

VisualOdometry

- camer_intrinsics: Eigen::Matrix3f
- prev_frame: cv::Mat
- curr_frame: cv::Mat
- keypoints: std::vector<cv::KeyPoint>
- matches: std::vector<cv::DMatch>
- + vo_pose: Eigen::Matrix4f

- + compute_keypoints(frame: const cv::Mat&):
std::vector<cv::KeyPoint>
- + match_keypoints(keypoints1: const std::vector<cv::KeyPoint>&,
keypoints2: const std::vector<cv::KeyPoint>&):
std::vector<cv::DMatch>
- + estimate_pose(matches: const std::vector<cv::DMatch>&,
camera_intrinsics: const Eigen::Matrix3f&):
Eigen::Matrix4f
- + track_motion(frame1: const cv::Mat&,
frame2: const cv::Mat&):
Eigen::Matrix4f
- + update_pose(): void

InertialOdometry

- accelerometer_data: std::vector<Eigen::Vector3f>
- gyroscope_data: std::vector<Eigen::Vector3f>
- bias: Eigen::Vector3f
- state_vector: Eigen::VectorXf
- + io_pose: Eigen::Matrix4f

- + preintegrate_IMU_data(imu_data: const
std::vector<Eigen::Vector3f>&,
bias: const Eigen::Vector3f&):
Eigen::MatrixXf
- + correct_bias(bias: const Eigen::Vector3f&):
Eigen::Vector3f
- + estimate_motion(preintegrated_data: const Eigen::MatrixXf&):
Eigen::Matrix4f
- + update_pose(): void

CeresOptimizer

- visual_estimates: std::vector<Eigen::Matrix4f>
- inertial_estimates: std::vector<Eigen::Matrix4f>
- initial_guess: Eigen::Matrix4f
- residual: std::function<double(const Eigen::Matrix4f&)>
- tolerance: double

- + set_initial_guess(initial_guess: const Eigen::Matrix4f&): void
- + add_residual_block(residual: const std::function<double(const Eigen::Matrix4f&)>&):
void
- + optimize(): Eigen::Matrix4f
- + set_tolerance(tolerance: double): void