**SIFT**
**SURF**
**ORB**
**SIFT**
**?**
**Scale-space Extrema Detection**
**Laplaciana di una Gaussiana**
$\sigma$
$\sigma$
$\sigma$
$k\sigma$
$\sigma$
$\sqrt{2}$
**Keypoint Localization**
**Orientation Assignment**
$\sigma$
**Keypoint Descriptor**
**Keypoint Matching**
*SiftManager*
**Fase Uno :**
*extractKeypoints*
*extractDescriptors*
**Fase Due :**
$_L2, true); matcher.match(objectDescriptors, sceneDescriptors, matches); returnmatches;$
**Fase Tre :**
$_DRAW_SINGLE_POINTS); List <$
$KeyPoint >$
$objkeys =$
$objectKeyPoints.toList(); List <$
$KeyPoint >$
$scnkeys =$
$sceneKeyPoints.toList(); List <$
$DMatch >$
$goodMatches =$
$matches.toList(); List <$
$Point >$
$oPoints =$
$goodMatches.stream().map(m- >$
$objkeys.get(m.queryIdx).pt).collect(Collectors.toList()); List <$
$Point >$
$sPoints =$
$goodMatches.stream().map(m- >$
$scnkeys.get(m.trainIdx).pt).collect(Collectors.toList()); MatOfPoint2fobjectMatrix =$
$newMatOfPoint2f(); MatOfPoint2fsceneMatrix =$
$newMatOfPoint2f(); objectMatrix.fromList(oPoints); sceneMatrix.fromList(sPoints); Mathomography =$
$Calib3d.findHomography(objectMatrix, sceneMatrix, Calib3d.RANSAC, 3); MatobjectCorners =$
$newMat(4, 1, CvType.CV_32FC2); MatsceneCorners =$
$newMat(4, 1, CvType.CV_32FC2); objectCorners.put(0, 0, newdouble[]0, 0); objectCorners.put(1, 0, newdouble[]objectIm$
$objectImage.cols(); Imgproc.line(output, newPoint(sceneCorners.get(0, 0)[0]+$
$offset, sceneCorners.get(0, 0)[1]), newPoint(sceneCorners.get(1, 0)[0]+$
$offset, sceneCorners.get(1, 0)[1]), RECT_SCALAR, 4); Imgproc.line(output, newPoint(sceneCorners.get(1, 0)[0]+$
$offset, sceneCorners.get(1, 0)[1]), newPoint(sceneCorners.get(2, 0)[0]+$
$offset, sceneCorners.get(2, 0)[1]), RECT_SCALAR, 4); Imgproc.line(output, newPoint(sceneCorners.get(2, 0)[0]+$
$offset, sceneCorners.get(2, 0)[1]), newPoint(sceneCorners.get(3, 0)[0]+$
$offset, sceneCorners.get(3, 0)[1]), RECT_SCALAR, 4); Imgproc.line(output, newPoint(sceneCorners.get(3, 0)[0]+$
$offset, sceneCorners.get(3, 0)[1]), newPoint(sceneCorners.get(0, 0)[0]+$
$offset, sceneCorners.get(0, 0)[1]), RECT_SCALAR, 4); returnoutput;$
*SiftManager*
*Sif-*