

mosaic-library

This is a python library for manipulating images for underwater mosaicking applications. Currently, this library is capable of:

1. Reading input videos with the VideoPlayer class
2. Applying color, contrast, and lighting balancing as well as detail enhancement.
3. Other image manipulations such as resizing for faster mosaicking.
4. Image registration, such as feature detection and description, matching, and transformation estimation in pure translation, similarity, affine, and perspective classes.
5. Infinite and finite homography transformations on imagery given extrinsic rotations and translations between the camera and a new “virtual” camera.
6. Basic time synchronisation between camera position and rotation in the world frame with the video.
7. Visualisation of the preprocessing, feature detection / registration, homographic transform, and mosaicking steps.
8. Basic mosaicking map generation, with minimal memory management support (tiling).

Future Roadmap:

1. Estimation of a ground plane given additional extrinsic information from the mobile camera platform.
2. Estimation and removal of vanishing line to reduce OutOfMemory errors due to excessive image warping.
3. Memory management through iterative updating of mosaic tiles according to a common mosaic coordinate system.

Getting Started

First, make sure you have installed either Python 3.8.10 or Python 3.9 for your system.

(Optional): install an Integrated Development Environment, for example: PyCharm Community.

Create and activate virtual environment (guide here for non-pycharm users).

Install the mosaic-library package: `python -m pip install mosaic-library` (internet connection required).

Preprocessing Images

Images are essentially 3-Dimensional Tensors (Videos are 4-Dimensional), and so typical linear algebraic operations (matrix multiplication, inversion, *etc.*) will work on them. Images can appear faded or have low-contrast due to underwater effects. Histogramming techniques can balance the color intensity between

channels (color fixing), balance the overall light intensity of the image, or balance the contrast of the image. Additionally, a sharpness filter `cv2.detailEnhance` has been exposed in `mosaicking.preprocessing.enhance_detail`

Example Color Correction

```
from mosaicking import preprocessing
from skimage.data import astronaut

img = astronaut() # Load a good image
img = img[:, :, [2, 1, 0]] # Convert to OpenCV BGR color channel ordering (not needed)
img[:, :, -1] = 0.4 * img[:, :, -1] # Reduce the Red channel to make the image seem underwater

color_balance = preprocessing.fix_color(img, 0.9) # Apply Color balancing
# Display the comparison
preprocessing.cv2.imshow("astronaut (push any key to quit)",
                        preprocessing.np.concatenate(
                            (img, color_balance), axis=1))
preprocessing.cv2.waitKey()
# Cleanup
preprocessing.cv2.destroyAllWindows("astronaut (push any key to quit)")
```

Transforming Images

Applying geometric transformations to images is relatively easy. A rotation matrix, R , that specifies the roll, pitch, and yaw angles can rotate an image about its optical center in sequence to appear as if looked at from an equivalent camera from a different orientation. This is useful for projecting an image onto a desired plane (such as a ground plane). Additionally, translation transformations can be applied to the image. This, in combination with rotation, can align images taken at different positions. In both situations it is essential to know the intrinsic properties of the camera in order to correct for changes in perspective. If we have some idea of where the camera is relative to the image, then we can warp the image by rotating it about the approximate camera position.

This example investigates and applies extrinsic and intrinsic transformations to the image. See if you can make the train tracks appear parallel. To exit, press ESC with either image window active.

Other questions:

Why do the rails appear to vanish to a single point?

What are we doing with the camera when we shift the perspective like this?

Mosaicking

Mosaicking is the process of finding correspondences between images and arranging them so that they create a larger image. Videos or sequences of images can be used to incrementally expand a mosaic by adding in new tiles with each new frame. The images are preprocessed first to increase clarity and sharpness (making it easier to find features), then a transformation is computed between the existing mosaic and the new image that maximises correlation between the matched features. Then similarity, affine or perspective transformations (similar to the train example) are applied to the image. These steps are iterated over each new image, and the mosaic grows.

The mosaicking script has been exposed as a command (make sure you have it added to your path) as well as through the module.

To see the usage:

```
python3 -m mosaicking.mosaic -h
```

```
usage: mosaic.py [-h] [--output_directory OUTPUT_DIRECTORY] [--start_time START_TIME | --start_frame START_FRAME]
                [--finish_time FINISH_TIME | --finish_frame FINISH_FRAME] [--frame_skip FRAME_SKIP]
                [--sync_points SYNC_POINTS SYNC_POINTS | --time_offset TIME_OFFSET] [--min_warp_size MIN_WARP_SIZE]
                [--max_warp_size MAX_WARP_SIZE [MAX_WARP_SIZE ...]] [--max_mosaic_size MAX_MOSAIC_SIZE]
                [--alpha ALPHA] [--show_rotation] [--show_mosaic] [--show_preprocessing] [-k INTRINSIC INTRINSIC INTRINSIC INTRINSIC INTRINSIC INTRINSIC INTRINSIC]
                [-y YROTATION] [-z ZROTATION] [-g GRADIENTCLIP] [-f] [--homography {rigid,similarity,perspective}]
                [--features {ORB,SIFT,SURF,BRISK,KAZE,ALL}] [{ORB,SIFT,SURF,BRISK,KAZE,ALL}]
                video
```

positional arguments:

video	Path to video file.
-------	---------------------

optional arguments:

-h, --help	show this help message and exit
--output_directory OUTPUT_DIRECTORY	Path to directory where output mosaics are to be saved. Default is /tmp
--start_time START_TIME	Time (secs) to start from.
--start_frame START_FRAME	Frame number to start from.
--finish_time FINISH_TIME	Time (secs) to finish at.
--finish_frame FINISH_FRAME	Frame number to finish at.
--frame_skip FRAME_SKIP	Number of frames to skip between each mosaic update.
--orientation_file ORIENTATION_FILE	

Path to .csv file containing orientation measurements that transform

--sync_points SYNC_POINTS SYNC_POINTS
Time points (sec) where video and orientation file are in sync, used in orientation file.

--time_offset TIME_OFFSET
Time offset (sec) between video and orientation file timestamps, used in orientation file.

--min_matches MIN_MATCHES
Minimum number of matches to proceed with registration.

--min_features MIN_FEATURES
Minimum number of features to detect in an image.

--max_warp_size MAX_WARP_SIZE [MAX_WARP_SIZE ...]
Maximum size of warped image (used to prevent OOM errors), if 1 argument is given, height, width.

--max_mosaic_size MAX_MOSAIC_SIZE
Largest allowable size (width or height) for mosaic. Creates a new t

--save_freq SAVE_FREQ
Save frequency for output mosaic (if less than 1 then output saves a

--scale_factor SCALE_FACTOR
Scale the input image with constant aspect ratio.

--alpha ALPHA
Alpha blending scalar for merging new frames into mosaic.

--show_rotation
Flag to display the rotation compensation using rotation data.

--show_mosaic
Flag to display the mosaic output.

--show_preprocessing
Flag to display the preprocessed image

--fix_color
Flag to preprocess image for color balance.

--fix_contrast
Flag to preprocess image for contrast equalization.

--fix_light
Flag to preprocess image for lighting equalization.

-c CALIBRATION, --calibration CALIBRATION
Path to calibration file, overrides --intrinsic and --distortion.

-k INTRINSIC INTRINSIC INTRINSIC INTRINSIC INTRINSIC INTRINSIC INTRINSIC INTRINSIC
Space delimited list of intrinsic matrix terms, Read as K[0,0],K[0,1],K[0,2],K[1,0],K[1,1],K[1,2],K[2,0],K[2,1],K[2,2], calibration file if intrinsic present.

-d DISTORTION [DISTORTION ...], --distortion DISTORTION [DISTORTION ...]
Space delimited list of distortion coefficients, Read as K1, K2, p1, p2, p3, p4, p5, p6, p7, p8, p9, p10, p11, p12, p13, p14, p15, p16, p17, p18, p19, p20, p21, p22, p23, p24, p25, p26, p27, p28, p29, p30, p31, p32, p33, p34, p35, p36, p37, p38, p39, p40, p41, p42, p43, p44, p45, p46, p47, p48, p49, p50, p51, p52, p53, p54, p55, p56, p57, p58, p59, p60, p61, p62, p63, p64, p65, p66, p67, p68, p69, p70, p71, p72, p73, p74, p75, p76, p77, p78, p79, p80, p81, p82, p83, p84, p85, p86, p87, p88, p89, p90, p91, p92, p93, p94, p95, p96, p97, p98, p99, p100, p101, p102, p103, p104, p105, p106, p107, p108, p109, p110, p111, p112, p113, p114, p115, p116, p117, p118, p119, p120, p121, p122, p123, p124, p125, p126, p127, p128, p129, p130, p131, p132, p133, p134, p135, p136, p137, p138, p139, p140, p141, p142, p143, p144, p145, p146, p147, p148, p149, p150, p151, p152, p153, p154, p155, p156, p157, p158, p159, p160, p161, p162, p163, p164, p165, p166, p167, p168, p169, p170, p171, p172, p173, p174, p175, p176, p177, p178, p179, p180, p181, p182, p183, p184, p185, p186, p187, p188, p189, p190, p191, p192, p193, p194, p195, p196, p197, p198, p199, p200, p201, p202, p203, p204, p205, p206, p207, p208, p209, p210, p211, p212, p213, p214, p215, p216, p217, p218, p219, p220, p221, p222, p223, p224, p225, p226, p227, p228, p229, p230, p231, p232, p233, p234, p235, p236, p237, p238, p239, p240, p241, p242, p243, p244, p245, p246, p247, p248, p249, p250, p251, p252, p253, p254, p255, p256, p257, p258, p259, p260, p261, p262, p263, p264, p265, p266, p267, p268, p269, p270, p271, p272, p273, p274, p275, p276, p277, p278, p279, p280, p281, p282, p283, p284, p285, p286, p287, p288, p289, p290, p291, p292, p293, p294, p295, p296, p297, p298, p299, p300, p301, p302, p303, p304, p305, p306, p307, p308, p309, p310, p311, p312, p313, p314, p315, p316, p317, p318, p319, p320, p321, p322, p323, p324, p325, p326, p327, p328, p329, p330, p331, p332, p333, p334, p335, p336, p337, p338, p339, p340, p341, p342, p343, p344, p345, p346, p347, p348, p349, p350, p351, p352, p353, p354, p355, p356, p357, p358, p359, p360, p361, p362, p363, p364, p365, p366, p367, p368, p369, p370, p371, p372, p373, p374, p375, p376, p377, p378, p379, p380, p381, p382, p383, p384, p385, p386, p387, p388, p389, p390, p391, p392, p393, p394, p395, p396, p397, p398, p399, p400, p401, p402, p403, p404, p405, p406, p407, p408, p409, p410, p411, p412, p413, p414, p415, p416, p417, p418, p419, p420, p421, p422, p423, p424, p425, p426, p427, p428, p429, p430, p431, p432, p433, p434, p435, p436, p437, p438, p439, p440, p441, p442, p443, p444, p445, p446, p447, p448, p449, p450, p451, p452, p453, p454, p455, p456, p457, p458, p459, p460, p461, p462, p463, p464, p465, p466, p467, p468, p469, p470, p471, p472, p473, p474, p475, p476, p477, p478, p479, p480, p481, p482, p483, p484, p485, p486, p487, p488, p489, p490, p491, p492, p493, p494, p495, p496, p497, p498, p499, p500, p501, p502, p503, p504, p505, p506, p507, p508, p509, p510, p511, p512, p513, p514, p515, p516, p517, p518, p519, p520, p521, p522, p523, p524, p525, p526, p527, p528, p529, p530, p531, p532, p533, p534, p535, p536, p537, p538, p539, p540, p541, p542, p543, p544, p545, p546, p547, p548, p549, p550, p551, p552, p553, p554, p555, p556, p557, p558, p559, p560, p561, p562, p563, p564, p565, p566, p567, p568, p569, p570, p571, p572, p573, p574, p575, p576, p577, p578, p579, p580, p581, p582, p583, p584, p585, p586, p587, p588, p589, p590, p591, p592, p593, p594, p595, p596, p597, p598, p599, p600, p601, p602, p603, p604, p605, p606, p607, p608, p609, p610, p611, p612, p613, p614, p615, p616, p617, p618, p619, p620, p621, p622, p623, p624, p625, p626, p627, p628, p629, p630, p631, p632, p633, p634, p635, p636, p637, p638, p639, p640, p641, p642, p643, p644, p645, p646, p647, p648, p649, p650, p651, p652, p653, p654, p655, p656, p657, p658, p659, p660, p661, p662, p663, p664, p665, p666, p667, p668, p669, p670, p671, p672, p673, p674, p675, p676, p677, p678, p679, p680, p681, p682, p683, p684, p685, p686, p687, p688, p689, p690, p691, p692, p693, p694, p695, p696, p697, p698, p699, p700, p701, p702, p703, p704, p705, p706, p707, p708, p709, p710, p711, p712, p713, p714, p715, p716, p717, p718, p719, p720, p721, p722, p723, p724, p725, p726, p727, p728, p729, p730, p731, p732, p733, p734, p735, p736, p737, p738, p739, p740, p741, p742, p743, p744, p745, p746, p747, p748, p749, p750, p751, p752, p753, p754, p755, p756, p757, p758, p759, p760, p761, p762, p763, p764, p765, p766, p767, p768, p769, p770, p771, p772, p773, p774, p775, p776, p777, p778, p779, p780, p781, p782, p783, p784, p785, p786, p787, p788, p789, p790, p791, p792, p793, p794, p795, p796, p797, p798, p799, p800, p801, p802, p803, p804, p805, p806, p807, p808, p809, p810, p811, p812, p813, p814, p815, p816, p817, p818, p819, p820, p821, p822, p823, p824, p825, p826, p827, p828, p829, p830, p831, p832, p833, p834, p835, p836, p837, p838, p839, p840, p841, p842, p843, p844, p845, p846, p847, p848, p849, p850, p851, p852, p853, p854, p855, p856, p857, p858, p859, p860, p861, p862, p863, p864, p865, p866, p867, p868, p869, p870, p871, p872, p873, p874, p875, p876, p877, p878, p879, p880, p881, p882, p883, p884, p885, p886, p887, p888, p889, p890, p891, p892, p893, p894, p895, p896, p897, p898, p899, p900, p901, p902, p903, p904, p905, p906, p907, p908, p909, p910, p911, p912, p913, p914, p915, p916, p917, p918, p919, p920, p921, p922, p923, p924, p925, p926, p927, p928, p929, p930, p931, p932, p933, p934, p935, p936, p937, p938, p939, p940, p941, p942, p943, p944, p945, p946, p947, p948, p949, p950, p951, p952, p953, p954, p955, p956, p957, p958, p959, p960, p961, p962, p963, p964, p965, p966, p967, p968, p969, p970, p971, p972, p973, p974, p975, p976, p977, p978, p979, p980, p981, p982, p983, p984, p985, p986, p987, p988, p989, p990, p991, p992, p993, p994, p995, p996, p997, p998, p999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063, 1064, 1065, 1066, 1067, 1068, 1069, 1070, 1071, 1072, 1073, 1074, 1075, 1076, 1077, 1078, 1079, 1080, 1081, 1082, 1083, 1084, 1085, 1086, 1087, 1088, 1089, 1090, 1091, 1092, 1093, 1094, 1095, 1096, 1097, 1098, 1099, 1100, 1101, 1102, 1103, 1104, 1105, 1106, 1107, 1108, 1109, 1110, 1111, 1112, 1113, 1114, 1115, 1116, 1117, 1118, 1119, 1120, 1121, 1122, 1123, 1124, 1125, 1126, 1127, 1128, 1129, 1130, 1131, 1132, 1133, 1134, 1135, 1136, 1137, 1138, 1139, 1140, 1141, 1142, 1143, 1144, 1145, 1146, 1147, 1148, 1149, 1150, 1151, 1152, 1153, 1154, 1155, 1156, 1157, 1158, 1159, 1160, 1161, 1162, 1163, 1164, 1165, 1166, 1167, 1168, 1169, 1170, 1171, 1172, 1173, 1174, 1175, 1176, 1177, 1178, 1179, 1180, 1181, 1182, 1183, 1184, 1185, 1186, 1187, 1188, 1189, 1190, 1191, 1192, 1193, 1194, 1195, 1196, 1197, 1198, 1199, 1200, 1201, 1202, 1203, 1204, 1205, 1206, 1207, 1208, 1209, 1210, 1211, 1212, 1213, 1214, 1215, 1216, 1217, 1218, 1219, 1220, 1221, 1222, 1223, 1224, 1225, 1226, 1227, 1228, 1229, 1230, 1231, 1232, 1233, 1234, 1235, 1236, 1237, 1238, 1239, 1240, 1241, 1242, 1243, 1244, 1245, 1246, 1247, 1248, 1249, 1250, 1251, 1252, 1253, 1254, 1255, 1256, 1257, 1258, 1259, 1260, 1261, 1262, 1263, 1264, 1265, 1266, 1267, 1268, 1269, 1270, 1271, 1272, 1273, 1274, 1275, 1276, 1277, 1278, 1279, 1280, 1281, 1282, 1283, 1284, 1285, 1286, 1287, 1288, 1289, 1290, 1291, 1292, 1293, 1294, 1295, 1296, 1297, 1298, 1299, 1300, 1301, 1302, 1303, 1304, 1305, 1306, 1307, 1308, 1309, 1310, 1311, 1312, 1313, 1314, 1315, 1316, 1317, 1318, 1319, 1320, 1321, 1322, 1323, 1324, 1325, 1326, 1327, 1328, 1329, 1330, 1331, 1332, 1333, 1334, 1335, 1336, 1337, 1338, 1339, 1340, 1341, 1342, 1343, 1344, 1345, 1346, 1347, 1348, 1349, 1350, 1351, 1352, 1353, 1354, 1355, 1356, 1357, 1358, 1359, 1360, 1361, 1362, 1363, 1364, 1365, 1366, 1367, 1368, 1369, 1370, 1371, 1372, 1373, 1374, 1375, 1376, 1377, 1378, 1379, 1380, 1381, 1382, 1383, 1384, 1385, 1386, 1387, 1388, 1389, 1390, 1391, 1392, 1393, 1394, 1395, 1396, 1397, 1398, 1399, 1400, 1401, 1402, 1403, 1404, 1405, 1406, 1407, 1408, 1409, 1410, 1411, 1412, 1413, 1414, 1415, 1416, 1417, 1418, 1419, 1420, 1421, 1422, 1423, 1424, 1425, 1426, 1427, 1428, 1429, 1430, 1431, 1432, 1433, 1434, 1435, 1436, 1437, 1438, 1439, 1440, 1441, 1442, 1443, 1444, 1445, 1446, 1447, 1448, 1449, 1450, 1451, 1452, 1453, 1454, 1455, 1456, 1457, 1458, 1459, 1460, 1461, 1462, 1463, 1464, 1465, 1466, 1467, 1468, 1469, 1470, 1471, 1472, 1473, 1474, 1475, 1476, 1477, 1478, 1479, 1480, 1481, 1482, 1483, 1484, 1485, 1486, 1487, 1488, 1489, 1490, 1491, 1492, 1493, 1494, 1495, 1496, 1497, 1498, 1499, 1500, 1501, 1502, 1503, 1504, 1505, 1506, 1507, 1508, 1509, 1510, 1511, 1512, 1513, 1514, 1515, 1516, 1517, 1518, 1519, 1520, 1521, 1522, 1523, 1524, 1525, 1526, 1527, 1528, 1529, 1530, 1531, 1532, 1533, 1534, 1535, 1536, 1537, 1538, 1539, 1540, 1541, 1542, 1543, 1544, 1545, 1546, 1547, 1548, 1549, 1550, 1551, 1552, 1553, 1554, 1555, 1556, 1557, 1558, 1559, 1560, 1561, 1562, 1563, 1564, 1565, 1566, 1567, 1568, 1569, 1570, 1571, 1572, 1573, 1574, 1575, 1576, 1577, 1578, 1579, 1580, 1581, 1582, 1583, 1584, 1585, 1586, 1587, 1588, 1589, 1590, 1591, 1592, 1593, 1594, 1595, 1596, 1597, 1598, 1599, 1600, 1601, 1602, 1603, 1604, 1605, 1606, 1607, 1608, 1609, 1610, 1611, 1612, 1613, 1614, 1615, 1616, 1617, 1618, 1619, 1620, 1621, 1622, 1623, 1624, 1625, 1626, 1627, 1628, 1629, 1630, 1631, 1632, 1633, 1634, 1635, 1636, 1637, 1638, 1639, 1640, 1641, 1642, 1643, 1644, 1645, 1646, 1647, 1648, 1649, 1650, 1651, 1652, 1653, 1654, 1655, 1656, 1657, 1658, 1659, 1660, 1661, 1662, 1663, 1664, 1665, 1666, 1667, 1668, 1669, 1670, 1671, 1672, 1673, 1674, 1675, 1676, 1677, 1678, 1679, 1680, 1681, 1682, 1683, 1684, 1685, 1686, 1687, 1688, 1689, 1690, 1691, 1692, 1693, 1694, 1695, 1696, 1697, 1698, 1699, 1700, 1701, 1702, 1703, 1704, 1705, 1706, 1707, 1708, 1709, 1710, 1711, 1712, 1713, 1714, 1715, 1716, 1717, 1718, 1719, 1720, 1721, 1722, 1723, 1724, 1725, 1726, 1727, 1728, 1729, 1730, 1731, 1732, 1733, 1734, 1735, 1736, 1737, 1738, 1739, 1740, 1741, 1742, 1743, 1744, 1745, 1746, 1747, 1748, 1749, 1750, 1751, 1752, 1753, 1754, 1755, 1756, 1757, 1758, 1759, 1760, 1761, 1762, 1763, 1764, 1765, 1766, 1767, 1768, 1769, 1770, 1771, 1772, 1773, 1774, 1775, 1776, 1777, 1778, 1779, 1780, 1781, 1782, 1783, 1784, 1785, 1786, 1787, 1788, 1789, 1790, 1791, 1792, 1793, 1794, 1795, 1796, 1797, 1798, 1799, 1800, 1801, 1802, 1803, 1804, 1805, 1806, 1807, 1808, 1809, 1810, 1811, 1812, 1813, 1814, 1815, 1816, 1817, 1818, 1819, 1820, 1821, 1822, 1823, 1824, 1825, 1826, 1827, 1828, 1829, 1830, 1831, 1832, 1833, 1834, 1835, 1836, 1837, 1838, 1839, 1840, 1841, 1842, 1843, 1844, 1845, 1846, 1847, 1848, 1849, 1850, 1851, 1852, 1853, 1854, 1855, 1856, 1857, 1858, 1859, 1860, 1861, 1862, 1863, 1864, 1865, 1866, 1867, 1868, 1869, 1870, 1871, 1872, 1873, 1874, 1875, 1876, 1877, 1878, 1879, 1880, 1881, 1882, 1883, 1884, 1885, 1886, 1887, 1888, 1889, 1890, 1891, 1892, 1893, 1894, 1895, 1896, 1897, 1898, 1899, 1900, 1901, 1902, 1903, 1904, 1905, 1906, 1907, 1908, 1909, 1910, 1911, 1912, 1913, 1914, 1915, 1916, 1917, 1918, 1919, 1920, 1921, 1922, 1923, 1924, 1925, 1926, 1927, 1928, 1929, 1930, 1931, 1932, 1933, 1934, 1935, 1936, 1937, 1938, 1939, 1940, 1941, 1942, 1943, 1944, 1945, 1946, 1947, 1948, 1949, 1950, 1951, 1952, 1953, 1954, 1955, 1956, 1957, 1958, 1959, 1960, 1961, 1962, 1963, 1964, 1965, 1966, 1967, 1968, 1969, 1970, 1971, 1972, 1973, 1974, 1975, 1976, 1977, 1978, 1979, 1980, 1981, 1982, 1983, 1984, 1985, 1986, 1987, 1988, 1989, 1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026, 2027, 2028, 2029, 2030, 2031, 2032, 2033, 2034, 2035, 2036, 2037, 2038, 2039, 2040, 2041, 2042, 2043, 2044, 2045, 2046, 2047, 2048, 2049, 2050, 2051, 2052, 2053, 2054, 2055, 2056, 2057, 2058, 2059, 2060, 2061, 2062, 2063, 2064, 2065, 2066, 2067, 2068, 2069, 2070, 2071, 2072, 2073, 2074, 2075, 2076, 2077, 2078, 2079, 2080, 2081, 2082, 2083, 2084, 2085, 2086, 2087, 2088, 2089, 2090, 2091, 2092, 2093, 2094, 2095, 2096, 2097, 2098, 2099, 2100, 2101, 2102, 2103, 2104, 2105, 2106, 2107, 2108, 2109, 2110, 2111, 2112, 2113, 2114, 2115, 2116, 2117, 2118, 2119, 2120, 2121, 2122, 2123, 2124, 2125, 2126, 2127, 2128, 2129, 2130, 2131, 2132, 2133, 2134, 2135, 2136, 2137, 2138, 2139, 2140, 2141, 2142, 2143, 2144, 2145, 2146, 2147, 2148, 2149, 2150, 2151, 2

<code>--show_matches</code>	Display the matches.
<code>--inliers_roi</code>	Only allow the convex hull of the inlier points to be displayed.
<code>--demo</code>	Creates a video of the mosaic creation process. For demo purposes only.
<code>--show_demo</code>	Display the demo while underway.

Try running the `mosaicking.mosaic` script on a video you've taken with your phone or otherwise.

Do any parts of the mosaic appear poorer in quality than others? If so, why?

It is important the camera's view is mostly occupied by a scene that lies on a plane (like the seabottom). Control of the camera to keep the ground plane in sight is particularly important as the new image will become smaller and smaller with respect to the mosaic as the camera moves "into" or "through" the mosaic. Eventually the number of common features between the mosaic and the image will diminish to the point where no good transformation can be found. Or the camera may view something that does not have enough features to estimate a transformation. Or, the mosaic size might get too big. The current implementation simply just dumps the mosaic to tiles (called `tile_001.png`, `tmp_002.png` etc.) and starts a new mosaic on the latest image if the memory consumption goes above a threshold.

What could be a way to reduce this diminishing matching features problem?

Activities

1. Create your own video and see how large a mosaic you can create (substitute your video path in for the one given in the instructions).
2. What kind of camera control and survey works best?
3. Apart from color, contrast and lighting transforms, are there any geometric transformations that we could apply to get better results?