

Frameworks for OpenCV GUI

Samuel Cheng

What we will talk about

- Built-in GUI framework (createTrackbar, createButton)
 - Very limited
 - Cvui
 - More flexible but very simple
 - Only use OpenCV drawing primitives (no OpenGL/Qt needed)
 - Conceptually very straightforward. No function callback
 - Qt
 - Cross-platform
 - More complicated (more boilerplate code)
 - Can create professional apps
 - Great documentation but usually in C++
 - Kivy
 - Like Qt, cross-platform and more polished
 - Better for mobile app
-
- We will use each of them to create a simple edge detection app with Canny

Qt

- Professional grade, better suit for desktop application
 - 3D slicer uses Qt
- When UI is activated, a **signal** will be sent
 - A **slot** can receive the signal and respond
- Designer
 - Signal/Slot editor is a convenient cheat to figure out the action name
- Use **pip install opencv-python-headless**
- Pyuic5
 - Convert ui to py file
- **@pyqtSlot(int)**
 - Create callback function
 - Be careful with the naming of the UI, it dies silently if the names do not match

Kivy

- More suitable for mobile, can create iPhone apps
- Link property to UI with NumericProperty/StringProperty/...
- Without designer, but overall quite easy to use as well
 - Need to specify the callback function explicitly. A plus to me as interpreter will be able to catch a typo on the callback function