Real-Time Vehicle Inference System: UWB-Driven UKF Update

Overview

This modification addresses the core data fusion strategy: using the **UWB measurement** to drive the Unscented Kalman Filter (UKF) update, rather than relying on the camera tracklet position. This is the theoretically correct approach, as the UWB tag is the primary source of ground truth for the person's identity and location.

Key Change in Data Fusion

The original implementation used the camera tracklet position for the UKF update, which is a simplification. The modified code now correctly uses the UWB measurement:

Fusion Strategy	Original (Simplified)	Modified (UWB-Driven)
UKF Update Source	Camera Tracklet Position	**UWB Measurement Position**
Trigger	UWB measurement available	UWB measurement available
Effect	Filter state is corrected by visual data	Filter state is corrected by UWB data (the identity anchor)

Code Snippet from `main_realtime_inference.py`

Python

```python
# --- 2. UKF Update (UWB-Driven) ---

# The UKF update is driven by the UWB measurement (3 FPS)
if uwb_measurement is not None:
    uwb_pos = uwb_measurement['position']
    # The UKF update is performed using the UWB position
    self.ukf.update_camera(uwb_pos, frame, timestamp)
```

Demonstration Results

The simulation was run with the controlled occlusion period (Frames 151-250).

Key Observations

The most significant change is the behavior of the **UKF Uncertainty** during the occlusion period:

Scenario	Original (Camera-Driven) UKF Uncertainty	Modified (UWB-Driven) UKF Uncertainty
Occlusion Period	**Grew rapidly** (from ~3 to ~351)	**Remained low** (at ~5.71)

Interpretation

1. **Original (Camera-Driven):** When the camera tracklet was lost, the UKF update was skipped entirely. The filter relied only on prediction, causing the uncertainty to skyrocket.
2. **Modified (UWB-Driven):** When the camera tracklet was lost, the UKF update **continued** to be performed by the UWB measurement (which is still available at 3 FPS). This UWB update anchors the filter state, preventing the uncertainty from growing and ensuring the UKF always has a high-confidence position estimate, even when the person is out of the camera frame.

Log Snippet Demonstrating UWB-Driven Stability

Plain Text

```
Frame 0150 | Time: 5.00s | Proc Time: 0.91ms | UKF Uncert: 5.71 |
Assignment: UWB 0 -> Cam 101 (OCCLUDED)
--- Occlusion Start at Frame 151 ---
Frame 0180 | Time: 6.00s | Proc Time: 0.41ms | UKF Uncert: 5.71 |
Assignment: None (OCCLUDED)
Frame 0210 | Time: 7.00s | Proc Time: 0.30ms | UKF Uncert: 5.71 |
Assignment: None (OCCLUDED)
Frame 0240 | Time: 8.00s | Proc Time: 0.31ms | UKF Uncert: 5.71 |
Assignment: None (OCCLUDED)
--- Occlusion End at Frame 250 ---
Frame 0270 | Time: 9.00s | Proc Time: 0.78ms | UKF Uncert: 5.71 |
Assignment: UWB 0 -> Cam 101
```

The constant, low UKF uncertainty during occlusion confirms that the UWB data is successfully maintaining the filter's state, which is the desired behavior for the Opt-in Camera system.

Files Delivered

File	Description
`uwb_ukf_2d_streaming.py`	Streaming UKF with buffer management (Unchanged)
`trajectory_matching_2d_streaming.py`	Real-time trajectory matching (Unchanged)
`vehicle_data_streamer.py`	Simulated asynchronous data stream generator (Unchanged)
`main_realtime_inference.py`	**MODIFIED:** Uses UWB measurement for UKF update
`REALTIME_VEHICLE_INFERENCE_UWB_DRIVEN_README.md`	This documentation

The complete, modified code is attached.