Camera Based 2D Feature Tracking

Mid term project Writeup

Data Buffer

MP.1 Data Buffer Optimization

Data buffer for storing the images of the input image data set is implemented as a ring buffer (using std::vector data structure) with a configurable maximum amount of elements stored inside. After reaching the maximum size of the buffer, the 'oldest' stored element is deleted from begin of the buffer before the new element is appended to the end, so that the maximum size of the buffer always remains the same. Considering the requirements of this mid-term project in the context of features tracking through sequential images, the maximum size of the data buffer was set to 2 elements.

Keypoints

MP.2 Keypoint Detection

Following keypoint detectors were implemented in */src/matching2D_Student.cpp*:

- **Harris**. Implemented in `detKeypointsHarris` function using `cv::cornerHarris()` function from OpenCV library with further filtering of detected keypoints by intensity threshold and non-maximum suppression of closely located neighboring keypoints.
- **Fast.** Implemented in `initDetector` function using `cv::Ptr<cv::FeatureDetector> cv::FastFeatureDetector::create()` from OpenCV library.
- **Brisk.** Implemented in `initDetector` function using `cv::Ptr<cv::FeatureDetector> cv::BRISK::create()` from OpenCV library.
- **ORB.** Implemented in `initDetector` function using `cv::Ptr<cv::FeatureDetector> cv::ORB::create()` from OpenCV library.
- **AKAZE.** Implemented in `initDetector` function using `cv::Ptr<cv::FeatureDetector> cv::AKAZE::create()` from OpenCV library.
- **SIFT.** Implemented in `initDetector` function using `cv::Ptr<cv::FeatureDetector> cv::SIFT::create()` from OpenCV library.

Selection of required keypoint detector is done via provision of corresponding detector name as string parameter `detectorType` of `processImages()` function in */src/MidTermProject_Camera_Student.cpp*

MP.3 Keypoint Removal

Additionally implemented (using the function `bool cv::Point2f::inside(cv::Rect)` from OPenCV library) the ability to remove the detected keypoints which are located outside of a pre-defined rectangle so that only the keypoints detected within the rectangle can be used for further processing.

Descriptors

MP.4 Keypoint Descriptors

Following descriptor extractors were implemented in */src/matching2D_Student.cpp*:

- **BRIEF.** Implemented in `descKeypoints` function using *`cv::Ptr<cv::DescriptorExtractor> cv::xfeatures2d::BriefDescriptorExtractor::create()`* from OpenCV library.
- **ORB.** Implemented in `descKeypoints` function using *`cv::Ptr<cv::DescriptorExtractor> cv::ORB::create()`* from OpenCV library.
- **FREAK.** Implemented in `descKeypoints` function using *`cv::Ptr<cv::DescriptorExtractor> cv::xfeatures2d::FEAK::create()`* from OpenCV library.
- **AKAZE.** Implemented in `descKeypoints` function using *`cv::Ptr<cv::DescriptorExtractor> cv::AKAZE::create()`* from OpenCV library.
- **SIFT.** Implemented in `descKeypoints` function using *`cv::Ptr<cv::DescriptorExtractor> cv::SIFT::create()`* from OpenCV library.

Selection of required descriptor extractor is done via provision of corresponding descriptor name as string parameter *`descriptorName`* of *`processImages()`* function in */src/MidTermProject_Camera_Student.cpp*

MP.5 Descriptor Matching

FLANN keypoint descriptor matcher is implemented using *`cv::Ptr<cv::DescriptorMatcher> cv::DescriptorMatcher::create(v::DescriptorMatcher::FLANNBASED)`* function of OpenCV library additionally to BruteForce descriptor matcher method in *`matchDescriptors`* function and can be selected via configuring *`matcherType`* parameter in the *`main`* function.

Additionally to default Nearest-Neighbor selection of best match for each matched descriptor it was also implemented the option of using K-Nearest Neighbor selection of K best matches during matching of keypoint descriptors using *`cv::DescriptorMatcher::knnMatch()`* function of OpenCV library in *`matchDescriptors`* function. Required best match selection method can be configured using the *`selectorType`* parameter in the *`main`* function.

MP.6 Descriptor Distance Ratio

If K-Nearest Neighbor selection is used for selecting the K best matches for each keypoint descriptor, then further filtering of ambiguous matches is done in *`matchDescriptors`* function using 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.

Performance Evaluation

MP.7 Performance Evaluation 1

Results of detection of keypoints located on the preceding vehicle for all 10 images of input image data set.

	SHITOMASI		HARRIS		FAST		BRISK		ORB		SIFT		AKAZE	
Image	Number of detected keypoints	Detection time, ms	Number of detected keypoints	Detection time, ms	Number of detected keypoints	Detection time, ms	Number of detected keypoints	Detection time, ms	Number of detected keypoints	Detection time, ms	Number of detected keypoints	Detection time, ms	Number of detected keypoints	Detection time, ms
00	125	10.6038	17	14.6013	149	0.75499	254	35.905	91	96.726	137	101.51	162	62.9549
01	118	6.42582	14	8.98093	152	0.69183	274	33.4833	102	5.43751	131	86.4991	157	61.2001
02	123	7.87735	18	9.33966	150	0.66861	276	30.5589	106	5.3451	121	71.6203	159	58.5503
03	120	7.48569	21	8.38192	155	0.79008	275	30.348	113	5.33614	135	70.3586	154	55.5903
04	120	6.44017	26	9.97809	149	0.84614	293	31.0151	109	5.4421	134	66.8553	162	57.9529
05	113	6.29211	43	24.0414	149	0.65411	275	29.479	124	5.70745	139	64.9842	163	64.4061
06	114	6.47628	18	7.7013	156	0.63020	289	31.4179	129	5.77799	136	65.8117	173	62.9977
07	123	8.23577	31	9.9142	150	0.67688	268	32.5219	127	5.30683	147	66.0101	175	58.0535
08	111	7.55766	26	8.24094	138	0.70126	259	31.8349	124	5.61381	156	67.852	175	58.6951
09	112	6.22328	34	12.0252	143	0.75373	250	32.2465	125	5.47615	135	67.4227	175	58.2166

MP.7 Performance Evaluation 2

Results of matching of detected keypoints located on the preceding vehicle for all 10 images of input image data set using various combinations of keypoint detectors (see detection results in above table in MP.7.1) and all possible combinations of descriptor extractors (note, AKAZE descriptor extractor can be used only in combination with AKAZE keypoint detector). In the matching step, the Brute Force approach is used with the descriptor distance ratio set to 0.8.

- Keypoint detector SHI-TOMASI

	BRISK		BRIEF		ORB		FREAK		SIFT	
Image	Descriptor extraction time, ms	Number of matched keypoints	Descriptor extraction time, ms	Number of matched keypoints	Descriptor extraction time, ms	Number of matched keypoints	Descriptor extraction time, ms	Number of matched keypoints	Descriptor extraction time, ms	Number of matched keypoints
00	1.88901		1.12889		1.57663		18.6163		12.1652	
01	1.10793	95	0.384379	115	1.40375	104	18.1524	90	8.41858	112
02	1.1423	88	0.393727	111	1.42096	103	17.7598	88	8.84204	109
03	1.0983	80	0.395782	104	1.3842	100	16.7732	87	8.62904	104
04	1.32013	90	0.388735	101	1.40405	102	15.9603	89	8.28012	103
05	1.09094	82	0.55027	102	1.45037	103	15.7214	83	8.89097	99
06	1.0541	79	0.387903	102	1.39604	98	15.7074	78	7.63165	101
07	1.12071	85	0.407737	100	1.39632	98	16.2009	81	9.01591	96
08	1.03214	86	0.382539	109	1.35085	102	15.8104	86	7.7235	106
09	1.16892	82	0.472927	100	1.36324	97	15.5023	84	9.0636	97

- Keypoint detector HARRIS

	BRISK		BRIEF		ORB		FREAK		SIFT	
Image	Descriptor extraction time, ms	Number of matched keypoints	Descriptor extraction time, ms	Number of matched keypoints	Descriptor extraction time, ms	Number of matched keypoints	Descriptor extraction time, ms	Number of matched keypoints	Descriptor extraction time, ms	Number of matched keypoints
00	1.24338		1.00975		1.69051		19.195		10.0525	
01	0.394891	12	0.255967	14	1.24588	12	16.8218	13	7.77696	14
02	0.414799	10	0.326559	11	1.72057	13	16.0532	13	7.42376	11
03	0.442669	14	0.270475	15	1.28775	16	15.2428	15	8.65899	16
04	0.447628	15	0.306534	20	1.29858	18	17.4447	15	8.00808	19
05	0.612298	16	0.286175	24	1.60501	24	17.4871	17	7.83419	22
06	0.44848	16	0.238932	26	1.28403	18	17.9786	20	8.24585	22
07	0.510159	15	0.250446	16	1.40349	15	17.8109	14	8.87489	13
08	0.603738	23	0.306885	24	1.73812	24	17.7385	21	7.74252	24
09	0.579949	21	0.317963	23	1.30124	20	16.3282	18	11.2634	22

- Keypoint detector FAST

	BRISK		BRIEF		ORB		FREAK		SIFT	
Image	Descriptor extraction time, ms	Number of matched keypoints	Descriptor extraction time, ms	Number of matched keypoints	Descriptor extraction time, ms	Number of matched keypoints	Descriptor extraction time, ms	Number of matched keypoints	Descriptor extraction time, ms	Number of matched keypoints
00	1.77227		1.09444		1.82948		19.0097		13.1915	
01	1.42291	97	0.359285	119	1.61288	122	15.5626	97	8.73089	118
02	1.16535	104	0.311155	130	1.60275	122	16.2729	98	7.85223	123
03	1.43476	101	0.315891	118	1.43812	115	18.8689	94	8.57893	110
04	1.28928	98	0.310026	126	1.2162	129	17.8786	99	7.94925	119
05	1.29987	85	0.31091	108	1.1403	107	18.3171	88	7.07037	114
06	1.35948	107	0.316689	123	1.35879	120	15.5726	99	7.23244	119
07	1.30235	107	0.510377	131	1.14812	126	16.054	104	7.86608	123
08	1.20966	100	0.317599	125	1.15677	122	16.0411	99	7.8892	117
09	1.3035	100	0.321372	119	1.13904	118	15.9295	103	7.04678	103

- Keypoint detector BRISK

	BRISK		BRIEF		ORB		FREAK		SIFT	
Image	Descriptor extraction time, ms	Number of matched keypoints	Descriptor extraction time, ms	Number of matched keypoints	Descriptor extraction time, ms	Number of matched keypoints	Descriptor extraction time, ms	Number of matched keypoints	Descriptor extraction time, ms	Number of matched keypoints
00	2.33822		0.553118		5.41094		20.2747		21.147	
01	1.98688	168	0.475736	174	4.80281	153	18.7183	154	20.4638	177
02	2.76891	169	0.490753	195	4.515	164	21.2833	173	20.2103	187
03	4.25315	157	0.575041	182	4.57509	155	22.9294	154	18.9673	171
04	3.07392	170	0.492599	177	4.55119	165	21.5513	168	21.3687	177
05	2.36921	171	0.506362	182	4.67322	150	20.016	157	20.9819	168
06	2.11749	186	0.497286	193	4.67618	179	18.8446	181	21.382	190
07	1.98758	174	0.473186	208	5.00181	169	20.2796	169	21.4366	193
08	2.33386	167	0.476862	185	4.71898	173	19.8432	175	21.4967	173
09	2.39738	182	0.466033	179	4.59245	171	21.1851	165	19.5269	181

- Keypoint detector ORB

	BRISK		BRIEF		ORB		FREAK		SIFT	
Image	Descriptor extraction time, ms	Number of matched keypoints	Descriptor extraction time, ms	Number of matched keypoints	Descriptor extraction time, ms	Number of matched keypoints	Descriptor extraction time, ms	Number of matched keypoints	Descriptor extraction time, ms	Number of matched keypoints
00	1.04763		0.439165		5.67964		20.2544		21.7818	
01	0.994693	73	0.336918	49	5.40937	65	16.2513	41	16.1726	66
02	1.11893	74	0.346797	43	5.55475	69	15.214	36	17.7809	79
03	1.13163	79	0.401929	45	5.42852	71	17.7788	45	15.8266	78
04	1.0901	85	0.302278	59	5.03423	85	16.3957	47	16.7771	79
05	1.56375	79	0.331552	53	4.58909	91	15.7791	44	15.8842	82
06	1.55261	90	0.318833	76	4.79681	99	16.0909	51	19.6454	93
07	1.198	88	0.363979	67	4.66324	94	17.4113	52	27.389	94
08	1.16558	86	0.450668	83	4.79392	91	16.0928	48	22.8127	93
09	1.43505	90	0.322087	65	4.96923	89	17.4691	53	21.4834	92

- Keypoint detector SIFT

	BRISK		BRIEF		FREAK		SIFT	
Image	Descriptor extraction time, ms	Number of matched keypoints	Descriptor extraction time, ms	Number of matched keypoints	Descriptor extraction time, ms	Number of matched keypoints	Descriptor extraction time, ms	Number of matched keypoints
00	2.17547		1.2301		22.5614		47.0613	
01	1.25369	63	1.34528	86	22.4598	63	46.926	81
02	1.18773	64	1.21227	76	22.8196	70	45.2265	79
03	1.19356	60	1.24939	72	22.1825	63	45.9821	83
04	1.25328	65	1.20825	83	22.0832	65	45.7732	92
05	1.28919	59	1.25965	69	22.4009	63	46.8624	90
06	1.1962	65	1.23064	75	22.057	59	49.6454	82
07	1.27038	64	1.24814	76	22.0983	64	46.9064	82
08	1.35571	67	1.28214	69	22.2571	65	45.6899	100
09	1.19334	79	1.25797	87	22.5165	79	49.6797	101

- Keypoint detector AKAZE

	BRISK		BRIEF		ORB		FREAK		SIFT		AKAZE	
Image	Descriptor extraction time, ms	Number of matched keypoints	Descriptor extraction time, ms	Number of matched keypoints	Descriptor extraction time, ms	Number of matched keypoints	Descriptor extraction time, ms	Number of matched keypoints	Descriptor extraction time, ms	Number of matched keypoints	Descriptor extraction time, ms	Number of matched keypoints
00	2.56851		1.2924		3.82641		21.868		14.5507		41.8819	
01	1.45058	134	1.15455	137	4.08099	127	20.505	123	13.7646	132	41.4513	135
02	1.41098	124	1.18829	133	3.75656	128	20.4998	127	13.0959	134	38.1852	138
03	1.41177	129	1.13525	130	4.26767	127	20.3635	129	13.3566	129	39.7468	132
04	1.53065	128	1.19032	130	3.92225	114	20.4162	121	13.3588	136	40.78	126
05	1.53454	130	1.12157	134	3.95613	131	20.3713	123	13.851	136	38.7469	128
06	1.49217	132	1.22531	146	4.09889	132	20.3135	132	14.2538	147	40.1297	146
07	1.54077	142	1.24603	150	3.94778	136	20.1478	145	13.4941	147	41.1496	147
08	1.60972	144	1.23827	147	3.73787	138	20.3396	146	13.0103	153	41.392	149
09	1.56344	141	1.21989	150	3.78631	144	20.2973	135	13.1406	149	47.8152	148

MP.7 Performance Evaluation 3

Considering results of testing MP.7 Performance Evaluation 1 part, the **FAST** keypoint detector produces the biggest amount of keypoints **(mean=149 keypoints per image**) for each image within shortest execution time **mean=0.717ms,** acceptable for usage e.g in real time systems. In this context top 3 descriptor extractor options in combination with FAST keypoint detector are BRIEF (mean=122 matches per image), ORB (mean=120 matches per image) and BRISK (mean=100 matches per image), which provide the highest number of keypoint matches and have shortest execution times: **mean=0.417ms**, **mean=1.364ms** and **mean=1.356ms** accordingly.