Camera Based 2D Feature Tracking
Mid -Term Report

Data Buffer

MP.1 Data Buffer Optimization

Satisfied by using the RingBuffer class (RingBuffer.h) instead of a vector of DataFrame. The RingBuffer class fills a vector up to the limit of 'dataBufferSize'. Then each new addition is added at the end of the vector after existing members have been shuffled down by 1. The class could be made more efficient by the use of an 'inptr' and its own iterator (so the existing member shuffle is not required when a new image is added).

KeyPoints

MP.2 Keypoint Detection

Implemented the 'detKeypointsHarris' fn for the Harris keypoint detector and the 'detKeypointsModern' for the FAST, BRISK, ORB, AKAZE, FREAK and SIFT detectors, selectable using the relevant detectorType string from the parent fn. Note: Although the code builds with the FREAK detector implementation, running this detector results in a 'Feature Not Implemented' exception from opencv and a core dump.

MP.3 Keypoint Removal

Implemented a keypoint removal filter based upon the rectangle supplied (in MidTermProject_Camera_Student.cpp). Note: Tried a narrower rectangle (550, 180, 150, 150) to remove more points just outside the vehicle boundary.

Descriptors

MP.4 Keypoint Descriptors

Implemented the student part of the 'descKeypoints' function. Added descriptor extractors for BRIEF, ORB, FREAK, AKAZE and SIFT, selectable by the appropriate descriptorType string.

MP.5 Descriptor Matching

Implemented the FLANN matcher (in matchDescriptors) and the knn selection (in matchDescriptors), both selectable by the appropriate selection strings.

MP.6 Descriptor Distance Ratio

Added the knn descriptor matching ratio filter as implemented in the preceding lesson segment.

Performance

MP.7 Performance Evaluation 1

No Of KeyPoints found

Image #	1	2	3	4	5	6	7	8	9	10
HARRIS	34	29	40	37	50	151	26	67	59	87
FAST	141	143	140	149	139	139	153	142	131	135
BRISK	254	274	276	275	293	275	289	268	258	249
ORB	87	101	105	110	106	121	128	120	117	117
AKAZE	162	157	159	154	162	163	173	175	175	175
SIFT	137	131	121	135	134	139	136	147	156	135

MP.8 Performance Evaluation 2 & MP.9 Performance Evaluation 3

To do what required here I changed/adapted the code in MidTermProject_Camera_Student.cpp with a compilation switch (RUN_AS_PERFORMANCE_EVALUATION) so I could collect the results into a results file (Results.dat) in csv format, which could then be loaded into a spreadsheet.

Note: There are several combinations of detector/descriptor extractor I couldn't get to work, especially the SIFT descriptor extractor. I don't know why. Each time I get an open CV exception in 'batchdistance' concerning the 'dType'. I've tried converting the descriptor cv::Mat objects to 32F, but this didn't fix it.

(I think I also got an 'out of memory' exception when I tried the SIFT/SIFT combination).

The results spreadsheet is 'SFND_2D_Feature_Matching_Results.xlsx'

Detector Type	Descriptor Type		Frame # (2 - 10) Match Counts										Frame # (1 - 10) Detector Times										Frame # (1 - 10) Descriptor Times									
			2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
HARRIS	BRISK	Match Counts	21	18	24	24	25	46	16	45	42	26.4265	19.4855	20.6214	20.5227	21.5093	100.662	17.8418	25.3602	21.4686	41.2401	388.312	393.543	402.427	395.481	391.22	393.683	391.745	390.245	389.62	399.569
HARRIS	BRIEF	Match Counts	20	19	29	27	22	60	18	49	43	22.9335	20.8717	19.0927	18.8799	21.1454	104.03	18.5352	24.6499	21.0171	39.7488	1.70965	1.48959	1.26106	1.15391	1.2451	2.17883	1.47583	1.35719	1.22923	0.607423
HARRIS	ORB	Match Counts	19	16	22	23	25	51	16	46	42	21.7767	20.6188	19.8919	19.8438	21.3715	102.971	14.0346	25.1637	21.799	42.0682	1.63524	0.860081	0.852141	1.06966	1.05951	1.09034	0.847471	1.09308	1.06661	1.21079
FAST	BRISK	Match Counts	92	97	96	96	79	104	104	96	94	0.920302	0.883776	0.936697	0.909605	0.976269	0.877127	0.900786	0.865285	0.908884	0.890064	397.61	387.703	388.069	387.637	387.836	389.292	388.5	387.793	388.35	388.841
FAST	BRIEF	Match Counts	112	121	110	123	99	116	128	120	113	0.920603	0.90603	1.3381	1.34818	1.15925	0.855093	0.865773	0.848976	1.09804	0.88253	0.766127	0.765718	1.2434	1.36753	0.717621	0.715347	0.786386	0.714429	0.689543	0.695748
FAST	ORB	Match Counts	110	116	108	121	98	115	118	117	112	0.892868	0.869947	1.2848	0.878511	1.15466	1.10768	0.866707	0.85219	0.862485	0.892727	1.03965	0.995128	1.68226	0.967729	1.35334	0.980265	1.00993	0.979022	0.962773	0.970921
BRISK	BRISK	Match Counts	168	169	157	170	171	186	174	167	184	447.687	433.675	438.533	434.291	433.055	435.215	432.615	431.897	432.412	433.514	388.444	392.794	389.712	390.345	389.515	388.096	389.751	391.36	389.086	389.139
BRISK	BRIEF	Match Counts	174	195	182	177	182	193	208	186	179	442.009	432.731	435.133	431.738	432.131	432.409	436.101	433.03	432.787	433.051	1.09132	1.07618	1.09139	1.08493	1.15951	1.18042	1.12245	1.07487	1.0425	1.01697
BRISK	ORB	Match Counts	155	167	155	161	156	181	165	170	170	445.612	437.279	432.176	432.731	435.068	433.264	431.561	433.204	433.682	433.453	4.77172	4.7252	4.79591	4.7734	4.8416	4.79149	5.29121	4.74871	4.74165	4.79223
ORB	BRISK	Match Counts	72	73	78	83	78	89	88	83	84	9.24232	7.57955	7.47849	7.41756	7.40568	7.48921	7.47153	7.51455	7.70955	7.58853	397.042	387.493	386.9	388.945	386.169	388.113	387.073	387.173	387.431	386.585
ORB	BRIEF	Match Counts	46	42	45	59	53	73	68	81	63	8.68113	10.4817	8.90048	7.37985	7.34494	7.47057	7.52551	7.44067	8.1313	7.46487	0.547019	0.975743	0.638869	0.662961	0.539083	0.601826	0.61636	0.610829	0.568247	0.576016
ORB	ORB	Match Counts	63	69	71	82	88	96	90	88	84	8.51295	11.3693	7.46541	7.40991	7.43904	7.74887	7.35237	7.47122	7.50324	7.53403	4.79979	7.76056	4.76777	4.88771	4.7936	4.80443	4.78921	4.77113	4.76108	4.77556
AKAZE	BRISK	Match Counts	134	124	129	128	130	132	142	144	141	123.68	110.764	112.605	111.449	114.081	113.332	112.841	116.317	113.624		387.648	387.407	393.097	389.919	388.964	390.807	389.317	387.906	387.889	388.847
AKAZE	BRIEF	Match Counts	137	133	130	130	134	146	150	147	150	121.705	112.86	111.565	112.941	112.817	112.974	111.461	113.353	111.567	113.55	1.65746	1.78785	0.848078	1.75461	0.845784	1.84864	1.99927	1.79856	2.0493	2.07124
AKAZE	ORB	Match Counts	128	128	126	118	129	131	136	136	143	124.516	110.758	110.331	114.089	113.357	115.85	114.317	113.457	113.127	111.275	4.60983	4.11273	3.6906	4.1312	3.94822	3.71214	4.25159	4.22764	4.2027	4.85138
AKAZE	AKAZE	Match Counts	135	138	132	126	128	146	147	149	148	123.151	113.734	103.157	103.028	102.202	106.859	113.947	102.642	99.9732	113.345	97.2403	92.1554	89.8096	89.0183	89.38	91.071	100.82	90.8939	89.8914	102.349
SIFT	BRISK	Match Counts	63	64	60	65	59	65	64	67	79	185.079	158.154	161.883	159.562	156.885	158.982	156.848	159.489	158.786	159.984	388.041	389.867	387.96	387.9	387.46	387.74	387.161	388.131	387.507	388.365
SIFT	BRIEF	Match Counts	86	76	72	83	69	75	76	69	87	169.653	161.778	154.881	161.109	156.366	162.54	158.979	155.591	162.076	159.062	0.801246	0.769615	0.766594	0.789639	0.813583	0.780804	0.779952	0.836889	0.844463	0.945673

From the results match counts, detector & descriptor times the best REAL TIME detector/descriptor extraction combinations are:

Rank # (1 = best)	Detector	Descriptor Extractor
1	FAST	BRIEF
2	FAST	ORB
3	ORB	BRIEF
4	ORB	ORB

From the point of view of most points matched the BRISK and AKAZE detectors paired with any of the descriptor extractors appear to be best performing, although these will not work in a real time environment where more than 1-2 fps are required.