# Software

Introduction of relevant software

# CMAKE

# CMAKE

- https://cmake.org

- Generate build scripts on for various environments/platforms
- A demonstration for OpenCV

- CMakeLists.txt
  - cmake_minimum_required(VERSION 2.8)
    project( DisplayImage )
    find_package( OpenCV REQUIRED )
    add_executable( DisplayImage DisplayImage.cpp )
    target_link_libraries( DisplayImage ${OpenCV_LIBS} )

# CMAKE -- OpenCV (1/2)

- CMakeLists.txt

```
cmake_minimum_required(VERSION 2.8)
project( HelloWorld_Proj )

find_package( OpenCV REQUIRED )

add_executable( HelloWorld_Exec src/main.cpp )
target_link_libraries( HelloWorld_Exec ${OpenCV_LIBS} )
```

- main.cpp

```
#include <stdio.h>
#include <opencv2/opencv.hpp>
.............
```

# CMAKE -- OpenCV (2/2)

- To generate build scripts:
  - cd <Where CMakeLists.txt is located>
    mkdir build
    cd build
    cmake-gui ..
  - (use „cmake ..” in console/terminal environment)
- To Compile your project
  - Linux
    - make
  - Windows
    - E.g. open the generated .sln using Visual Studio …

# nanoflann

# Nanoflann (1/2)

- https://github.com/jlblancoc/nanoflann

- Build kd-trees and perform NN searches
- C++, header only
- #include <nanoflann.hpp>

- Examples
  - https://github.com/jlblancoc/nanoflann/tree/master/examples
  - Eigen example: https://github.com/jlblancoc/nanoflann/blob/master/examples/matrix_example.cpp

# Nanoflann (2/2)

```
typedef KDTreeEigenMatrixAdaptor<Eigen::Matrix<num_t, Dynamic, Dynamic>>
  my_kd_tree_t;

my_kd_tree_t mat_index(dim, std::cref(mat), 10 /* max leaf */);
mat_index.index->buildIndex();

// do a knn search
const size_t num_results = 3;
vector<size_t> ret_indexes(num_results);
vector<num_t> out_dists_sqr(num_results);

nanoflann::KNNResultSet<num_t> resultSet(num_results);

resultSet.init(&ret_indexes[0], &out_dists_sqr[0]);
mat_index.index->findNeighbors(resultSet, &query_pt[0],
                               nanoflann::SearchParams(10));

std::cout << "knnSearch(nn=" << num_results << "): \n";
for (size_t i = 0; i < num_results; i++)
  std::cout << "ret_index[" << i << "]=" << ret_indexes[i]
            << " out_dist_sqr=" << out_dists_sqr[i] << endl;
```

# wave_geometry

# wave_geometry (1/4)

- Capabilites:
  - C++ library (expression templates)
  - Handles various transformations
  - Automatic Differentation
  - Compile-time coordinate frame semantics!

- Dependencies:
  - Eigen 3.3.2+
  - Boost 1.5.8+
  - C++17

- Install:
  - git clone https://github.com/wavelab/wave_geometry
  - cd wave_geometry
    mkdir build
    cd build
    cmake .. -DBUILD_TESTING=OFF
    sudo make install

# wave_geometry (2/4)

CMAKE:

- CMakeLists.txt

```
cmake_minimum_required (VERSION 3.8)
project (example)

set (CMAKE_CXX_STANDARD 17)

find_package (wave_geometry REQUIRED)

add_executable (example example.cpp)
target_link_libraries (example wave_geometry)
```

- example.cpp
  - #include <wave/geometry/geometry.hpp>
    …

# wave_geometry (3/4)

```
struct BodyFrame;
struct CameraFrame;
struct WorldFrame;

wave::RotationMFd<WorldFrame, BodyFrame> r1;
wave::RotationMFd<CameraFrame, WorldFrame> r2;

// Let's get the rotation between World and Camera (maybe)
wave::RotationMFd<WorldFrame, CameraFrame> result = r1 * r2;

// Let's get the rotation between World and Camera (fixed)
// wave::RotationMFd<WorldFrame, CameraFrame> result = r1 * inverse(r2);
```

http://paulfurgale.info/news/2014/6/9/representing-robot-pose-the-good-the-bad-and-the-ugly

# wave_geometry (4/4)

```
wave::RotationMd R = wave::RotationMd::Random();
wave::Translationd p1 = wave::Translationd::Random();
wave::Translationd p2 = R * p1;

// compute Jacobian w.r.t. R
Eigen::Matrix3d J_p2_wrt_R = (R * p1).jacobian(R);

// combine evaluation and multiple Jacobian calculations
auto [p2, J_p2_wrt_R, J_p2_wrt_p1] = (R * p1).evalWithJacobians(R, p1);
```

g2o

# g2o

- C++ framework
- Optimizing graph-based nonlinear error functions
- https://github.com/RainerKuemmerle/g2o

# g2o – Curve fitting example (1/3)

- https://github.com/RainerKuemmerle/g2o/blob/master/g2o/examples/data_fitting/curve_fit.cpp

```
g2o::SparseOptimizer optimizer;

g2o::OptimizationAlgorithmLevenberg* solver = new
   g2o::OptimizationAlgorithmLevenberg(
      g2o::make_unique<MyBlockSolver>(g2o::make_unique<MyLinearSolver>()));

optimizer.setAlgorithm(solver);


…
// Add vertices and edges to the graph
…

optimizer.initializeOptimization();
optimizer.optimize(maxIterations);
```

# g2o – Curve fitting example (2/3)

```
// the params, a, b, and lambda for a * exp(-lambda * t) + b
class VertexParams : public g2o::BaseVertex<3, Eigen::Vector3d> {
    ...
    virtual void oplusImpl(const double* update) {
        Eigen::Vector3d::ConstMapType v(update);
        _estimate += v;
    }
}
```

```
class EdgePointOnCurve : public g2o::BaseUnaryEdge<1, Eigen::Vector2d, VertexParams> {
    void computeError() { ... }
}
```

# g2o – Curve fitting example (3/3)

```
// build the optimization problem given the points

// 1. add the parameter vertex
VertexParams* params = new VertexParams();
params->setId(0);
params->setEstimate(Eigen::Vector3d(1,1,1)); // some initial value for the params
optimizer.addVertex(params);

// 2. add the points we measured to be on the curve
for (int i = 0; i < numPoints; ++i) {
  EdgePointOnCurve* e = new EdgePointOnCurve;
  e->setInformation(Eigen::Matrix<double, 1, 1>::Identity());
  e->setVertex(0, params);
  e->setMeasurement(points[i]);
  optimizer.addEdge(e);
}
```

# g2o – „ICP” example (1/2)

- https://github.com/RainerKuemmerle/g2o/blob/master/g2o/examples/icp/gicp_demo.cpp

```
// set up rotation and translation
Eigen::Isometry3d sensor_pose; // e.g. LiDAR pose
sensor_pose = Quaterniond(…);
sensor_pose.translation() = Vector3d(…);


// set up vertices of the graph for vertex_id = 0 and vertex_id = 1
VertexSE3 *vc = new VertexSE3();
vc->setEstimate(sensor_pose);
vc->setId(vertex_id);
// vc->setFixed(true);

optimizer.addVertex(vc);
```

# g2o – „ICP” example (2/2)

```
Edge_V_V_GICP * e = new Edge_V_V_GICP();

// add viewpoints
e->setVertex(0, vp0);
e->setVertex(1, vp1);


// add measurement (point-to-point correspondence)
e->setMeasurement(meas);


optimizer.addEdge(e);
```