Camera Mid Term Report. — Ivan Ouyang

Data Buffer

CRITERIA	MEETS SPECIFICATIONS
MP.1 Data Buffer Optimization	Implement a vector for dataBuffer objects whose size does not exceed a limit (e.g. 2 elements). This can be achieved by pushing in new elements on one end and removing elements on the other end.

MP. 1

Implementation:
Check to see if the size of *dataBuffer* equals to the *dataBufferSize.* If so, then erase the first frame of the image because the next image will be pushed into the buffer later.

Keypoints

CRITERIA	MEETS SPECIFICATIONS
MP.2 Keypoint Detection	Implement detectors HARRIS, FAST, BRISK, ORB, AKAZE, and SIFT and make them selectable by setting a string accordingly.
MP.3 Keypoint Removal	Remove all keypoints outside of a pre-defined rectangle and only use the keypoints within the rectangle for further processing.

MP. 2 Keypiont Detection

Implementation:

Fill in the detKeypointsModern function. Take *keypoints, imgGray, detectorType, false* as input parameters. By calling different detectorType, the function will process the imgGray by making use of different detector function call. Visualize the results at the end of the function.

MP. 3 Keypiont Removal

Implementation:
Use a *for* loop to go through all the keypoints. By making use of the *vehicleRect,* filter out all the key points that are not reside in this rectangle. The key points left over are the key points we are looking for.

Descriptors

CRITERIA	MEETS SPECIFICATIONS
MP.4 Keypoint Descriptors	Implement descriptors BRIEF, ORB, FREAK, AKAZE and SIFT and make them selectable by setting a string accordingly.
MP.5 Descriptor Matching	Implement FLANN matching as well as k-nearest neighbor selection. Both methods must be selectable using the respective strings in the main function.
MP.6 Descriptor Distance Ratio	Use the K-Nearest-Neighbor matching to implement the descriptor distance ratio test, which looks at the ratio of best vs. second-best match to decide whether to keep an associated pair of keypoints.

MP.4 Keypoint Descriptors

Implementation:
Similar to MP.2. We can get the *Keypoints* from Step3. By comparing the *descriptorType*, different descriptor pattern can be chosen. These patten will be applied to the *Keypoints,* the output will be descriptor, ie, a vector of pixels of the image.

MP.5

Impelementation:
As SIFT is not similar to other binary descriptor(such as BRIEF, ORB, BRISK, FREAK), by making use of Euclidean distance to compare and match the descriptor will have a performance boost. So a L-2 norm method is used here.
FLANN is short for Fast Library for Approximate Nearest Neighbors. It works way much faster than the Brutal Force matcher for large dataset.

MP. 6

Implementation:
FLANN
According to the paper on “SIFT”, Lowe’s ratio 0.7 is selected while KNN is used.

Performance

CRITERIA	MEETS SPECIFICATIONS
MP.7 Performance Evaluation 1	Count the number of keypoints on the preceding vehicle for all 10 images and take note of the distribution of their neighborhood size. Do this for all the detectors you have implemented.
MP.8 Performance Evaluation 2	Count the number of matched keypoints for all 10 images using all possible combinations of detectors and descriptors. In the matching step, the BF approach is used with the descriptor distance ratio set to 0.8.

MP.9 Performance Evaluation 3	Log the time it takes for keypoint detection and descriptor extraction. The results must be entered into a spreadsheet and based on this data, the TOP3 detector / descriptor combinations must be recommended as the best choice for our purpose of detecting keypoints on vehicles.

MP.7

Implementation:

FRAME #	HARRIS	FAST	BRISK	ORB	AKAZE	SIFT
0	50	420	152	91	162	137
1	54	431	159	102	157	131
2	53	408	154	106	159	121
3	55	430	161	113	154	135
4	56	389	155	109	162	134
5	58	417	154	124	163	139
6	57	422	160	129	173	136
7	61	413	150	127	175	147
8	60	401	155	124	175	156
9	57	405	141	125	175	135

MP. 8

Implementation:

Detector: ShiTomasi

Descriptors/ Frame#	BRISK	ORB	FREAK	AKAZE	SIFT
0	1370	1370	1370	NA	1370
1	1301	1301	1301	NA	1301
2	1361	1361	1361	NA	1361
3	1358	1358	1358	NA	1358
4	1333	1333	1333	NA	1333
5	1284	1284	1284	NA	1284
6	1322	1322	1322	NA	1322
7	1366	1366	1366	NA	1366
8	1389	1389	1389	NA	1389
9	1339	1339	1339	NA	1339

Detector: HARRIS

Descriptors/ Frame#	BRISK	ORB	FREAK	AKAZE	SIFT
0	492	492	492	NA	492
1	502	502	502	NA	502
2	516	516	516	NA	516
3	524	524	524	NA	524
4	523	523	523	NA	524
5	511	511	511	NA	511
6	505	505	505	NA	505
7	510	510	510	NA	510
8	529	529	529	NA	529
9	520	520	520	NA	515

Detector: FAST

Descriptors/ Frame#	BRISK	ORB	FREAK	AKAZE	SIFT
0	5063	5063	5063	NA	5063
1	4952	4952	4952	NA	4952
2	4863	4863	4863	NA	4863
3	4840	4840	4840	NA	4840
4	4856	4856	4856	NA	4856
5	4899	4899	4899	NA	4899
6	4870	4870	4870	NA	4870
7	4868	4868	4868	NA	4868
8	4996	4996	4996	NA	4996
9	4997	4997	4997	NA	4997

Detector: ORB

Descriptors/ Frame#	BRISK	ORB	FREAK	AKAZE	SIFT
0	500	500	500	NA	500
1	500	500	500	NA	500
2	500	500	500	NA	500
3	500	500	500	NA	500
4	500	500	500	NA	500
5	500	500	500	NA	500
6	500	500	500	NA	500
7	500	500	500	NA	500
8	500	500	500	NA	500
9	500	500	500	NA	500

Detector: AKAZE

Descriptors/ Frame#	BRISK	ORB	FREAK	AKAZE	SIFT
0	1351	1351	1351	1351	1351
1	1327	1327	1327	1327	1327
2	1331	1331	1331	1331	1331
3	1351	1351	1351	1351	1351
4	1360	1360	1360	1360	1360
5	1347	1347	1347	1347	1347
6	1363	1363	1363	1363	1363
7	1331	1331	1331	1331	1331
8	1357	1357	1357	1357	1357
9	1331	1331	1331	1331	1331

Detector: SIFT

Descriptors/ Frame#	BRISK	ORB	FREAK	AKAZE	SIFT
0	1438	NA	1438	NA	1438
1	1371	NA	1371	NA	1371
2	1380	NA	1380	NA	1380
3	1335	NA	1335	NA	1335
4	1305	NA	1305	NA	1305
5	1370	NA	1370	NA	1370
6	1396	NA	1396	NA	1396
7	1382	NA	1382	NA	1382
8	1463	NA	1463	NA	1463
9	1422	NA	1422	NA	1422

MP. 9

TIME OF DETECTOR

	BRISK	ORB	FREAK	AKAZE	SIFT
SHITOMASI	174.41 ms	148.913 ms	121.59 ms	NA	135.619 ms
HARRIS	156.087 ms	138.279 ms	124.323 ms	NA	128.104 ms
FAST	25.6386 ms	21.3233 ms	22.1093 ms	NA	20.8753 ms
ORB	84.8292 ms	83.7626 ms	78.5994 ms	NA	86.9558 ms
AKAZE	838.481 ms	838.432 ms	757.233 ms	798.701 ms	806.566 ms
SIFT	1128.08 ms	NA	1336.64 ms	NA	1071.21 ms

TIME OF DESCRIPTOR

	BRISK	ORB	FREAK	AKAZE	SIFT
SHITOMASI	29.35 ms	8.9328 ms	385.683 ms	NA	160.824 ms
HARRIS	14.8603 ms	7.98811 ms	380.884 ms	NA	153.596 ms
FAST	54.8574 ms	27.3408 ms	448.119 ms	NA	385.714 ms
ORB	15.8505 ms	50.1791 ms	415.348 ms	NA	501.998 ms
AKAZE	26.9118 ms	32.5048 ms	439.817 ms	710.898 ms	263.381 ms
SIFT	17.6061 ms	NA	434.3 ms	NA	860.683 ms

Top 3 best Performer:
FAST/ORB
FAST/BRISK
ORB/BRISK