Sphere Center Definition and Virtual-Sphere Handling Guide

Manual Boundary Selection and Initialization for Correlation Analysis

---

1. Purpose

This document describes the procedure used to define the sphere center and radius required for all subsequent spatial and temporal correlation analyses.

The same procedure is applied to:

- **solid-sphere recordings**, and
- **background-only recordings**, where a *virtual sphere* is defined for control analysis.

All steps are implemented using the script `make_center_from_boundary.py`.

---

2. Source Directory Structure

The analysis directory is organized as follows:

```
guide/
results_compare/
results_horizontal/
results_radial_autocorr/
results_vertical/
sample/

analyze_horizontal_continuity.py
```

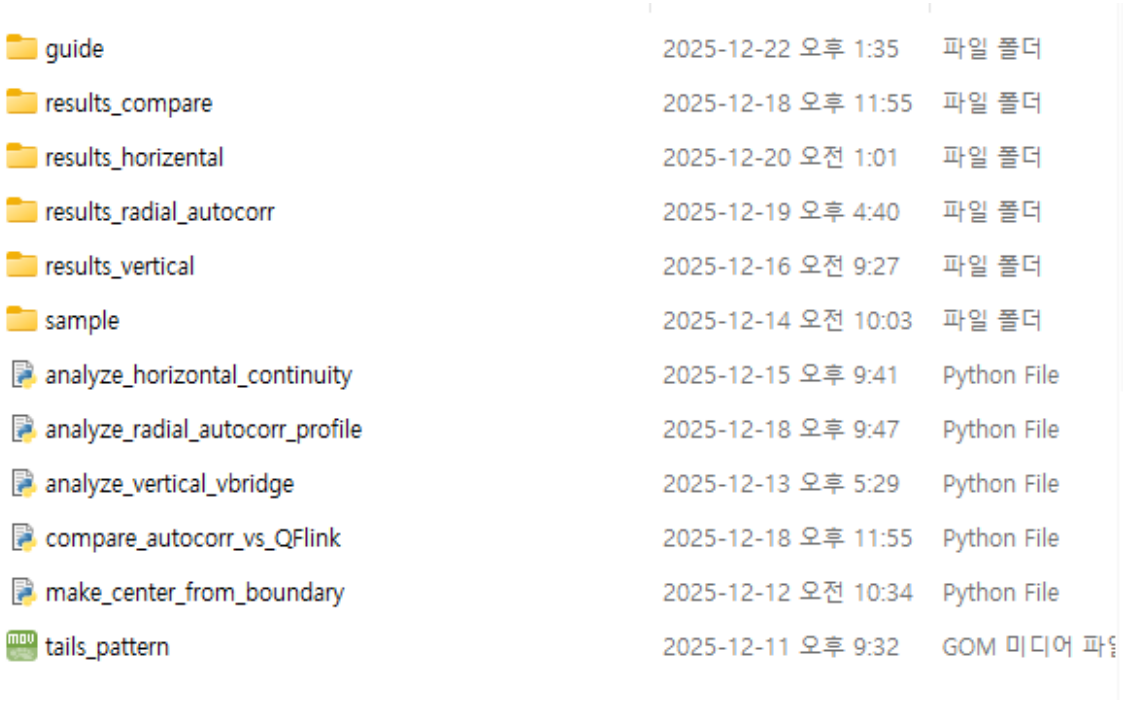
analyze_radial_autocorr_profile.py
analyze_vertical_vbridge.py
compare_autocorr_vs_QFlink.py
make_center_from_boundary.py

tails_pattern.mov

- All analysis scripts reside in the **same directory**.
- The input video file is placed in the same directory and named:

tails_pattern.mov

Figure 1. Example directory structure of the analysis pipeline.



---

3. Input Video File

- The input video file (tails_pattern.mov) is a **20-second recording**.
- Two types of videos are used:
    1. **Solid-sphere video** (sphere visible)

2. **Background-only video** (no sphere present)

The same filename convention is used for both cases to ensure a unified workflow.

---

4. Manual Boundary Selection (Solid Sphere Case)

4.1 Automatic Frame Selection

When `make_center_from_boundary.py` is executed:

- The script automatically loads the input video.
- The **middle frame** of the video is selected and displayed.
- Console output confirms:
  - total number of frames,
  - selected frame index.

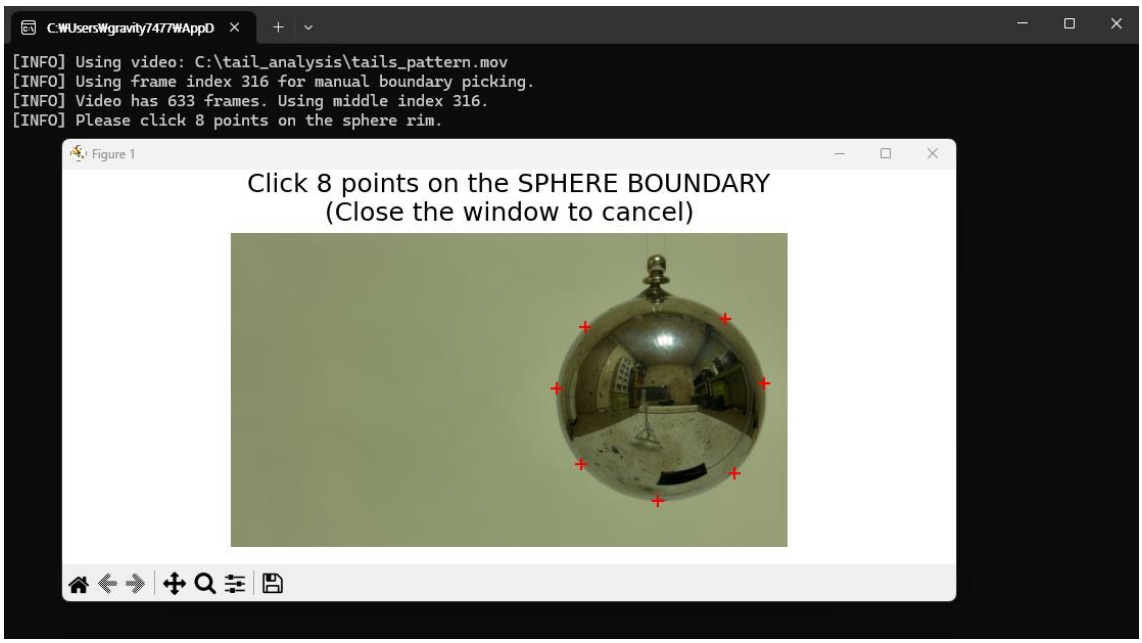
4.2 Manual Point Selection

The user is prompted to:

"Click 8 points on the sphere boundary"

- Exactly **8 points** are selected along the visible rim of the sphere.
- Points should be distributed evenly to capture the circular boundary.

Figure 2. Manual selection of 8 boundary points on the sphere rim.



4.3 Circle Fitting and Preview

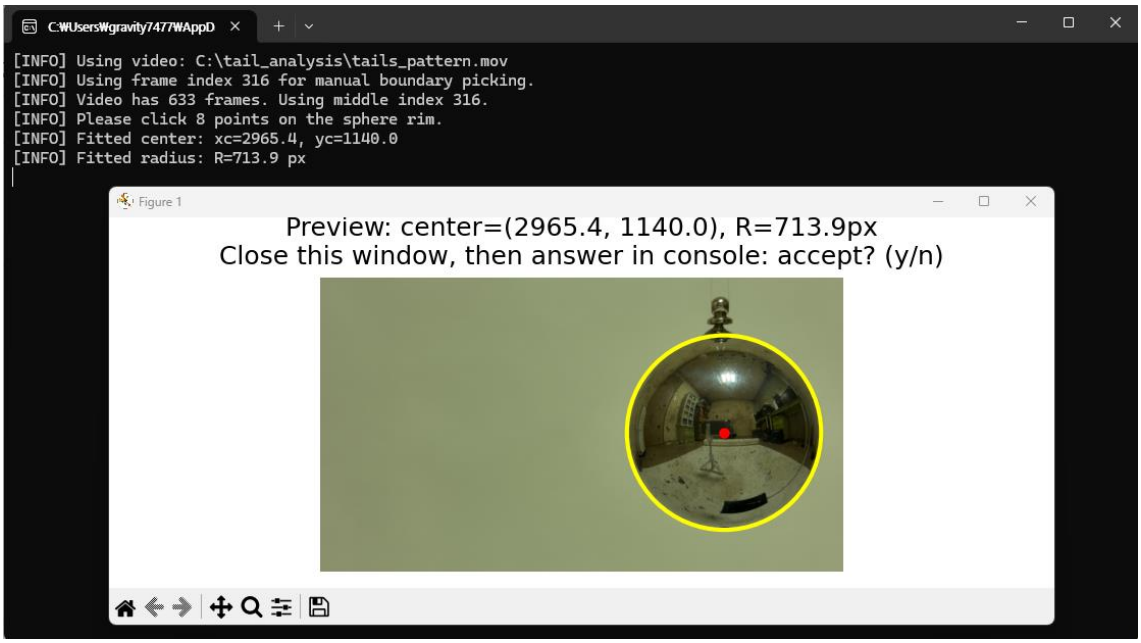
After point selection:

- A circle is fitted to the selected boundary points.
- The fitted result is displayed as a preview, showing:
    - estimated center position (xc,yc)(x_c, y_c)(xc,yc),
    - estimated radius RRR in pixels.

The user is then asked:

“Accept this fitted circle? (y/n)”

Figure 3. Preview of fitted sphere center and radius before confirmation.



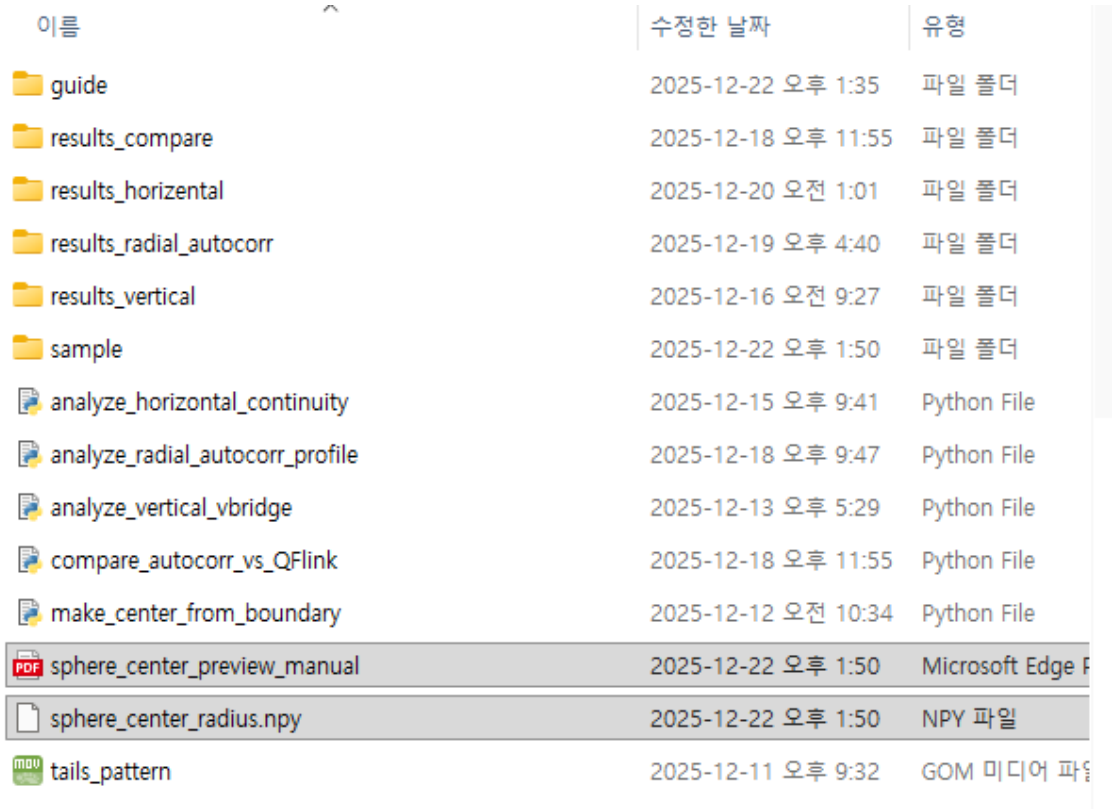
4.4 Confirmation and Output Generation

If the user confirms by entering **y**:

Two output files are generated automatically:

1. **sphere_center_preview_manual.pdf**
    - Visual record of the fitted circle
    - Used for documentation and verification
2. **sphere_center_radius.npy**
    - Binary data file containing:
        - sphere center coordinates
        - sphere radius
    - Used by all subsequent analysis scripts

Figure 4. Console confirmation and generated output files.



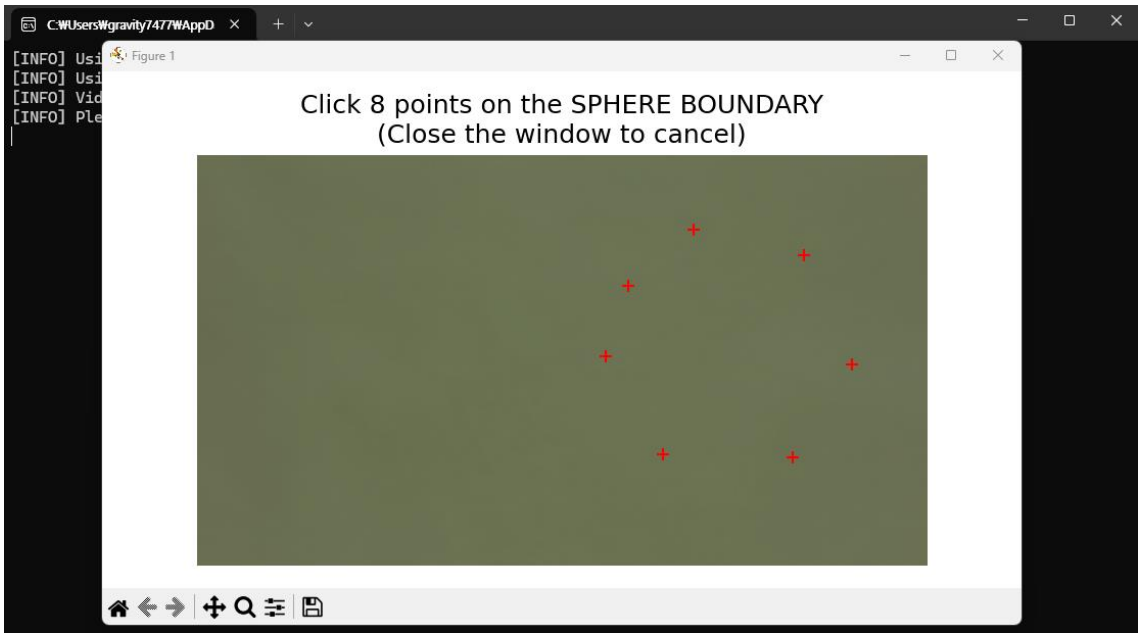
---

5. Virtual Sphere Definition (Background-Only Case)

For background-only recordings where no physical sphere is present:

- The same script is executed.
- Instead of selecting a visible boundary, the user places 8 points defining a **virtual circular region**.
- This virtual sphere defines:
  - a reference center,
  - a reference radius,
    used for control analyses.

Figure 5. Example of virtual sphere placement on a background-only frame.



This approach ensures that:

- spatial sampling geometry,
- radial binning,
- and analysis parameters

remain identical between solid-sphere and background-only cases.

---

6. Role in the Analysis Pipeline

The generated `sphere_center_radius.npy` file is a **foundational input** for:

- horizontal continuity analysis,
- radial autocorrelation profiles,
- vertical synchrony (bridge) analysis,
- comparative solid vs background studies.

All subsequent scripts assume that this file exists and defines the spatial reference frame.

7. Summary

In summary:

1. Place `tails_pattern.mov` in the analysis directory.
2. Run `make_center_from_boundary.py`.
3. Select 8 boundary points (physical or virtual).
4. Confirm the fitted circle.
5. Use the generated center and radius for all downstream analyses.

This procedure guarantees consistent spatial referencing across all experimental conditions.