Summary of the changes applied to the manuscript

Chapter 1 - Introduction

- ☑ Better introduce key elements of gesture interaction for less-initiated readers. `Section 1.1`
    - ☑ Mid-air gestures.
    - ☑ Vision-based vs. radar-based sensors, LMC vs. radar.
- ☑ Discuss applications of gesture recognition (healthcare, art, education). `Section 1.1`
    - ☑ Where is gestural interaction appropriate?

Chapter 2 - State of the Art

- ☑ Discuss the problem of sensor fusion. `Section 2.1.2 / Sensor Limitations`
- ☑ Discuss the question of using DTW for segmentation and link to dynamic recognition. `Section 2.1.2 / Gesture Segmentation`
- ☑ Discuss UX challenges of gesture interaction. `Section 2.1.2`
    - ☑ Gesture discoverability and affordance
    - ☑ User fatigue (see "Consumed Endurance: A Metric to Quantify Arm Fatigue of Mid-Air Interactions"), mention how to compute the metric of "Gorilla-arm effect"
- ☑ Gesture set customization
- ☑ Provide a justification for limiting the SLR to the ACM DL (e.g., why not IEEE?). `Section 2.2`
- ☑ Emphasize that the use of point clouds does not require convolutional layers and discuss the impact. `Section 2.3.2 / Algorithms`

Chapter 3 - Facilitating the Development of Gesture-based Applications

- ☑ Clarify that "removeGesture(name)" removes all samples from the same gesture in the gesture set. `Section 3.1.1`

Chapter 4 - Streamlining the Evaluation of Gesture Recognizers

No modification.

Chapter 5 - Designing Highly Usable Gesture-based Applications

- ☑ Fix formula 5.1: P_1 should be P in the denominator. `Formula 5.1`
- ☑ Explicit the modifications applied to Vatavu's $P+ recognizer in $P3+. `Section 5.2.2 / Static Gestures`
- ☑ Consider the thinking aloud as a complementary method for evaluation. `Section 5.4.2`
    - Evaluation protocol is mostly quanti (LSM-EPL style), how about qualitative data? Cross-analyze

Chapter 6 - Addressing the Challenges of Radar-based Gesture Recognition

- ☑ Fix typo in computational complexity estimation: O(n ∗ k) should be O(n · k). `Section 6.2.5`
- ☑ Have you studied the effect of angle? It's mentioned "distance d from the radar and is orthogonal to the radar-hand direction". `Section 6.2.5`

- ☑ In data normalization, discuss the transfer from one sensor to another one (Radar challenges) `Section 6.2.2`
  - Would learned features transfer between radars or models? So many formats: FFT-based, Doppler-based, point clouds, etc.

Chapter 7 - Experimenting with Radar-based Gestures

- ☑ Discuss the problem of sensor fusion. `Section 7.2.4`
- ☑ Explicitly include, quote, and discuss the reference Leiva:2020 `Section 7.4.1` `Section 7.4.4`
  - Chapter 7: Missing citation (shameless plug): "The wearable radar" paper. We used the same training/testing aproach as shown in Table 146.
  - Gestures through materials dataset: Any thoughts about how would the results transfer/replicate between studies? See e.g. the Slovenia dataset.

Chapter 8 - Conclusion

- ☑ Discuss the problem of sensor fusion. `Section 8.2`

Miscellaneous

- ☑ Fix academic name of Luis A. Leiva. `Title page`
- ☑ Various formatting improvements.
- ☑ Added missing references of papers "in press" or just published.
- ☑ Fix typos. `Everywhere`

Suggested modifications not applied to the revised thesis

- □ Update Table 2.2 with more up-to-date references, since only 1 ref from 2021 was included.
  - I did not update the table, with the approval of my supervisors, as it would require analyzing all the new papers published since 2021 (around 40 relevant papers per year), including indentification, screening, eligibility, and inclusion steps, as well as updating the text and figures to account for the new references. This would not have been possible before the deadline for this revision of the thesis.
- □ Explain why JackKnife has 725 LOCs? It's substantially higher than the rest of recognizers. Was it written in the same programming language?
  - Answer to the question: We included the entire JavaScript implementation of Jackknife, which defines multiple parameters, parameters, and classes, that can be combined to adapt Jackknife to different use cases. A particular configuration of Jackknife could fit in a smaller number of LOCs but would require more work to slim down its code than just using it "as is", which is why we decided to count all of its LOCs.