|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 变换类型  效果参数 | 亮度  source + cv::Scalar(t,t,t,t) | 高斯模糊  cv::GaussianBlur()实现，高斯核是参数边长为t\*2+1大小的矩阵 | 旋转  warpAffine函数实现旋转。getRotationMatrix2D函数得到旋转中心，旋转角度  cv::INTER\_CUBIC内插 | 尺度缩放  Resize()实现  source.cols \* t + 0.5f  (source.rows \* t + 0.5f))  cv::INTER\_AREA内插 |
| 矩阵误差。配准求得的变换矩阵与真实的矩阵之积应该是单位阵。矩阵之积与单位阵之差，求其无穷范数（只取小于1的部分） | 表现最好的是SIFT、SURF、BRISK。ORB在亮度-100的部分最差 | 趋势都是模糊程度越高，效果也差。表现最好的依次是KAZE、AKAZE、SIFT、SURF。但KAZE在核大于7后有一个跃变 | 表现最好的依次是KAZE、AKAZE、BRISK。  SIFT的性能曲线较平滑，在参数180度取极小值，左右对称分布，大体呈上凸的抛物线 | SIFT性能最好，且最稳定。其次是SURF和KAZE，KAZE波动更大。 |
| correctMatchesPercent  =correctMatches / matchesCount  正确匹配对与初始匹配对之比。距离误差小于3认为是正确匹配 | SIFT、BRISK、KAZE、AKAZE效果最好， | 表现最好的依次是AKAZE、KAZE，在80%以上。ORB性能次之，但性能下降最快 | 表现最好的是AKAZE、SIFT、ORB。最差的是SURF。所以算法都在90  度倍数附近有较大极值 | 表现最好的是AKAZE、KAZE、ORB、SIFT |
| percentOfMatches=matchesCount / (s.totalKeypoints)初匹配对数与变换后的待配准图像提取出的特征点的数目之比 | 表现最好的依次是AKZE、SIFT、KAZE，都在90%以上 | AKAZE、BRISK、KAZE表现最好，且能维持在90%以上。SIFT在高斯核变大后反而性能变好，也可达90% | AKAZE表现最好，ORB、SIFT、KAZE其次。都有几个明显的极值，分别是90、180、270度左右 | AKAZE表现最好，其余除了ORB都差不多。但尺度大于1时除了BRISK和ORB，其余都急剧下降到低水平 |
| matchingRatio() = correctMatchesPercent \* percentOfMatches \* 100.0f; | 表现最好的是AKAZE。性能曲线走势和correctMatchesPercent相近 | 同左 | 同左 | 同左 |
| Meandistance 将待配准图像按照求出的矩阵的逆变换回去后的特征点与原图的距离的均值 | SURF、KAZE距离的平均值几乎为0，KAZE稍高一些也只有0.2左右。其次是SIFT | SURF、KAZE距离的平均值几乎为0，KAZE稍高一些也只有0.2左右。其次是AKAZE、ORB | SURF、KAZE距离的平均值几乎为0，KAZE稍高一些也只有0.2左右。其次依次是ORB、AKAZE。所有算法都有明显的极小值 | SURF、KAZE距离的平均值几乎为0，KAZE稍高一些也只有0.2左右  其次是ORB、AKAZE |
|  |  |  |  |  |
|  |  |  |  |  |

KAZE算法原理与源码分析中比较了不同的配准算法在不同变换下的表现，然而在即便符合了OpenCV2.4的条件，要想加入kaze在zou的代码下还是缺少utility,换用opencv3.2可以实现kaze，但要想加入sift还要再编译conmtrib部分，并且代码形式也发生了变换。最终实现了kaze、AKAZE、orb、brisk、sift、surf的比较，并且在表格中利用不同评价标准进行评价，还加入了特征点数目的比较，之后还准备加入以峰值信噪比的评价标准。