# Integration Plan for Pixel Tracker GUI + imageutils.py

This document tracks the status of integrating the powerful command-line and batch utilities in imageutils.py into the PyQt5 GUI system, making all tools available in a user-friendly interface.

## ✅ Completed So Far

### ✅ GUI System Core

* PyQt5-based window with pixel selection
* Image sequence loading (.exr, .png, .jpg)
* AOV dropdown for EXR layers
* Frame slider for scrubbing sequences
* Pixel RGB tracking across simulations
* Export RGB data + categorical change flag as CSV and NPZ

### ✅ Backend Capabilities via imageutils.py

* AOV channel listing and extraction
* MP4 encoding for sequences and AOVs
* Colormap application to single-channel data
* Dataset analyzer (AOVs, resolution, formats, frame counts)
* Thumbnail generation and video summaries
* EXR channel visualization + comparison grid export
* Cleanup and batch frame operations

## 🔄 Current Work: GUI Integration of Tools

### 🧩 Tool 1: Dataset Analyzer

**Goal:** GUI button to scan a dataset root, then print: - number of sequences - AOVs found - resolution stats

**Status:** In Progress (Side console log output requested) **Function Used:** analyze\_dataset() and print\_dataset\_stats() from imageutils.py **Log Console:** Will be implemented with: - Read-only QTextEdit widget - Supports copy/paste - Auto-scroll on new messages - Timestamped log entries

### 🧩 Tool 2: AOV Extractor

**Goal:** GUI input for folder + list of AOV names + output format + colormap + video encoding - Extract frames to PNG or JPG - Apply colormaps if needed - Optionally create AOV MP4s

**Status:** Ready to implement **Function Used:** extract\_aovs\_from\_sequences()

### 🧩 Tool 3: Encode MP4 from a Sequence

**Goal:** Select a render folder and output an MP4 video of the image sequence **Status:** Ready to implement **Function Used:** encode\_sequence()

### 🧩 Tool 4: Thumbnail Frame Video

**Goal:** Create a summary video where each frame represents a different simulation’s result at a specific frame number. **Status:** Ready to implement **Function Used:** create\_thumbnail\_video()

### 🧩 Tool 5: Side-by-Side AOV Comparison Image

**Goal:** Choose one EXR frame and several AOV channels to visualize as a grid **Status:** Ready to implement **Function Used:** create\_aov\_comparison()

### 🧩 Tool 6: Cleanup Utilities

**Goal:** Button to scan or delete temp files, broken thumbnails, or cache folders **Status:** Ready to implement **Function Used:** cleanup\_temp\_files()

## 📌 Next Steps

1. Add a **Tools tab or side panel** in the GUI
2. Build buttons and file/folder pickers for each tool above
3. Add a **dockable log console** to show output (✅ Dataset Analyzer will use this)
   * Read-only QTextEdit
   * Supports copy/paste
   * Auto-scroll
   * Timestamps on log messages
4. Verbosely comment each tool integration script for clarity
5. Implement job/worker thread if needed for long tasks

## 🛠️ Optional Extras

* Create CLI frontend for headless operation
* Add per-frame pixel heatmap view
* Support patch/region-based pixel tracking
* Export timeline plots
* Export JSON project metadata for reproducibility

This list will be updated as each new feature is implemented. Let Sergio (that’s you) decide the priority — we can go visual, headless, or hybrid next!