VisualOdometry

- camera_intrinsics: cv::Mat
- distortion_coefficients: cv::Mat
- new_camera_matrix: cv::Mat
- image_width: int
- image_height: int
- vo_pose: Eigen::Matrix4d
- kp_curr: std::vector<cv::KeyPoint>
- des_curr: cv::Mat
- kp_prev: std::vector<cv::KeyPoint>
- des_prev: cv::Mat
- orb_descriptor: cv::Ptr<cv::ORB>
- flann_matcher: cv::FlannBasedMatcher

+ VisualOdometry(initial_pose: Eigen::Matrix4d)
+ ~VisualOdometry()
+ update_pose(image: cv::Mat)
+ get_pose(): Eigen::Matrix4d

DataLoader

- imu_file: std::ifstream
- gt_file: std::ifstream
- image_file: std::ifstream
- dataset_path: std::string
- current_image_index: size_t
- first_image_found: bool
- first_valid_timestamp: double
+ x_gt, y_gt, z_gt: std::vector<long double>
+ qx_gt, qy_gt, qz_gt, qw_gt: std::vector<long double>
+ start_gt_time: double
+ finish_gt_time: double

+ DataLoader(dataset_location: const std::string&)
+ ~DataLoader()
+ get_imu_data(): std::tuple<long double, Eigen::Vector3d,
                              Eigen::Vector3d>
+ parse_gt_data()
+ get_image_data(): std::tuple<double, cv::Mat, std::string>

InertialOdometry

- accelerometer_data: Eigen::Vector3d
- gyroscope_data: Eigen::Vector3d
- io_pose: Eigen::Matrix4d
- dt: float = 0.001

+ InertialOdometry(initial_pose: Eigen::Matrix4d)
+ ~InertialOdometry()
+ update_pose(a: Eigen::Vector3d, w: Eigen::Vector3d): void
+ rodrigues_formula(w: Eigen::Vector3d): Eigen::Matrix3d
+ get_pose(): Eigen::Matrix4d