# 词袋fbow/fbow.h：

**lib：${fbow\_LIBS}**

**namespace：fbow**

|  |  |  |
| --- | --- | --- |
| Vocabulary | | 词汇表 |
| **方法** | readFromFile / saveToFile(bin\_file) | 二进制文件**读写** |
| **fBow** transform(**cv::Mat** features) | **特征 →** 词袋 |

|  |  |  |
| --- | --- | --- |
| VocabularyCreator | | 词汇表训练器 |
| **属性** | ::Params | 参数表 **(k=32, L=-1, nthreads=1, maxIters=11)** |
| **方法** | create(**Vocabulary** voc,  **std::vector<cv::Mat>** features,  **std::string** desc\_name, **Params** params) | 训练 |

|  |  |  |
| --- | --- | --- |
| fBow | | 词袋 |
| **方法** | ::score(v1, v2) | L2范数 |

## DBoW3/DBoW3.h：

**lib：/usr/local/lib/libDBoW3.so**

**namespace：DBoW3**

**回环检测的阈值 (连续判定)：3 \* score(v[cur], v[cur - 1])**

|  |  |  |
| --- | --- | --- |
| Vocabulary | | 词汇表 **(k叉树)** |
| **形参** | k=10, L=5, weighting=TF\_IDF, scoring=L1\_NORM | 配置信息 |
| filename | 文件 |
| 方法 | create(**std::vector<cv::Mat>** training\_features) | 训练 |
| empty() | 空判断 |
| save("\*.yaml.gz") | 保存 |
| transform(**cv::Mat** features, **BowVector** out) | **特征 →** 词向量 |
| score(**BowVector** v1, **BowVector** v2) | 相似度 |

|  |  |  |
| --- | --- | --- |
| Database(**Vocabulary** voc, false) | | 数据库 |
| **方法** | add(**cv::Mat** features) | 添加**数据** |
| query(**cv::Mat** features, **QueryResults** out, max\_results) | 查询**相似数据** |

# 图像opencv2/opencv.hpp：

**C++：**

**package：OpenCV**

**lib：${OpenCV\_LIBS}**

**namespace：cv**

Python：复制Lib\site-packages\cv2\cv2.pyd到上一级

## 相机calib3d.hpp：

**点云x-y坐标的线性变换 (x-y轴缩放，原点平移，相机内参数)：**

|  |  |
| --- | --- |
| getOptimalNewCameraMatrix(cameraMatrix, distCoeffs,  imageSize, alpha, newImgSize) | 相机内参变换 |

|  |  |
| --- | --- |
| 标定： | |
| **bool** findChessboardCorners(img, chessSize, corners) | 寻找棋盘角点 |
| drawChessboardCorners(img, chessSize, corners) | 绘制棋盘角点 |
| cornerSubPix(gray, corners, winSize, zeroZone, criteria) | 修正角点位置 |
| calibrateCamera(objectPoints, imagePoints, imageSize, cameraMatrix,  distCoeffs, rvecs, tvecs, flags, criteria) | 相机标定 |

|  |  |  |
| --- | --- | --- |
| 去畸： | | |
| initUndistortRectifyMap(cameraMatrix, distCoeffs, R,  newCameraMatrix, size, m1type, mapx, mapx) | | 计算去畸坐标映射 |
| undistort(src, dst, cameraMatrix, distCoeffs) | | 图像去畸 |
| undistortPoints(src, dst, cameraMatrix, distCoeffs) | | 关键点去畸 |
| stereoRectify(cameraMatrix1, distCoeffs1, cameraMatrix2, distCoeffs2,  imgSize, R, T, R1, R2, P1, P2, Q) | | 双目矫正 |
| 形参 | Mat\_<float> cameraMatrix(3, 3); | 相机内参 |
| Mat\_<float> distCoeffs(5, 1); | 畸变参数 |

|  |  |  |
| --- | --- | --- |
| StereoSGBM | | 立体图像 |
| 方法 | ::create(0, 96, 9, 8\*9\*9, 32\*9\*9, 1, 63, 10, 100, 32) | 初始化 |
| compute(left, right, dst) | 计算视差 (CV\_16S) |

**// Depth = f \* B / Disparity**

**// 齐次坐标 (x, y, z, w) → 非齐次坐标 (x/w, y/w, z/w)**

|  |  |
| --- | --- |
| 2D-2D对极几何： | |
| findFundamentalMat(pts1, pts2, method) | 计算基本矩阵F |
| findEssentialMat(pts1, pts2, cameraMatrix) | 计算本质矩阵E () |
| findHomography(pts1, pts2, method) | 计算单应矩阵H (纯旋转) |
| recoverPose(E, pts1, pts2, cameraMatrix, R, t) | 恢复三维变换信息 (单位向量t) |
| triangulatePoints(projMatr1, projMatr2, pts1, pts2, points4D) | 三维点重建，返回点云齐次坐标 |

|  |  |
| --- | --- |
| 旋转： | |
| Rodrigues(src, dst) | 旋转向量 ↔ 旋转矩阵 |

|  |  |
| --- | --- |
| 3D-2D PnP： | |
| solvePnP(objPts1, imgPts2, cameraMatrix, distCoeffs, R, t) | 计算三维变换信息 |

## 内核core.hpp：

|  |  |
| --- | --- |
| cvRound / cvFloor / cvCeil(x) | 取整 |

|  |  |  |
| --- | --- | --- |
| FileStorage(file, FileStorage::Mode flags, encoding) | | xml / yaml / json文件  yaml头：%YAML:1.0 |
| 属性 | isOpened() | 打开 |
| FileNode operator[](key) | 键值查找 |
| 方法 | release() | 释放 |

|  |  |  |
| --- | --- | --- |
| **RNG(seed)** | | **随机数生成器** |
| **方法** | **uniform(low, high)** | **均匀分布** |
| **gaussion(sigma)** | **正态分布** |

|  |  |
| --- | --- |
| Ptr<dtype> | 引用计数指针 |
| TermCriteria(type, maxCount, eps) | **迭代**控制 |
| parallel\_for\_(**Range** range, functor) | 并行运算 |

|  |  |
| --- | --- |
| Size(w, h) | 2D尺寸 |
| Point(x, y) | 2D点 |
| Point3d(x, y, z) | 3D点 |
| Scalar(x, y, z, w) | 像素值 |

|  |  |  |
| --- | --- | --- |
| Rect | | 2D切片 |
| 形参 | x, y, w, h | 边界框 |
| pt1, pt2 | 对角线 |
| 方法 | tl / br() | 边界点 |
| area() | 面积 |
| contains(pt) | 判断包含点 |

|  |  |  |
| --- | --- | --- |
| InputArray | | 输入 |
| 属性 | getMat() | 图像 |

|  |  |  |
| --- | --- | --- |
| OutputArray | | 输入 |
| 属性 | getMat() | 图像 |
| 方法 | release() | 释放 |
| create(...) | 创建 (类似Mat) |

|  |  |  |
| --- | --- | --- |
| Mat | | 图像 |
| Mat\_<dtype> | | 矩阵 |
| 形参 | rows, cols, type, s | 2D矩阵 |
| ndims, sizes, type, s | nD矩阵 |
| 运算 | << x0, x1, ... | 写入元素 |
| 属性 | rows / cols / channels() | 形状 |
| dims / size() | 维度 |
| total() | 元素数 **(需乘通道)** |
| elemSize() | **元素**占用空间 |
| type / depth() | 类型 (位于**core/hal/interface.h**) |
| empty() | 为空 |
| 访问 | ptr<T>(0) / ptr<T>(r, c) | 数据 **/** 像素头指针 |
| at<T>(idx) | 像素 |
| row / col(i) | 单行 |
| rowRange / colRange(s, e) | 多行 |
| (rect) | 任意矩形 |
| 内存 | clone() / copyTo(mat) | 深拷贝 |
| convertTo(dst, rtype, w, b) | 类型转换**，并线性变换** |
| release() | 释放 |
| 填充 | ::eye | 单位阵，形参同初始化 |
| ::zeros / ::ones | 默认值，形参同初始化 |
| setTo(value, mask) | 指定值 |

|  |  |  |
| --- | --- | --- |
| 2D操作： | | |
| 常规操作 | print(src) | 输出 |
| flip(src, dst, flipCode) | 翻转 |
| 数值计算 | repeat(src, ny, nx) | 重复 |
| LUT(src, lut, dst) | 查表取值 |
| reduce(src, dst, axis, ReduceTypes) | 聚合 |
| 形状修改 | copyMakeBorder(src, dst, t, b, l, r, borderTypes btype, value) | 边界填充 |
| hconcat / vconcat(std::vector<Mat> src, dst) | 图像拼接 |
| transpose(src, dst) | 转置 |
| reshape(channels, rows) | 重塑 |

|  |  |
| --- | --- |
| 内存管理： | |
| copyTo(src, dst, mask) | 拷贝 |
| convertFp16(src, dst) | CV\_32F ↔ CV\_16S |
| parallel\_for\_(Range, func) | 并行 |

|  |  |
| --- | --- |
| 数值计算： | |
| add / subtract / multiply / divide /  bitwise\_and / bitwise\_or / bitwise\_xor(src1, src2, dst) | 逐元素运算 (可用运算符) |
| addWeighted(src1, alpha, src2, beta, gamma) | 逐元素加权 |
| max / min(src1, src2, dst) | 逐元素最值 |
| sum / mean(src) | 逐通道聚合 |
| sqrt(src, dst) / pow(src, power, dst) | 幂运算 |
| exp / log(src, dst) | 对数运算 |
| polarToCart / cartToPolar(x, y, magnitude, angle) | 极坐标转换 |
| norm(src) | 聚合 |
| hasNonZero / countNonZero(src) | 非零聚合 |
| flipND(src, dst, axis) | 翻转 |
| normalize(src, dst, alpha, beta, norm\_type) | 规范化 (默认单位化) |
| minMaxLoc(src, minVal, maxVal, minLoc, maxLoc) | 最值及位置 |

|  |  |
| --- | --- |
| 形状修改： | |
| merge(std::vector<Mat> src, dst) | 通道拼接 |
| split(src, std::vector<Mat> dst) | 通道拆分 |
| rotate(src, dst, RotateFlags rotateCode) | 旋转 |
| transposeND(src, std::vector<int> order, dst) | 通道重排 |
| reshape(nc, std::vector<int> newshape) | 重塑 |

## 特征feature2d.hpp：

|  |  |  |
| --- | --- | --- |
| **Keypoint** | | **关键点** |
| **属性** | **pt** | **位置** |
| **size** | **邻域直径** |
| **angle** | **夹角** |
| **response** | **响应** |
| **octave** | **金字塔层** |
| **方法** | **::convert(std::vector<Keypoint>, std::vector<Point2f>)** | **转换** |

|  |  |  |
| --- | --- | --- |
| **DMatch** | | **关键点描述子匹配** |
| **属性** | **queryIdx / trainIdx** | **关键点索引** |
| **distance** | **距离 (< 运算符)** |

|  |  |
| --- | --- |
| **绘制：** | |
| drawKeypoints(image, keypoints, outImage, **DrawMatchesFlags** flags) | 关键点 |
| drawMatches(img1, img2, matches1to2, outImg) | 匹配结果 |

|  |  |  |
| --- | --- | --- |
| **GFTT特征：** | | |
| GFTTDetector | | **GFTT特征**提取器 |
| **方法** | ::create(maxCorners, qualityLevel, minDistance) | 初始化 |
| detect(image, keypoints) | **检测**关键点 |

|  |  |  |  |
| --- | --- | --- | --- |
| **ORB特征：** | | | |
| **代码** | ORB | | **ORB特征**提取器 |
| **方法** | ::create(nfeatures) | 初始化 |
| detect(image, keypoints) | **检测**关键点 |
| compute(image, keypoints, descriptors) | **计算**描述子 |
| detectAndCompute(image, mask, keypoints, descriptors) | **检测并计算** |
| **Oriented FAST**  **关键点** | 角点 | **邻域圆的16个像素、中心点像素间比较，筛选角点** | |
| 尺度 | 图像金字塔分层检测 | |
| 旋转 | 中心点→质心 (**sum(灰度 \* 坐标) / sum(灰度)**) | |
| BRIEF描述子 | 随机选点比较得到0/1，转存为uint8 (使用汉明距离匹配) | | |

|  |  |  |
| --- | --- | --- |
| DescriptorMatcher | | 描述子匹配器 |
| **方法** | ::create("BruteForce-Hamming") | 初始化 |
| match(queryD, trainD, std::**vector<DMatch>** matches) | 匹配 |

**// 以max(min\_DMatch->distance, 30.0) 为阈值，筛选匹配结果**

## 交互highgui.hpp：

|  |  |  |
| --- | --- | --- |
| imshow(winname, mat) | 显示图像 | |
| waitKey(delay) | 0 | 无限期等待键盘输入 |
| int | 等待对应ms后关闭 |
| destroyAllWindows() | 关闭所有展示窗口 | |

## 编码imgcodecs.hpp：

|  |  |  |
| --- | --- | --- |
| **图像编码：** | | |
| **.png** | **CV\_8U / CV\_16U** | **无损压缩，支持BGRA** |
| **.jpg / .jpeg** | **CV\_8U / CV\_16U** | **有损压缩** |
| **.tiff** | **CV\_8U / CV\_16U / CV\_32F / CV\_64F** | **支持多图像** |
| **.exr** | **CV\_32F** | **支持压缩** |
| **.hdr** | **CV\_32F** | **无压缩** |

|  |  |
| --- | --- |
| imencode(suffix, img) | **图像编码**字节流 |
| imread(filename, ImreadModes flags = -1) | 打开图像并返回 |
| imwrite(filename, img) | 导出图像到文件 |

## 运算imgproc.hpp：

四通道：Blue, Green, Red, Opacity

Gray = 0.299 \* R + 0.587 \* G + 0.114 \* B

|  |  |
| --- | --- |
|  | |
| 色调 H |  |
| 饱和 S |  |
| 亮度 V |  |

|  |  |
| --- | --- |
| resize(src, dst, size) | 图像重置尺寸 |
| cvtColor(src, dst, ColorConversionCodes code) | 颜色空间转换 |
| threshold(src, dst, thresh, maxval, ThresholdTypes type) | 二值化 |
| remap(src, dst, mapx, mapy, InterpolationFlags flags) | 像素坐标重映射 |
| getRectSubPix(img, patchSize, center, patch) | 亚像素访问 |

|  |  |  |
| --- | --- | --- |
| 滤波： | | |
| filter2D(src, dst, ddepth, kernel) | 单通道卷积 | |
| blur(src, dst, ksize) | 均值滤波 | |
| GaussianBlur(src, dst, ksize, sigmaX=1) | 高斯滤波 | |  |  |  | | --- | --- | --- | | 0.6 | 0.8 | 0.6 | | 0.8 | 1 | 0.8 | | 0.6 | 0.8 | 0.6 | |
| medianBlur(src, dst, ksize) | 中值滤波 | |

|  |  |  |  |
| --- | --- | --- | --- |
| 腐蚀膨胀： | | | |
| erode(src, dst, kernel, iterations, borderValue) | | 腐蚀 (最小池化) | |
| dilate(src, dst, kernel, iterations, borderValue) | | 膨胀 (最大池化) | |
| morphologyEx(src, dst, MorphTypes op,  kernel, iterations, borderValue) | | 形态学运算 | |
| op：  运算模式 | MORPH\_OPEN | 开运算除去毛刺 | 腐蚀 => 膨胀 |
| MORPH\_CLOSE | 闭运算 | 膨胀 => 腐蚀 |
| MORPH\_GRADIENT | 梯度运算求边缘 | 膨胀 - 腐蚀 |
| MORPH\_TOPHAT | 礼帽求毛刺 | 原图 - 开运算 |
| MORPH\_BLACKHAT | 黑帽 | 闭运算 - 原图 |

|  |  |  |
| --- | --- | --- |
| 梯度算子： | | |
| Sobel(src, dst, ddepth, dx, dy, ksize) | Sobel | |  |  |  | | --- | --- | --- | | -1 | 0 | +1 | | -2 | 0 | +2 | | -1 | 0 | +1 | |
| Scharr(src, dst, ddepth, dx, dy) | Scharr | |  |  |  | | --- | --- | --- | | -3 | 0 | +3 | | -10 | 0 | +10 | | -3 | 0 | +3 | |
| Laplacian(src, dst, ddepth, ksize) | Laplacian | |  |  |  | | --- | --- | --- | | 0 | 1 | 0 | | 1 | -4 | 1 | | 0 | 1 | 0 | |

|  |  |  |
| --- | --- | --- |
| 直方图均衡： | | |
| Ptr<CLAHE> createCLAHE(clipLimit, tileGridSize) | | CLAHE算法 |
| 方法 | apply(src, dst) | 应用 |

|  |  |  |  |
| --- | --- | --- | --- |
| 轮廓检测： | | | |
| findContours(gray, contours, RetrievalModes mode, ContourApproximationModes method) | | | |
| mode：  检索模式 | RETR\_EXTERNAL | 外轮廓 | |
| RETR\_LIST | 所有  轮廓 | 保存到列表 |
| RETR\_CCOMP | 分为外轮廓、内轮廓 |
| RETR\_TREE | 重构嵌套轮廓的整个层次 |
| method：  逼近方法 | CHAIN\_APPROX\_NONE | 以Freeman链码的方式输出轮廓，  所有其他方法输出多边形 (顶点的序列) | |
| CHAIN\_APPROX\_SIMPLE | 压缩直线部分，只保留终点部分 | |

|  |  |  |
| --- | --- | --- |
| 曲线： | | |
| contourArea(contour) | 返回曲线属性 | 面积 |
| arcLength(curve, closed=False) | 周长 |
| boundingRect(curve\_or\_gray) | 曲线/图像外接矩形信息 | |
| minEnclosingCircle(points, center, radius) | 外接圆信息 | |

|  |  |
| --- | --- |
| 绘制： | |
| line(img, pt1, pt2, color, thickness) | 直线 |
| rectangle(img, pt1, pt2, color, thickness) | 矩形 |
| circle(img, center, radius, color, thickness) | 圆形 (实心：thickness=-1) |
| ellipse(img, center, axes, angle,  startAngle, endAngle, color, thickness) | 椭圆 |
| polylines(img, pts, isClosed, color, thickness) | 多边形 |
| putText(img, text, org, fontFace, fontScale, color, thickness) | 字符串 |
| drawContours(img, contours, idx, color, thickness) | 轮廓 (idx：索引，-1所有) |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 模板匹配： | | | | | |
| matchTemplate(image, templ, result, TemplateMatchModes method) | | | | | |
| method：  计算方法 | TM\_SQDIFF\_NORMED | 归一化 | 平方差 | 越 → 0 | 越相关 |
| TM\_CCORR\_NORMED | 相关性 | 越 → 1 |
| TM\_CCOEFF\_NORMED | 相关系数 |

## 视频video.hpp：

|  |  |  |
| --- | --- | --- |
| VideoCapture(file=0) | | 打开视频 / 摄像头 |
| 运算 | >> img | 读取视频帧 |
| 方法 | isOpened() | 检测状态 |
| read(dst) | 读取视频帧 |
| get(VideoCaptureProperties propID) | 获取参数 |
| set(VideoCaptureProperties propID, value) | 设置参数 |

|  |  |
| --- | --- |
| VideoCaptureProperties | |
| CAP\_PROP\_FRAME\_WIDTH | 图像宽度 |
| CAP\_PROP\_FRAME\_HEIGHT | 图像高度 |
| CAP\_PROP\_FRAME\_COUNT | 帧数 |
| CAP\_PROP\_FPS | FPS |
| CAP\_PROP\_POS\_MSEC | 视频当前位置 (ms) |
| CAP\_PROP\_CONTRAST | 相机亮度 |
| CAP\_PROP\_GAIN | 相机白平衡 |
| CAP\_PROP\_EXPOSURE | 相机曝光 |

|  |  |  |
| --- | --- | --- |
| 光流法： | | |
| 代码 | calcOpticalFlowPyrLK(prevImg, nextImg, prevPts,  nextPts, status, err) | LK稀疏光流 (关键点跟踪) |
| calcOpticalFlowFarneback(prev, next, flow, pyr\_scale,  levels, winsize, iterations, poly\_n, poly\_sigma, flags) | Farneback稠密光流 (流图像) |
| 假设 | 连续两帧图像之间，目标的像素亮度不改变；相邻的像素之间有相似的运动 | |
| 意义 | 像素的位移值，两通道分别表示x,y | |

# Device：

**Docker USB：<https://github.com/dorssel/usbipd-win>**

## librealsense2/rs.hpp：

**package：realsense2**

**lib：${realsense2\_LIBRARY}**

**namespace：rs2**

**ROS: <https://github.com/IntelRealSense/realsense-ros>**

**IMU标定: <https://github.com/gaowenliang/imu_utils>**

|  |  |
| --- | --- |
| RS2\_API\_VERSION\_STR | 版本号 |

|  |  |  |
| --- | --- | --- |
| context | | **上下文**管理器 |
| **属性** | device\_list query\_devices() | 设备列表 |

|  |  |  |
| --- | --- | --- |
| config | | 配置 |
| **方法** | enable\_stream(rs2\_stream, w, h, rs2\_format, framerate=0) | 启用流 |

|  |  |  |
| --- | --- | --- |
| pipeline | | 流水线 |
| **方法** | pipeline\_profile start(config cfg=NULL) | 启动 |
| frame\_set wait\_for\_frames(timeout\_ms) | 等待帧集 |
| bool try\_wait\_for\_frames(frame\_set \*, timeout\_ms) |

|  |  |  |
| --- | --- | --- |
| error | | 错误 |
| **属性** | get\_failed\_function() | 函数 |
| get\_failed\_args() | 参数 |

|  |  |  |
| --- | --- | --- |
| stream\_profile | | 简介 |
| **属性** | rs2\_format format() | 格式 |
| int fps() | 帧率 |
| std::string stream\_name() | 名称 |
| rs2\_stream stream\_type() | 类型 |

|  |  |  |
| --- | --- | --- |
| frameset | | 帧集 |
| **属性** | size() | 尺寸 |
| **方法** | foreach\_rs(action) | 映射 |
| frame first\_or\_default(rs2\_stream, rs2\_format) | 索引指定帧 |
| frame get\_color\_frame() | 索引RGB图 |
| frame get\_depth\_frame() | 索引深度图 |
| frame get\_fisheye\_frame(index=0) | 索引鱼眼图 |
| frame get\_pose\_frame(index=0) | 索引位姿 |

|  |  |  |
| --- | --- | --- |
| frame | | 帧 |
| **属性** | as<T>() | 类型转换 |
| get\_data() | 数据指针 |
| stream\_profile get\_profile() | 简介 |

|  |  |  |
| --- | --- | --- |
| video\_frame | | 视频帧 (public: frame) |
| **方法** | get\_width / get\_height() | 尺寸 |

|  |  |  |
| --- | --- | --- |
| pose\_frame | | 位姿帧 (public: frame) |
| **方法** | rs2\_pose get\_pose\_data() | 位姿 |

|  |  |  |
| --- | --- | --- |
| depth\_frame | | 视频帧 (public: video\_frame) |
| **方法** | get\_units() | 单位距离 |