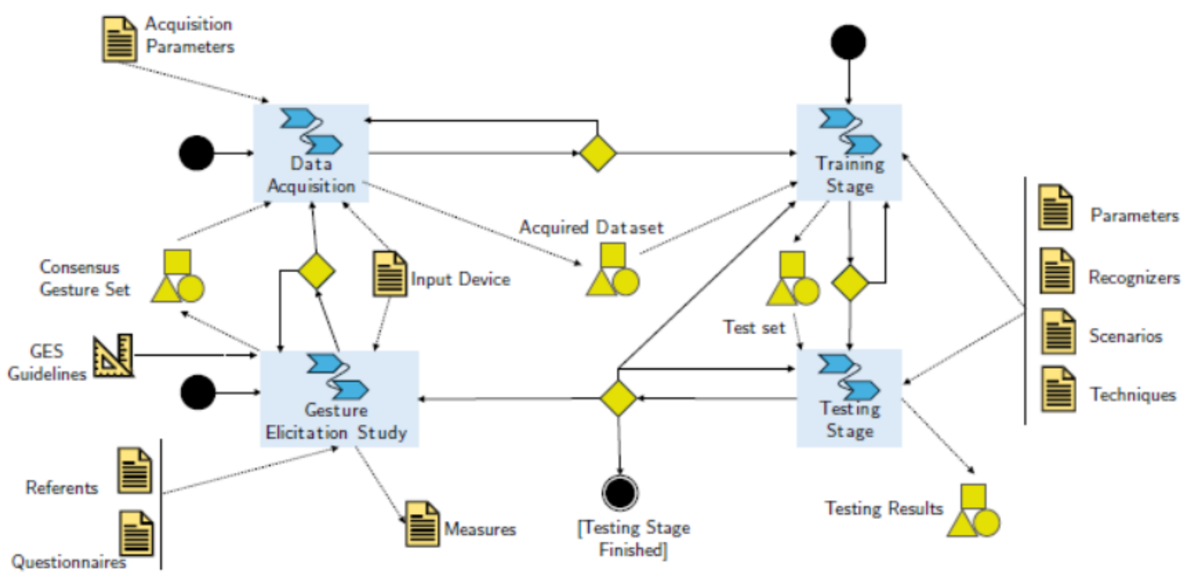


Figure 1: Gesture recognition testing procedure workflow. [17]

Visual Computing Magazine

A Systematic Procedure for Comparing Template-based Gesture Recognizers

Mehdi Ousmer, LRIM, University Catholic of Louvain La Neuve, Belgium

2.2.1 Recognizers

Template-based recognizers form the core testing component. They identify gestures by matching them against training templates. These recognizers must use the same programming language and be compatible with 3D hand gesture datasets.

2.2.2 Techniques

Many key evaluation techniques exist. These include k-fold cross-validation (dividing data into k test groups), 50-50 splits, leave-one-out cross-subject evaluation, and leave-one-out cross-session evaluation[5].

2.2.3 Scenarios

Testing covers single and multi-device configurations. Two main approaches are used: user-dependent (same user for training and testing) and user-independent (separate users for training and testing).

2.2.4 Datasets

Testing relies on gesture datasets collected from various users and devices. Public datasets are preferred for reproducibility. Dataset selection must consider the context of use, gesture diversity and variability, and data availability.

2.2.5 Parameters

The main parameters include the number of templates (T) and repetitions (number of tests) (R) for the user-dependent scenario, plus the number of participants (P) for the user-independent scenario, with additional parameters for multi-device scenarios. Parameter values are based on experience, prior research, and dataset constraints.

2.3 Leap Motion Comparative Testing

The gesture recognition testing procedure shares elements with comparative testing. A unified testing process with repeatable stages that provides a systematic approach to this procedure through clearly defined activities. The use of consistent activities ensures completeness and systematization. This standardization enables valid comparisons between recognition algorithms, while using different procedures would lead to incomparable results.

Visual Computing Magazine

A Systematic Procedure for Comparing Template-based Gesture Recognizers

Mehdi Ousmer, LRIM, University Catholic of Louvain La Neuve, Belgium

The comparative testing presented in the following section aims to fill the gap in the literature [19]. We selected many datasets that were available for various reasons, including reproducibility. We evaluate the recognizers ($P3+ [18],$F [18], $P3+X: a variant of $P3+, Jackknife [23], PennyPincher3D:
a 3D adaptation of [22], 3¢ [7]) in a user-independent scenario. We test them on the full hand gestures provided by the LMC (Leap Motion Controller) skeletal hand model available in the SHREC2019 [6] (Figure 2) and Jackknife-LM [23]
datasets (Figure 3).

We measure both recognition rate and execution time for the 12 basic configurations (6 RECOGNIZER × 2 DATASET), following established evaluation methods used in the literature to evaluate gesture recognizers [3, 27, 24]: the user-independent scenario evaluates the recognition on gestures produced by users who are different from those used for training the recognizer. In this scenario, the basic configurations are refined depending on A (Figure 4), the number of joints, on T, the number of templates, and depending on N, the number of resampling points to train the recognizer.

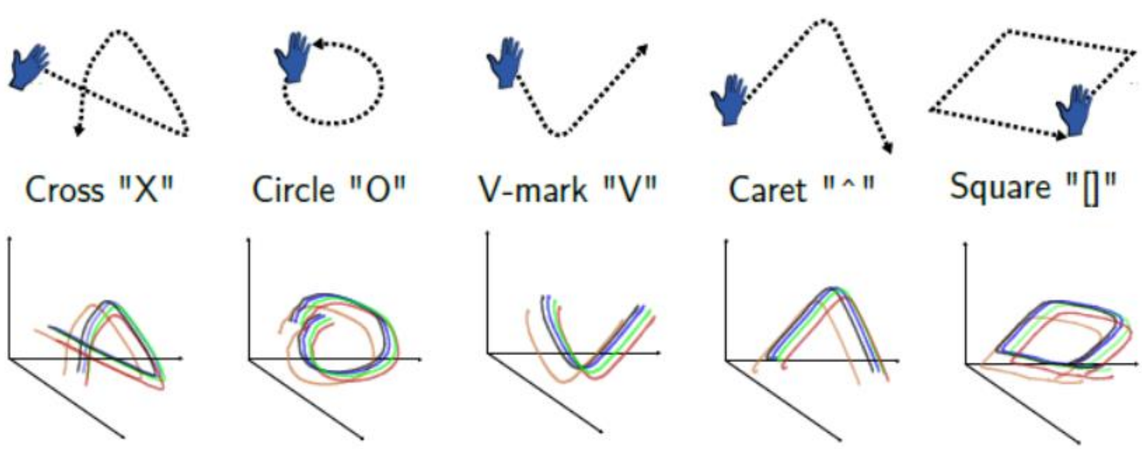


Figure 2: The SHREC2019 gesture classes and samples. [6]

Visual Computing Magazine

A Systematic Procedure for Comparing Template-based Gesture Recognizers

Mehdi Ousmer, LRIM, University Catholic of Louvain La Neuve, Belgium

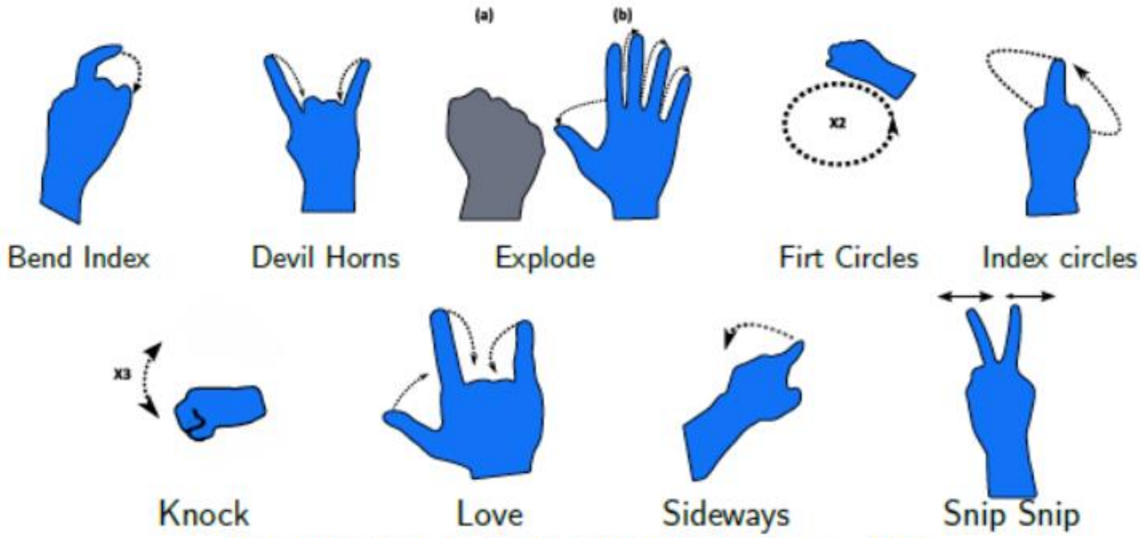


Figure 3: The Jackknife-LM gesture classes. [23]

3 Results & discussion

3.1 SHREC2019 Overall Recognition Rate

Figure 5 shows the average recognition rates on all tests. $P3+ leads with (M=85.90%, SD=15.44%), followed by $P3+X (M=84.90%, SD=16.45%) and Jackknife (M=82.11%, SD=14.06%). PennyPincher3D achieves (M=80.02%, SD=12.38%), while 3 Cent (M=78.09%, SD=18.12%) and $F (M=77.36%, SD=19.27%) fall below 80%.

Statistical analysis revealed that none of the recognition rates followed a normal distribution on the RECOGNIZER variable. Kruskal-Wallis testing showed highly significant differences between recognizers (p<.001***). $P3+ significantly outperformed all others, including $F (Z=48.84, p<.001***), Jackknife (Z=27.86, p<.001***), and $P3+X (Z=5.247, p<.001***). While $F and 3 Cent showed no significant difference (Z=1.928, n.s.), PennnyPincher3D outperformed 3 Cent (Z=3.055, p<.05*). For specific gesture classes:

- "Caret" and "V-mark" have average recognition rates greater than or equal to 90%.
- "Cross" outperformed "Square" and "Circle" in $F, $P3+ and $P3 + X.

A Systematic Procedure for Comparing Template-based Gesture Recognizers

Mehdi Ousmer, LRIM, University Catholic of Louvain La Neuve, Belgium

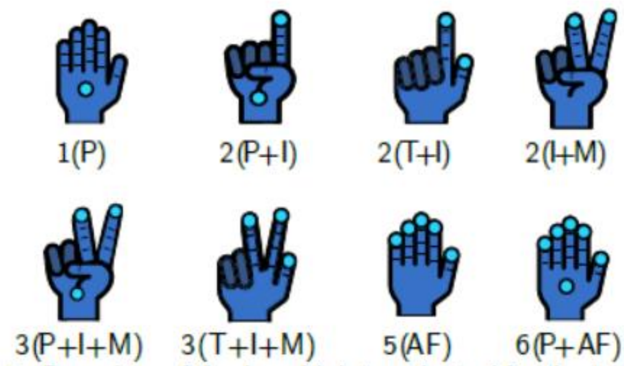


Figure 4: Overview of the hand joints selected for the testing [*A*].

• Jackknife best recognized "Cross" (M=93.98%), compared to $P3+ (M=83.35%) and $F (M=67.75%).
• "Square" and "Cross" showed similar recognition rates for Jackknife and PennyPincher3D.
• "Circle" performed poorly with Jackknife (M=34.43%), PennyPincher3D (M=21.91%), and 3 Cent (M=46.50%).
While PennyPincher3D, 3 Cent, and $F rarely met the 90% user expectation [14, 26], $P3+ and $P3 +X consistently did. For SHREC2019 testing, we relaxed this to τ≥80%.

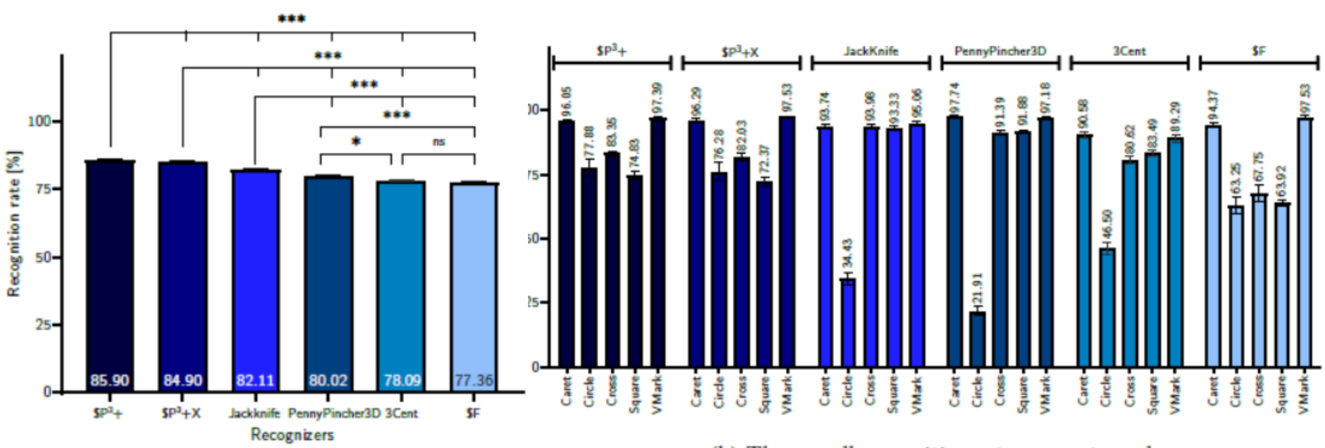
3.2 SHREC2019 Overall Execution Time

In Figure 6, recognizer execution times show that, the PennyPincher3D is the fastest (M=0.047 ms, SD=0.060 ms), while the $P3+ (M=1.611 ms, SD=3.328ms) is the slowest recognizer. Kruskal-Wallis and Dunn's tests revealed significant differences between all recognizer pairs except 3 Cent and $P3+X. PennyPincher3D was significantly faster than Jackknife (M=0.195 ms, difference: 148μs, Z=73.62, p<.001) and $P3+ (Z=147.7, p<.001).

Visual Computing Magazine

A Systematic Procedure for Comparing Template-based Gesture Recognizers

Mehdi Ousmer, LRIM, University Catholic of Louvain La Neuve, Belgium



(a) Average recognition Rate

(b) The overall recognition rate per gesture class

Figure 5: Recognition rates of all recognizers for the SHREC2019 for all conditions. Error bars show a confidence interval of 95%.

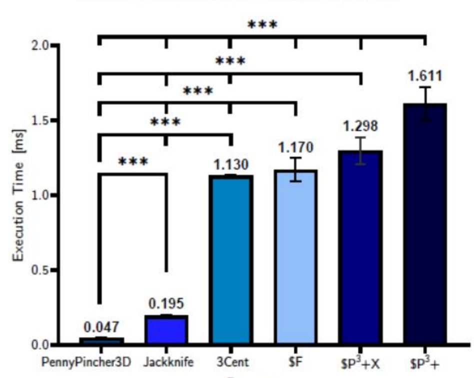


Figure 6: Average execution times by recognizer. Error bars show a confidence interval of 95%.

3.3 Jackknife-LM Overall Recognition Rate

The Jackknife-LM dataset contains more complex gesture classes than SHREC2019, where different joints can move independently of the hand movement. Testing results in Figure 7 showed no recognizer exceeded 80% average accuracy for all tests under all conditions. Jackknife performed best (M=73.60%, SD=19.75%), followed by \$P3+ (M=68.75%, SD=19.73%) and \$P3+X (M=68.15%, SD=20.51%).

A Systematic Procedure for Comparing Template-based Gesture Recognizers

Mehdi Ousmer, LRIM, University Catholic of Louvain La Neuve, Belgium

Lower performance was seen from 3 Cent (M=63.66%, SD=18.29%), $F (M=62.96%, SD=20.12%), and PennyPincher3D (M=56.87%, SD=19.99%). Most recognizers achieved high accuracy (τ≥80%) for "FistCircles", "Knock" and "Sideways" gestures, except PennyPincher3D which scored lower (M=62.77%) for "Knock". "Love", "BendIndex", "DevilHorns", and "SnipSnip" showed poor recognition (τ≤60%) in most recognizers, with Jackknife performing better (> 70%) on "SnipSnip" and "BendIndex". Results show better recognition for static finger poses compared to gestures involving finger movements.

3.4 Jackknife-LM Overall Execution Time

The average execution time results in Figure 8 showed significant variations between recognizers. PennyPincher3D was fastest (M=0.079 ms, SD=0.108 ms), while 3 Cent was slowest (M=10.585 ms, SD=5.911 ms) due to its Cubic Spline interpolation. $P3+, $F, and $P3 +X had similar performance (around 2 ms), while Jackknife combined high accuracy with fast execution (M=0.351ms, SD=0.508 ms).

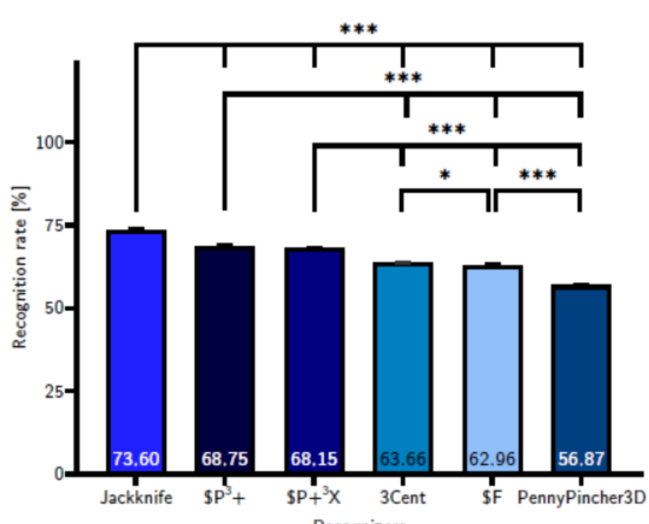


Figure 7: Recognition rates of all recognizers for the Jackknife-LM for all conditions. Error bars show a confidence interval of 95%.

A Systematic Procedure for Comparing Template-based Gesture Recognizers

Mehdi Ousmer, LRIM, University Catholic of Louvain La Neuve, Belgium

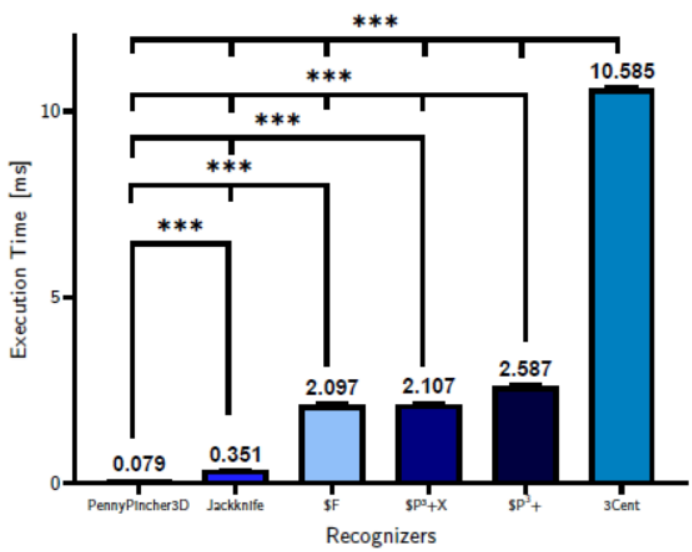


Figure 8: Average execution times by recognizer. Error bars show a confidence interval of 95%.

4 Conclusion

The comparative testing of the six selected gesture recognizers demonstrates the potential benefits of the systematic procedure for the comparative testing method. It provides many significant results depending on the dataset: the SHREC2019 dataset (an LMC-Based dataset with simple gestures) and the Jackknife-LM (which contains more complex gestures). Overall, the evaluation results showed that the recognizers perform differently for each dataset.

We analyzed the recognizers under many conditions and compared them under particular optimal conditions [19, 17]. These tests confirmed some of our previous results, showing that certain recognizers perform well for several configurations. Furthermore, we found that some recognizers are better suited to recognize specific gestures.

A Systematic Procedure for Comparing Template-based Gesture Recognizers

Mehdi Ousmer, LRIM, University Catholic of Louvain La Neuve, Belgium

References

[1] A. Ahmad. What is comparison testing, 2022.
[2] V. Andiappan and Y.Wan. Distinguishing approach, methodology, method, procedure and technique in process systems engineering. Clean Technologies and Environmental Policy, 22(3):547–555, apr 2020.
[3] L. Anthony and J. O. Wobbrock. A lightweight multistroke recognizer for user interface prototypes. In Proceedings of Graphics Interface 2010, GI '10, pages 245–252, Toronto, Ont., Canada, Canada, 2010. Canadian Information Processing Society.
[4] M. Asadi-Aghbolaghi, A. Clap´es, M. Bellantonio, H. J. Escalante, V. Ponce-L´opez, X. Bar´ o, I. Guyon, S. Kasaei, and S. Escalera. Deep Learning for Action and Gesture Recognition in Image Sequences: A Survey, pages 539–578. The Springer Series on Challenges in Machine Learning. Springer International Publishing, Cham, 2017.
[5] A. D. Berenguer, M. C. Oveneke, H.-U.-R. Khalid, M. Alioscha-Perez, A. Bourdoux, and H. Sahli. Gesturevlad: Combining unsupervised features representation and spatio-temporal aggregation for doppler-radar gesture recognition. IEEE Access, 7:137122–137135, 2019.
[6] F. M. Caputo, S. Burato, G. Pavan, T. Voillemin, H. Wannous, J. P. Vandeborre, M. Maghoumi, E. M. Taranta II, A. Razmjoo, J. J. LaViola Jr., F. Manganaro, S. Pini, G. Borghi, R. Vezzani, R. Cucchiara, H. Nguyen, M. T. Tran, and A. Giachetti. Online Gesture Recognition. In S. Biasotti, G. Lavou´ e, and R. Veltkamp, editors, Eurographics Workshop on 3D Object Retrieval, pages 93–102. The Eurographics Association, 2019.
[7] F. M. Caputo, P. Prebianca, A. Carcangiu, L. D. Spano, and A. Giachetti. A 3 cent recognizer: Simple and effective retrieval and classification of mid-air gestures from single 3d traces. In Proceedings of the Conference on Smart Tools and Applications in Computer Graphics, STAG '17, page 9–15, Goslar, DEU, 2017. Eurographics Association.
[8] I. A. S. Filho, E. N. Chen, J. M. da Silva Junior, and R. da Silva Barboza. Gesture recognition using leap motion: A comparison between machine learning algorithms. In ACM SIGGRAPH 2018 Posters, SIGGRAPH '18, New York, NY, USA, 2018. Association for Computing Machinery.
[9] H. M. Jais, Z. R. Mahayuddin, and H. Arshad. A review on gesture recognition using kinect. In 2015 International Conference on Electrical Engineering and Informatics (ICEEI), page 594–599, Denpasar, Bali, Indonesia, aug 2015. IEEE.
[10] A. S. Khalaf, S. A. Alharthi, I. Dolgov, and Z. O. Toups. A comparative study of hand gesture recognition devices in the context of game design. In Proceedings of the 2019 ACM International Conference on Interactive Surfaces and Spaces, ISS '19, page 397–402, New York, NY, USA, 2019. Association for Computing Machinery.
[11] R. Z. Khan. Comparative study of hand gesture recognition system. Volume 2, pages 203–213, 07 2012.
[12] S. Kotsiantis, I. Zaharakis, and P. Pintelas. Machine learning: A review of classification and combining techniques. Artificial Intelligence Review, 26:159–190, nov 2006.
[13] J. J. LaViola. 3d gestural interaction: The state of the field. International Scholarly Research Notices, 2013, 2013.

A Systematic Procedure for Comparing Template-based Gesture Recognizers

Mehdi Ousmer, LRIM, University Catholic of Louvain La Neuve, Belgium

[14] G. Marin, F. Dominio, and P. Zanuttigh. Hand gesture recognition with jointly calibrated leap motion and depth sensor. Multimedia Tools Appl., 75(22):14991–15015, November 2016.
[15] J. Nielsen. Usability Engineering. Interactive Technologies. Elsevier Science, 1994.
[16] O. M. G. (OMG). Software & systems process engineering metamodel specification, version 2.0. http://www.omg.org/spec/SPEM/2.0/PDF/, Apr 2008.
[17] M. Ousmer. A Systematic Procedure for Comparative Testing of 3D Hand Gesture Template-Based Recognizers in Multiple Contexts of Use. PhD thesis, UCL UCL UCL - SSH/LouRIM - Louvain Research Institute in Management and Organizations SST/ICTM - Institute of Information and Communication Technologies, Electronics and Applied Mathematics Ecole Polytechnique de Louvain, 2024.
[18] M. Ousmer, A. Slu¨yters, N. Magrofuoco, P. Roselli, and J. Vanderdonckt. Recognizing 3d trajectories as 2d multi-stroke gestures. Proceedings of the ACM on Human-Computer Interaction, 4(ISS):1–21, nov 2020.
[19] M. Ousmer, A. Slu¨yters, N. Magrofuoco, P. Roselli, and J. Vanderdonckt. A systematic procedure for comparing template-based gesture recognizers. In M. Kurosu, S. Yamamoto, H. Mori, D. D. Schmorrow, C. M. Fidopiastis, N. A. Streitz, and S. Konomi, editors, HCI International 2022 - Late Breaking Papers. Multimodality in Advanced Interaction Environments, pages 160–179, Cham, 2022. Springer Nature Switzerland.
[20] S. Sharma and S. Singh. Vision-based hand gesture recognition using deep learning for the interpretation of sign language. Expert Systems with Applications, 182:115657, 2021.
[21] J. Suarez and R. R. Murphy. Hand gesture recognition with depth images: A review. In 2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication, page 411–417. IEEE,sep 2012.
[22] E. M. Taranta, II and J. J. LaViola, Jr. Penny pincher: A blazing fast, highly accurate $-family recognizer. In Proceedings of the 41st Graphics Interface Conference, GI '15, pages 195–202, Toronto, Ont., Canada, Canada, 2015. Canadian Information Processing Society.
[23] E. M. Taranta II, A. Samiei, M. Maghoumi, P. Khaloo, C. R. Pittman, and J. J. LaViola Jr. Jackknife: A reliable recognizer with few samples and many modalities. In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, CHI '17, pages 5850–5861, New York, NY, USA, 2017. ACM.
[24] R.-D. Vatavu, L. Anthony, and J. O. Wobbrock. Gestures as point clouds: A $p recognizer for user interface prototypes. In Proceedings of the 14th ACM International Conference on Multimodal Interaction, ICMI '12, pages 273–280, New York, NY, USA, 2012. ACM.
[25] T. Vineet Nanda. Comparison testing in software engineering, 2021.
[26] C. Wang, Z. Liu, and S.-C. Chan. Superpixel-based hand gesture recognition with kinect depth camera. IEEE Transactions on Multimedia, 17(1):29–39, 2015.
[27] J. O. Wobbrock, A. D. Wilson, and Y. Li. Gestures without libraries, toolkits or training: A $1 recognizer for user interface prototypes. In Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology, UIST '07, pages 159–168, New York, NY, USA, 2007. ACM.
[28] M. Yasen and S. Jusoh. A systematic review on hand gesture recognition techniques, challenges and applications. PeerJ Computer Science, 5:e218, September 2019.