

## Portability

All Kudan computer vision technology is designed to be hyper portable. We achieve this through multiple design decisions:

- **Pure C++ codebase**
  - Strict C++14 code
  - No operating system calls
  - Assembly language optimisations are all optional over a pure C++ version
- **C++ API**
  - Trivial to wrap with other languages (including C)
  - Compact API makes hook-up easy
  - API accepts multiple data formats and performs necessary conversions internally
  - API has rich error checking and warning messages to prevent misuse
- **No hardware access**
  - Hardware access is the responsibility of the caller, meaning we support anything that can produce camera data
  - Doesn't interfere with management of hardware interfaces
  - User can pre-process sensor data prior to calling KudanCV

## Optimisation

- Multiple ways to optimise components
  - SIMD instruction sets
  - GPU
  - FPGA
  - DSP
- Modular design allows selective optimisation
- Can utilise optimised OpenCV version

## Use-cases

- Head-tracking
  - Low jitter
  - Ability to track pre-made maps
  - Low latency
  - High accuracy/precision
- Mobile AR
  - Monocular rolling shutter
  - Mobile CPU friendly
  - Easy to combine with additional computer vision or other trackers
- Drones/robotics
  - Ability to make large maps
  - Ability to provide dense reconstructions for additional processing
  - Supports wide variety of sensors found on such devices
- Autonomous driving
  - Ability to make very large maps
  - Ability to utilise GPS data
  - Ability to utilise LiDAR data
  - Can utilise pre-made maps