**Сбор и очистка сырых данных**

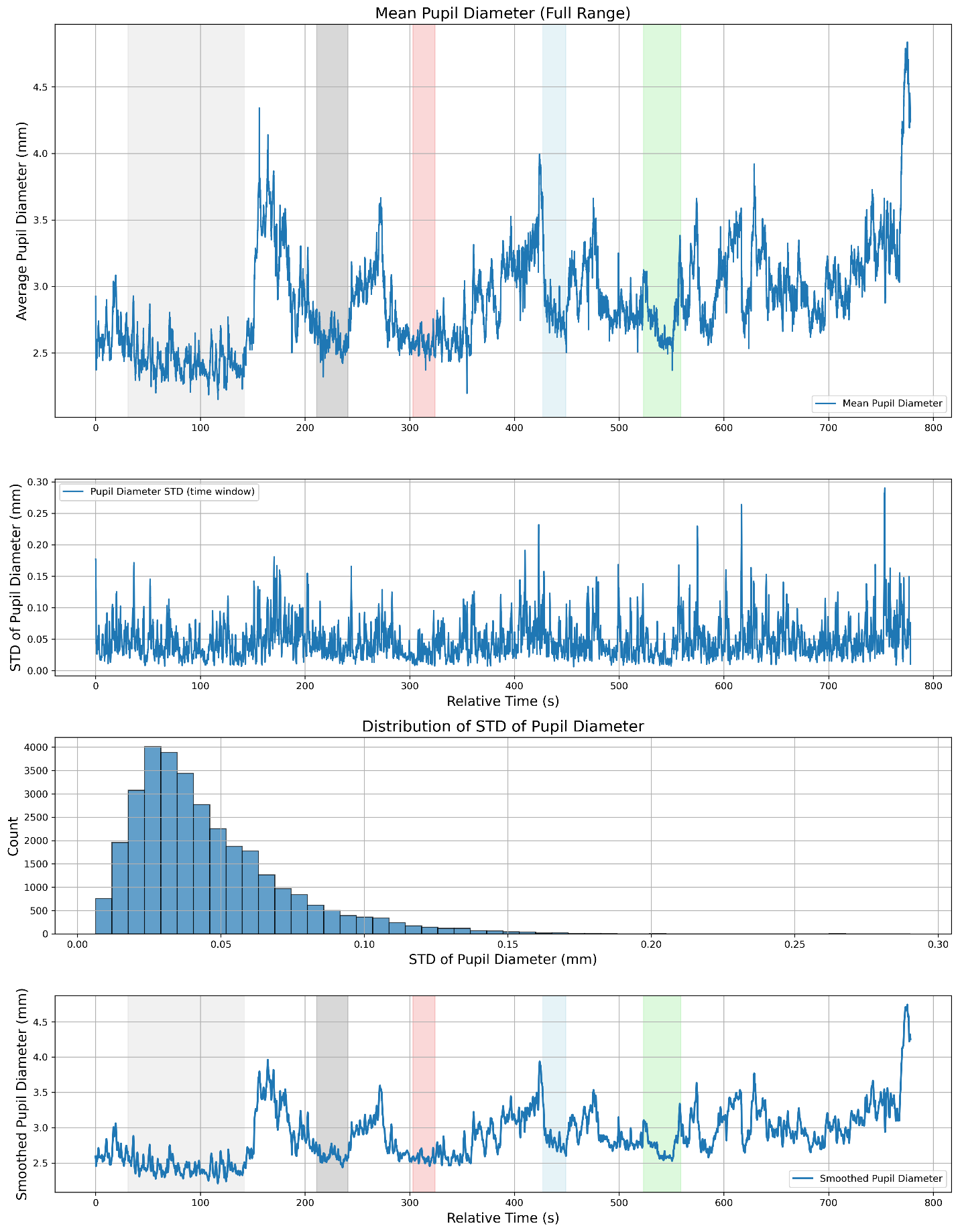
* Сначала вы загружаете «сырые» данные из .gz-файла через класс RawData.
* С помощью RawDataFilter создаёте объект FilteredData, в котором поочерёдно применяете серию фильтров (пропущенные значения, только gaze-события, проверка таймстемпов, диапазона размеров зрачка, скорости изменений и т. д.).
* Внутри FilteredData при каждом обновлении данных автоматически пересчитываются вспомогательные поля:
  + relative\_timestamp
  + avg\_pupildiameter
  + std\_pupildiameter
  + avg\_pupildiameter\_smooth

**Калибровка модели по тестовой сессии**

* Зная точные временные границы фаз эксперимента (Baseline, Positive, Neutral\_2, Negative) из файла timings\_by\_shelf.json, загружаем их через CalibrationProcessor.load\_time\_ranges.
* На фазе **Baseline** рассчитываем базовые характеристики реакции зрачка:
  + baseline\_mean и baseline\_std (среднее и СКО smoothed-значений).
* Для фаз **Positive** и **Negative** определяем динамику отклика:
  + скорость роста (rise\_speed) — насколько быстро зрачок расширяется от baseline до пика;
  + скорость спада (fall\_speed) — насколько быстро он возвращается к «обычному» уровню;
  + среднее отношение размера зрачка к baseline (mean\_ratio) — отражает амплитуду реакции.
* Для фазы **Neutral\_2** оцениваем скорость восстановления (recovery\_speed).
* Все эти вычисления выполняются статическими методами в CalibrationProcessor, а результаты упаковываются в контейнер CalibrationData.

**Использование параметров калибровки для автоматического анализа**

* Полученные в CalibrationData параметры (mean, std, rise/fall speeds, ratios и т. д.) образуют «модель» индивидуальной реакции данного клиента.
* При обработке новых данных вы повторяете этапы фильтрации (RawDataFilter → FilteredData), получаете аналогичные smoothed-показатели и относительные временные ряды.
* Сравниваете динамику нового сигнала с калибровочными метриками:
  + если скорость роста/спада и относительные изменения размерности попадают в диапазоны, характерные для Positive/Negative фаз — вы определяете соответствующее эмоциональное состояние;
  + отклонения от baseline\_mean и пороги на основе baseline\_std дают дополнительные сигналы «нейтральности» или «аномалий».



Я изучил все тестовые данные и отобразил средний размер зрачка поверх четырёх фаз, заданных в timings\_by\_shelf.json. На графике (четыре вертикальные полосы соответствуют фазам Baseline, Positive, Neutral\_2 и Negative) заметной синхронности между изменениями диаметра зрачка и переходами между фазами нет.

Это указывает на то, что текущие временные интервалы в timings\_by\_shelf.json не совпадают с реальным поведением данных — либо метки фаз смещены, либо сама разметка нуждается в корректировке. Чтобы устранить рассогласование, рекомендую: