Bundle Adjustment in the Large

Supplementary Material

In this document we report details of the various problems used in our experiments and detailed information about the performance of each algorithm. For each of the five data sets, we present three things.

1. A table describing the problem size, complexity and the memory usage of each of the bundle adjustment algorithms. To characterize the problem complexity, we measure the sparsity of the reduced camera matrix S and the sparsity of the sparse Cholesky factorization $S = LL^\top$ of the reduced camera matrix/Schur complement as returned by CHOLMOD after applying a fill reducing permutation. $\mathrm{sparsity}(S)$ and $\mathrm{sparsity}(L)$ together are a measure of the input and output complexity of inverting the Newton system using the Schur complement trick exactly. For each algorithm we report the peak memory usage in bytes for that algorithm. NA is used to indicate problems for which the Schur complement S was too large to be calculated and stored in memory (more than 48GB).

2. A table describing for each problem, the initial error, best error achieved by an algorithm, and the time required to achieve it. NA is used where the problem was unable to run because it ran out of memory.

3. A representative selection of plots describing the performance of the algorithms. We plot the RMS error both as a function of wall clock time. The three black dashed horizontal lines in each plot correspond to the three tolerances, i.e., $r^* + 0.01, r^* + 0.001$ and $r^* + 0.001$, where r^* was the best RMS error achieved across all solvers in 50 iterations.

We also reproduce the two main figures of the paper at the end at a larger scale for reasons of clarity. Finally we present an elementary proof of the fact that Schur complements are better conditioned than the full linear system.

1 Dubrovnik

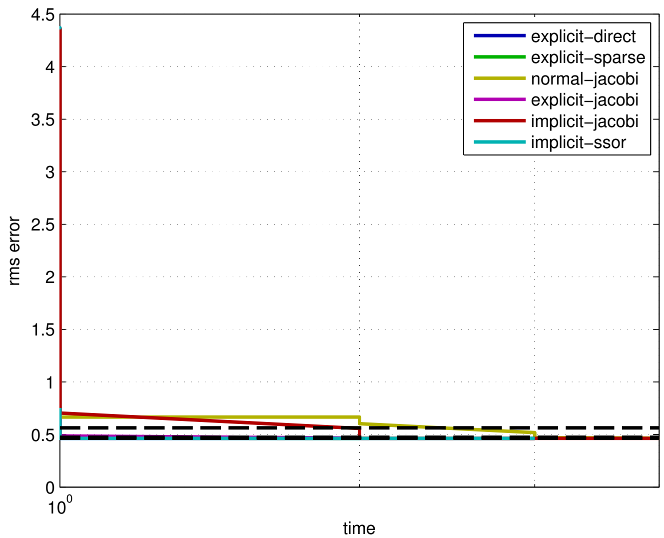
This section presents the results for bundle adjustment on problems generated during the incremental reconstruction of the skeletal set for the Dubrovnik dataset.

Table 1: Problem statistics and memory usages for the incremental reconstruction of the Dubrovnik Skeletal Sets

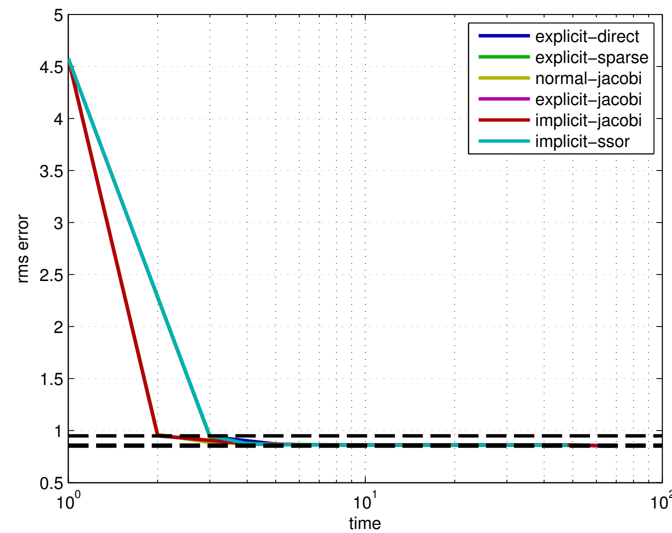
					Memory Usage (Bytes)					
images	points	obs	sparsity(S)	sparsity(L)	explicit-direct	explicit-sparse	explicit-jacobi	implicit-jacobi	implicit-ssor	normal-jacobi
16	22106	83718	0.00	0.00	5.1e+07	5.1e+07	5.1e+07	3.6e+07	3.8e+07	3.8e+07
88	64298	383937	0.13	0.00	2.3e+08	2.4e+08	2.3e+08	1.5e+08	1.5e+08	1.5e+08
135	90642	553336	0.21	0.08	3.4e+08	3.5e+08	3.3e+08	2.1e+08	2.2e+08	2.2e+08
142	93602	565608	0.24	0.06	3.5e+08	3.6e+08	3.4e+08	2.2e+08	2.2e+08	2.3e+08
150	95821	568119	0.29	0.03	3.5e+08	3.7e+08	3.4e+08	2.2e+08	2.3e+08	2.3e+08
161	103832	592019	0.35	0.09	3.7e+08	3.8e+08	3.5e+08	2.3e+08	2.4e+08	2.4e+08
173	111908	634570	0.36	0.10	4.0e+08	4.1e+08	3.8e+08	2.5e+08	2.6e+08	2.6e+08
182	116770	668705	0.36	0.08	4.2e+08	4.4e+08	4.0e+08	2.6e+08	2.7e+08	2.7e+08
202	132796	751652	0.36	0.12	4.8e+08	5.0e+08	4.5e+08	2.9e+08	3.0e+08	3.1e+08
237	154414	858331	0.35	0.11	5.5e+08	5.9e+08	5.2e+08	3.3e+08	3.5e+08	3.5e+08
253	163691	899155	0.37	0.12	5.9e+08	6.2e+08	5.5e+08	3.5e+08	3.7e+08	3.7e+08
262	169354	919688	0.38	0.13	6.0e+08	6.4e+08	5.7e+08	3.6e+08	3.8e+08	3.8e+08
273	176305	942970	0.38	0.13	6.2e+08	6.6e+08	5.8e+08	3.7e+08	3.9e+08	3.9e+08
287	182023	971292	0.39	0.13	6.5e+08	6.9e+08	6.0e+08	3.8e+08	4.0e+08	4.0e+08
308	195089	1045197	0.41	0.13	7.0e+08	7.5e+08	6.5e+08	4.1e+08	4.3e+08	4.3e+08
356	226730	1255268	0.46	0.09	8.5e+08	9.1e+08	7.8e+08	4.9e+08	5.1e+08	5.2e+08

Table 2: Perfomance statistics for the incremental reconstruction of the Dubrovnik Skeletal Sets

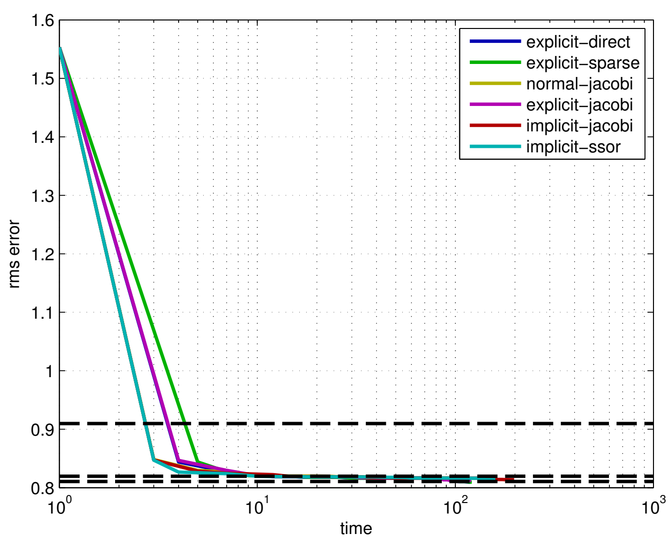
				explicit-direct		explicit-sparse		explicit-jacobi		implicit-jacobi		implicit-ssor		normal-jacobi	
images	points	obs	initial	time	error	time	error	time	error	time	error	time	error	time	error
16	22106	83718	4.369	1	0.464	2	0.464	2	0.464	4	0.464	2	0.464	1	0.464
88	64298	383937	4.577	69	0.851	69	0.851	71	0.850	55	0.851	65	0.851	51	0.851
135	90642	553336	1.553	116	0.810	118	0.810	99	0.813	94	0.816	187	0.814	159	0.813
142	93602	565608	1.037	114	0.691	117	0.691	97	0.691	114	0.691	99	0.691	56	0.691
150	95821	568119	0.776	119	0.653	122	0.653	97	0.653	121	0.653	113	0.653	87	0.653
161	103832	592019	0.817	43	0.636	44	0.636	38	0.636	148	0.636	102	0.636	63	0.636
173	111908	634570	0.836	134	0.636	138	0.636	120	0.636	171	0.636	210	0.636	101	0.636
182	116770	668705	0.971	49	0.667	52	0.667	38	0.667	166	0.667	139	0.667	103	0.667
202	132796	751652	1.321	30	0.656	31	0.656	50	0.656	145	0.656	198	0.656	183	0.656
237	154414	858331	1.375	69	0.642	72	0.642	86	0.642	222	0.642	459	0.642	478	0.642
253	163691	899155	0.855	83	0.617	88	0.617	123	0.617	567	0.617	836	0.617	557	0.617
262	169354	919688	0.784	87	0.608	95	0.608	111	0.608	1315	0.608	1076	0.608	1005	0.608
273	176305	942970	0.717	79	0.601	81	0.601	115	0.601	402	0.601	1297	0.601	748	0.601
287	182023	971292	0.913	69	0.601	71	0.601	102	0.601	784	0.601	1167	0.601	871	0.601
308	195089	1045197	1.461	69	0.609	73	0.609	98	0.609	1015	0.609	517	0.609	275	0.609
356	226730	1255268	1.858	76	0.625	78	0.625	142	0.625	572	0.625	544	0.625	569	0.625



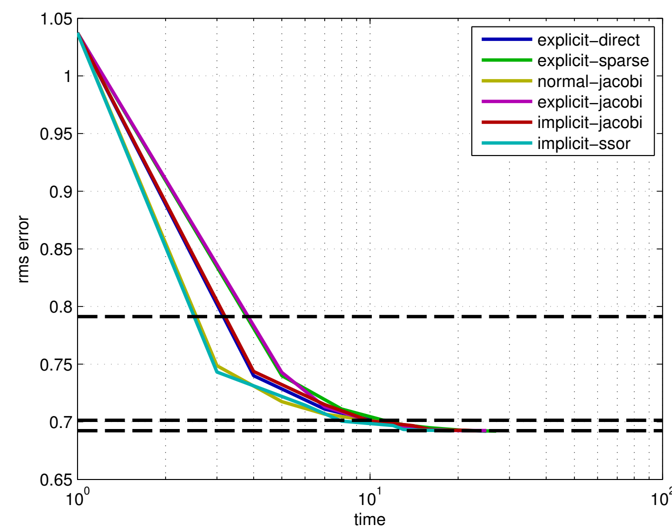
images = 16, points = 22106, observations= 83718



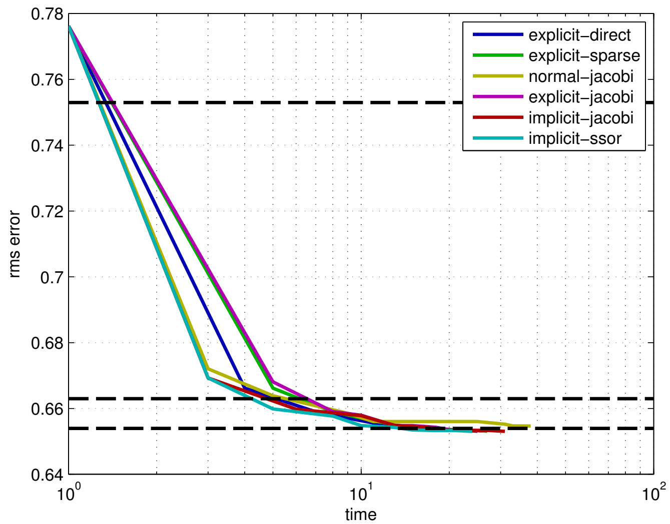
images = 88, points = 64298, observations= 383937



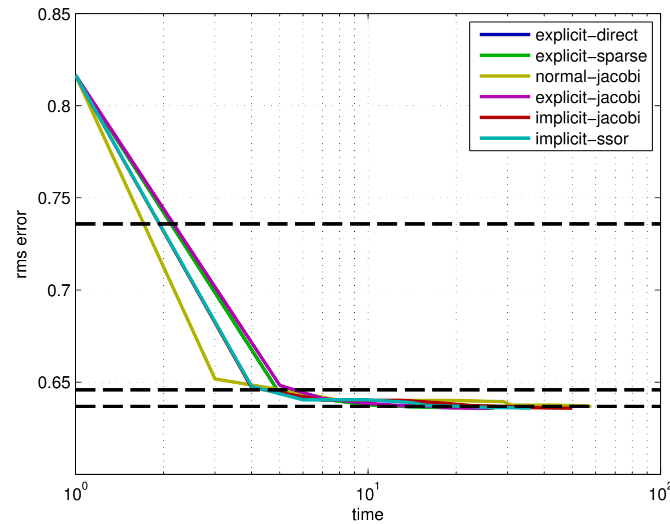
images = 135, points = 90642, observations= 553336



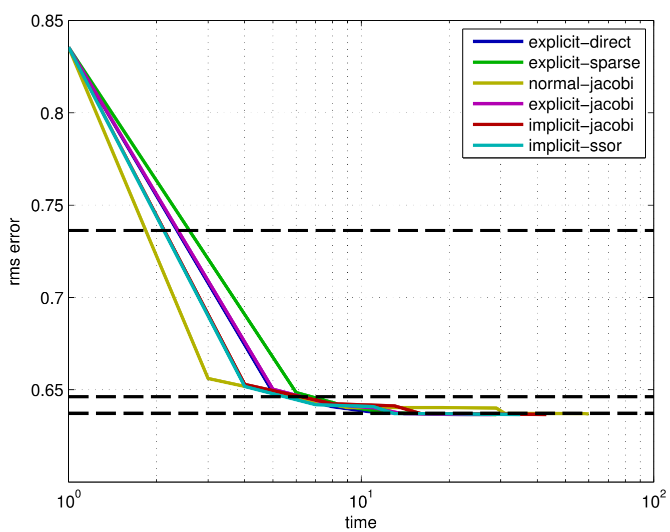
images = 142, points = 93602, observations= 565608



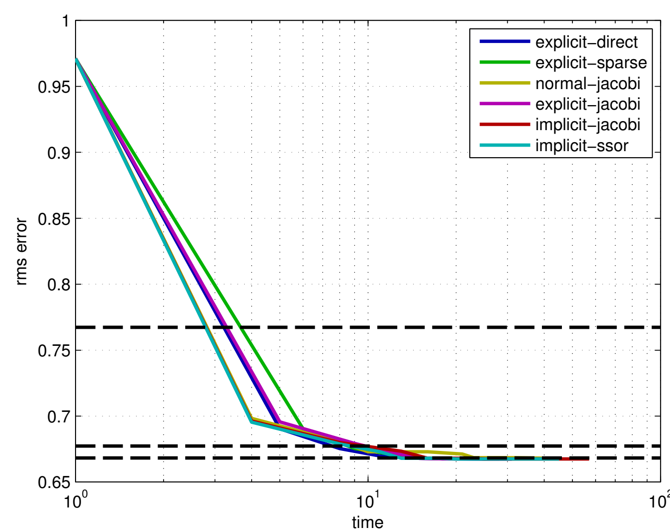
images = 150, points = 95821, observations= 568119



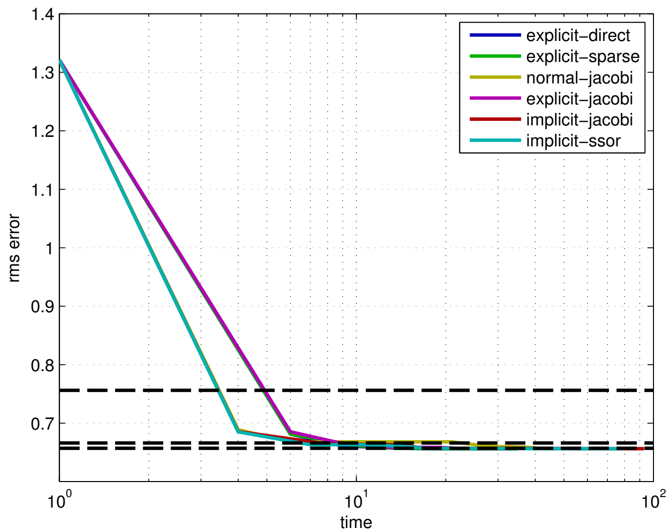
images = 161, points = 103832, observations= 592019



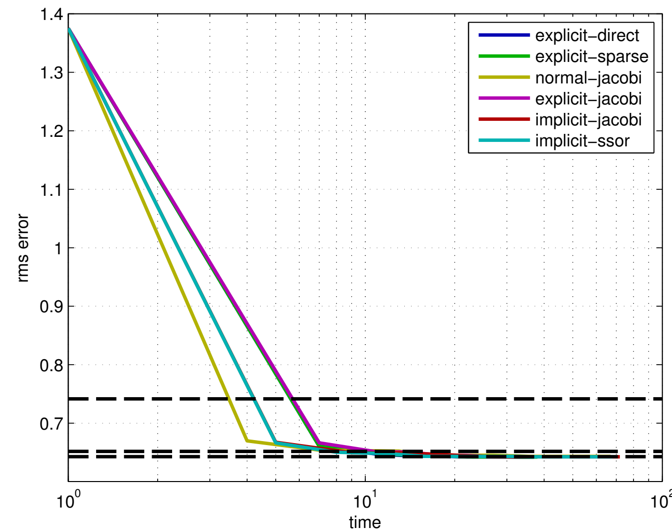
images = 173, points = 111908, observations= 634570



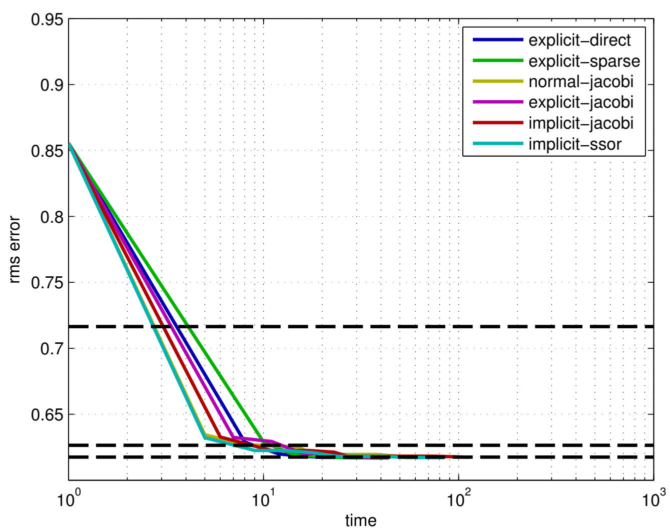
images = 182, points = 116770, observations= 668705



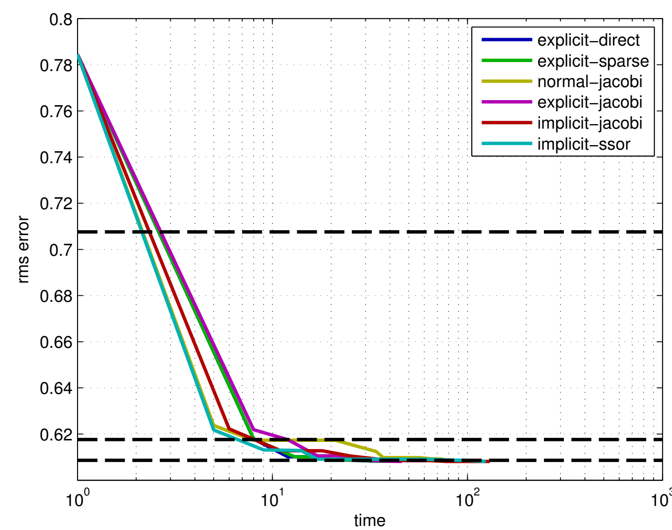
images = 202, points = 132796, observations= 751652



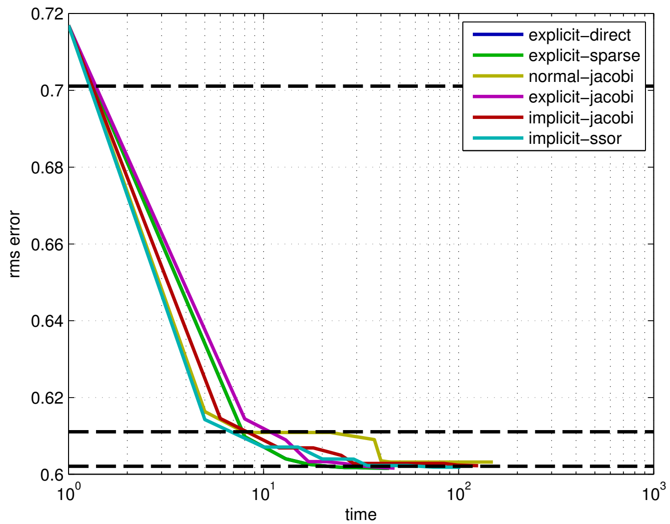
images = 237, points = 154414, observations= 858331



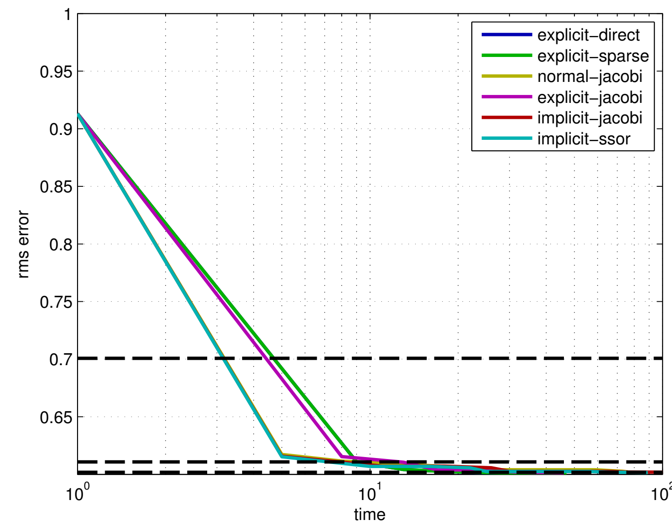
images = 253, points = 163691, observations= 899155



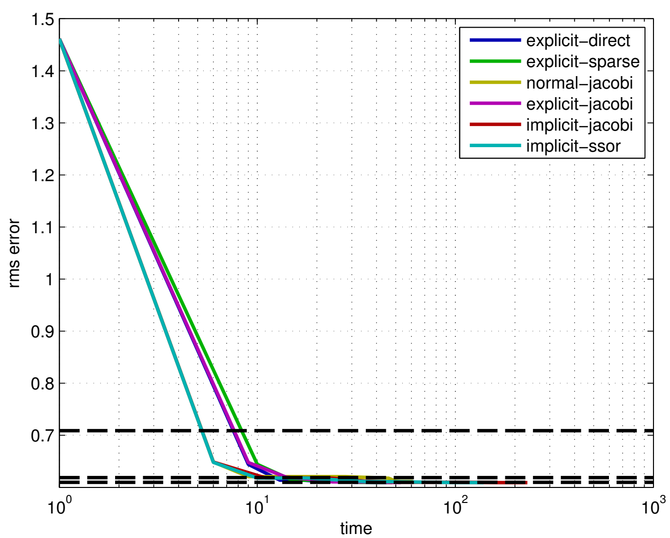
images = 262, points = 169354, observations= 919688



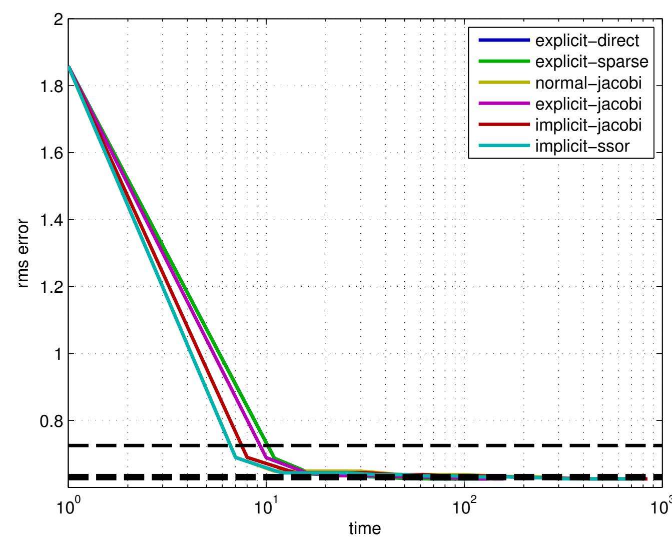
images = 273, points = 176305, observations= 942970



images = 287, points = 182023, observations= 971292



images = 308, points = 195089, observations= 1045197



images = 356, points = 226730, observations= 1255268

2 Trafalgar Square

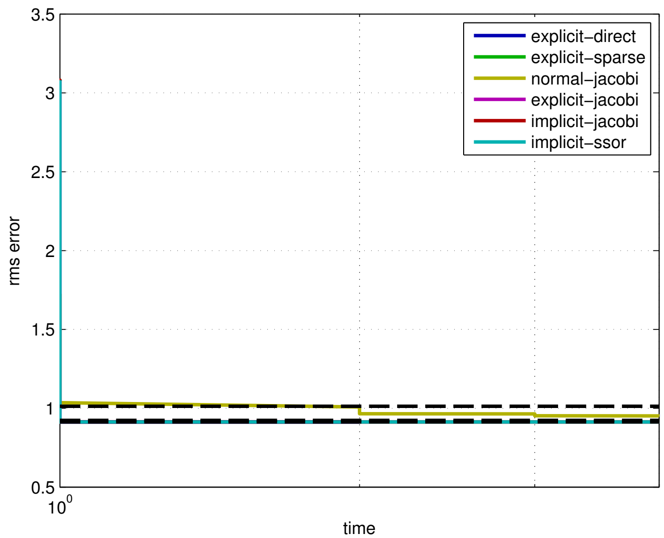
This section presents the results for bundle adjustment on problems generated during the incremental reconstruction of the skeletal set for the Trafalgar square dataset.

Table 3: Problem statistics and memory usages for the incremental reconstruction of the Trafalgar square Skeletal Sets

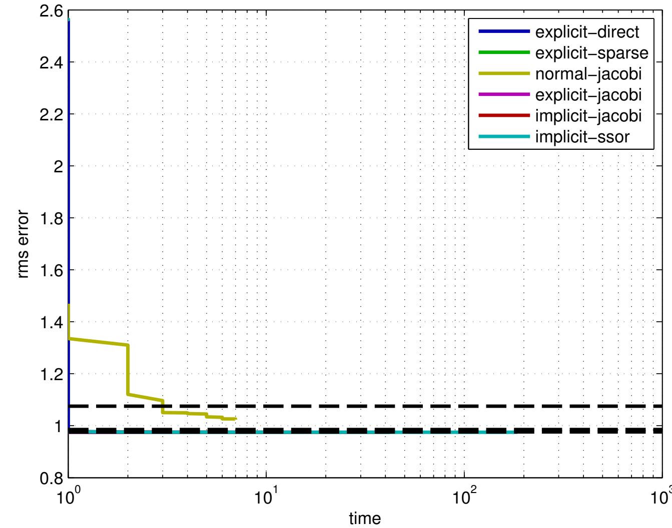
					Memory Usage (Bytes)					
images	points	obs	sparsity(S)	sparsity(L)	explicit-direct	explicit-sparse	explicit-jacobi	implicit-jacobi	implicit-ssor	normal-jacobi
21	11315	36455	0.10	0.08	2.3e+07	2.4e+07	2.3e+07	1.7e+07	1.8e+07	1.8e+07
39	18060	63551	0.15	0.08	4.1e+07	4.2e+07	4.1e+07	2.8e+07	3.0e+07	3.0e+07
50	20431	73967	0.24	0.11	4.8e+07	5.0e+07	4.7e+07	3.3e+07	3.5e+07	3.5e+07
126	40037	148328	0.62	0.37	1.0e+08	1.1e+08	9.7e+07	6.5e+07	6.9e+07	6.9e+07
138	44033	165899	0.64	0.31	1.2e+08	1.2e+08	1.1e+08	7.2e+07	7.8e+07	7.8e+07
161	48126	182072	0.66	0.30	1.3e+08	1.4e+08	1.2e+08	7.9e+07	8.6e+07	8.6e+07
170	49267	185815	0.67	0.37	1.4e+08	1.4e+08	1.2e+08	8.1e+07	8.9e+07	8.7e+07
174	50489	188809	0.67	0.39	1.4e+08	1.4e+08	1.3e+08	8.3e+07	8.9e+07	8.9e+07
193	53101	196526	0.69	0.22	1.5e+08	1.7e+08	1.3e+08	8.6e+07	9.4e+07	9.4e+07
201	54427	199938	0.71	0.26	1.6e+08	1.7e+08	1.3e+08	8.8e+07	9.6e+07	9.5e+07
206	54562	200715	0.71	0.24	1.6e+08	1.7e+08	1.4e+08	8.9e+07	9.5e+07	9.8e+07
215	55910	204202	0.72	0.26	1.6e+08	1.7e+08	1.4e+08	9.0e+07	1.0e+08	1.0e+08
225	57665	208622	0.73	0.28	1.7e+08	1.8e+08	1.4e+08	9.3e+07	1.0e+08	1.0e+08
257	65132	225911	0.76	0.33	1.9e+08	2.0e+08	1.6e+08	1.0e+08	1.1e+08	1.1e+08

Table 4: Perfomance statistics for the incremental reconstruction of the Trafalgar square Skeletal Sets

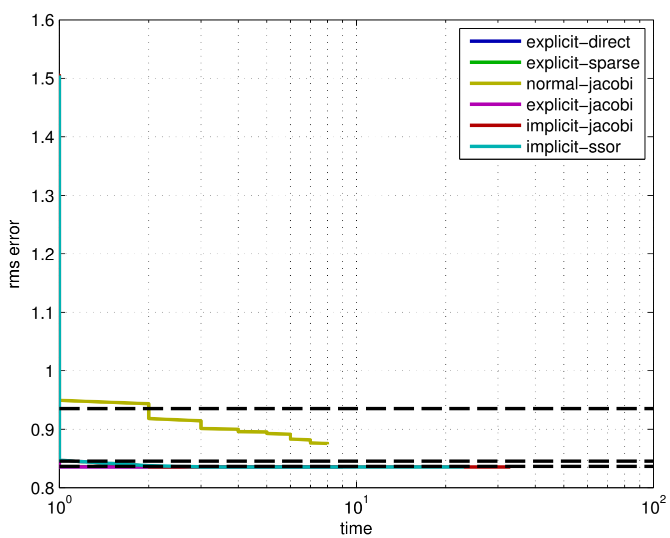
				explicit-direct		explicit-sparse		explicit-jacobi		implicit-jacobi		implicit-ssor		normal-jacobi	
images	points	obs	initial	time	error	time	error	time	error	time	error	time	error	time	error
21	11315	36455	3.081	1	0.913	0	0.913	1	0.913	4	0.946	1	0.913	2	0.913
39	18060	63551	2.563	1	0.975	1	0.975	1	0.975	7	1.023	2	0.975	2	0.975
50	20431	73967	1.505	1	0.835	2	0.835	2	0.835	8	0.874	3	0.835	3	0.835
126	40037	148328	2.010	7	0.787	7	0.787	14	0.787	18	0.821	160	0.787	592	0.786
138	44033	165899	1.343	10	0.724	10	0.724	23	0.724	19	0.751	269	0.724	259	0.724
161	48126	182072	0.834	14	0.680	14	0.680	34	0.680	21	0.707	323	0.680	315	0.680
170	49267	185815	0.841	16	0.680	16	0.680	33	0.680	22	0.709	338	0.680	346	0.680
174	50489	188809	0.810	17	0.674	17	0.674	38	0.674	22	0.701	384	0.674	781	0.673
193	53101	196526	0.857	16	0.674	16	0.674	25	0.673	23	0.701	367	0.674	264	0.674
201	54427	199938	0.814	37	0.677	38	0.677	44	0.671	23	0.711	321	0.671	277	0.671
206	54562	200715	0.782	16	0.670	16	0.670	57	0.670	24	0.696	455	0.670	305	0.670
215	55910	204202	0.832	16	0.669	16	0.669	22	0.669	24	0.696	408	0.668	347	0.668
225	57665	208622	0.765	16	0.662	17	0.662	43	0.662	25	0.688	401	0.662	341	0.662
257	65132	225911	0.734	37	0.654	37	0.654	48	0.654	28	0.679	472	0.654	361	0.654



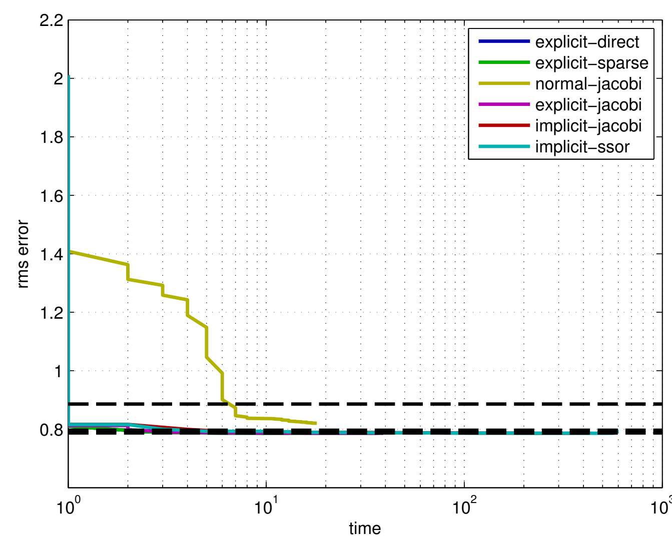
images = 21, points = 11315, observations= 36455



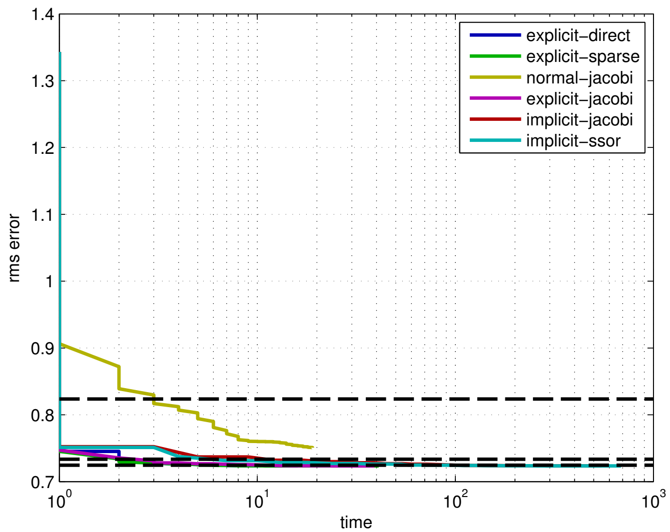
images = 39, points = 18060, observations= 63551



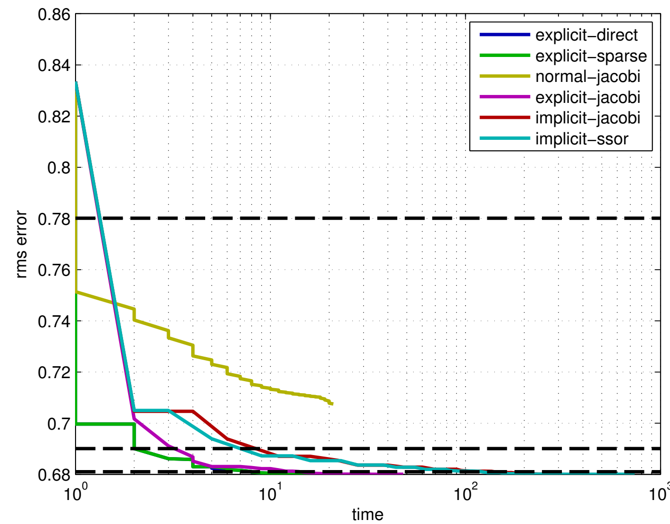
images = 50, points = 20431, observations= 73967



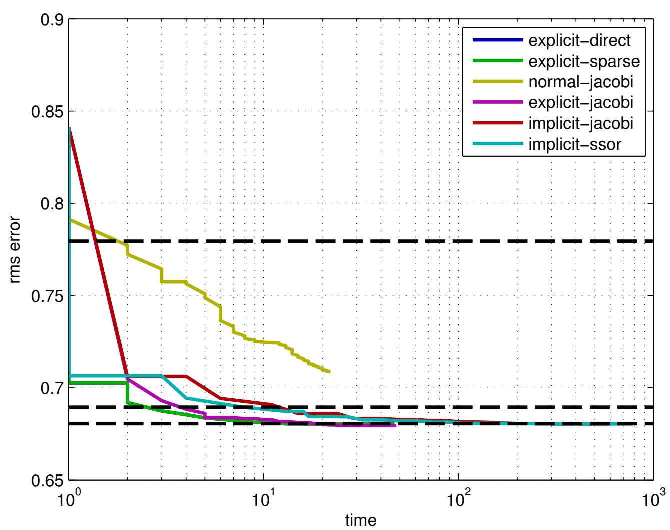
images = 126, points = 40037, observations= 148328



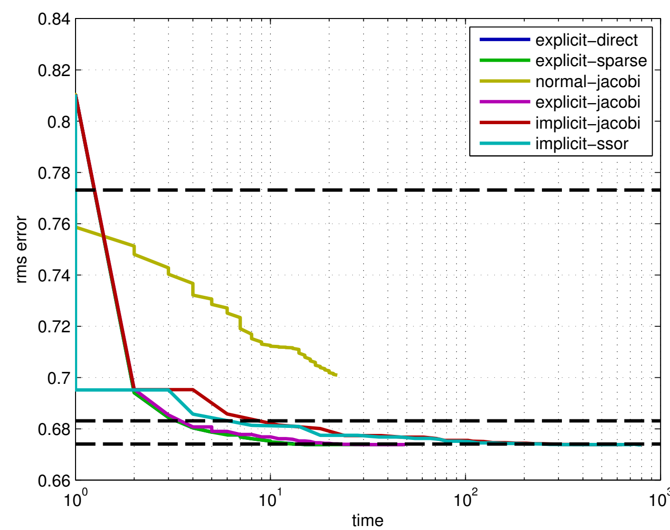
images = 138, points = 44033, observations= 165899



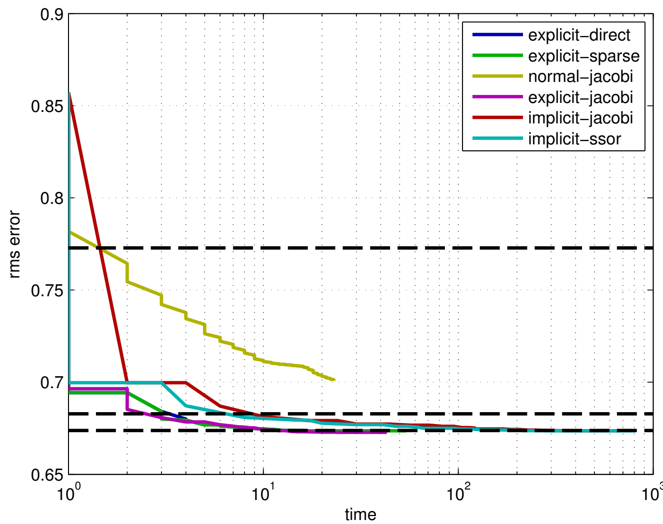
images = 161, points = 48126, observations= 182072



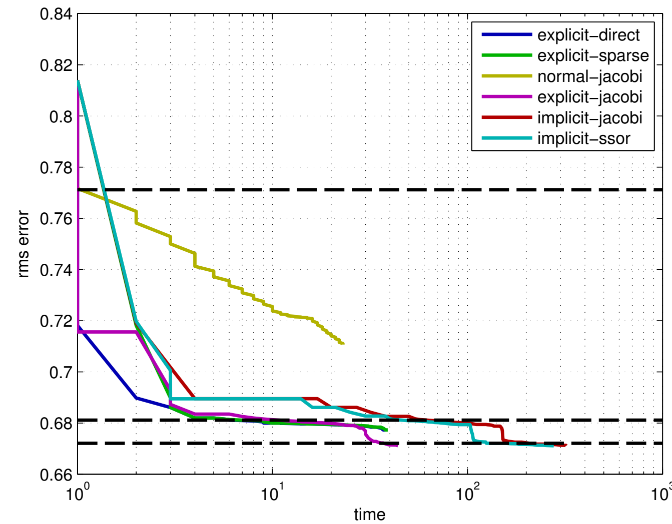
images = 170, points = 49267, observations= 185815



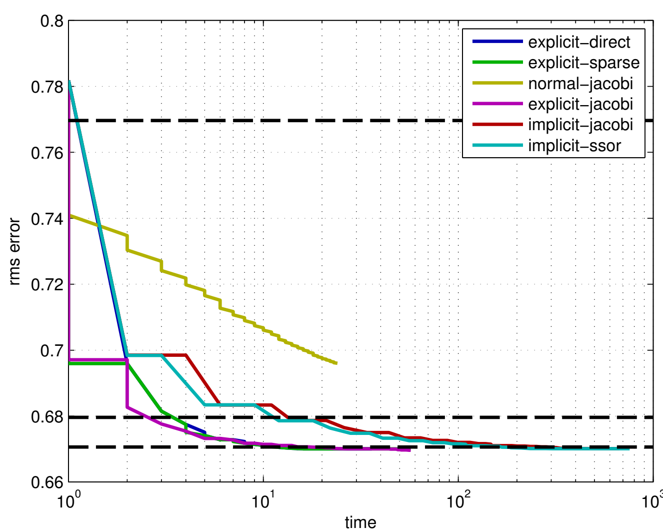
images = 174, points = 50489, observations= 188809



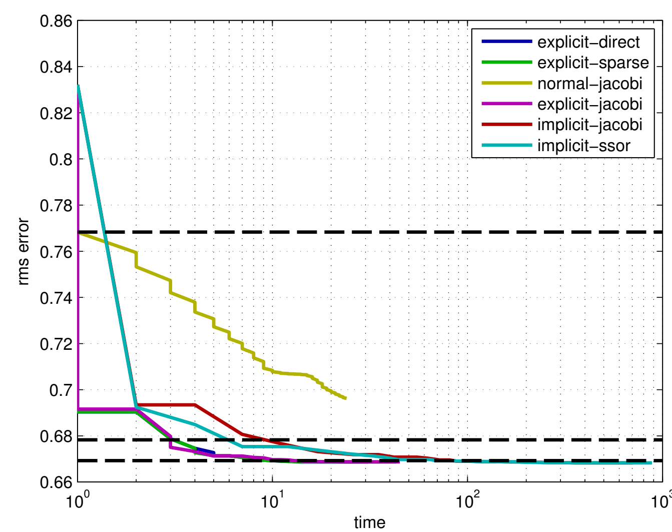
images = 193, points = 53101, observations= 196526



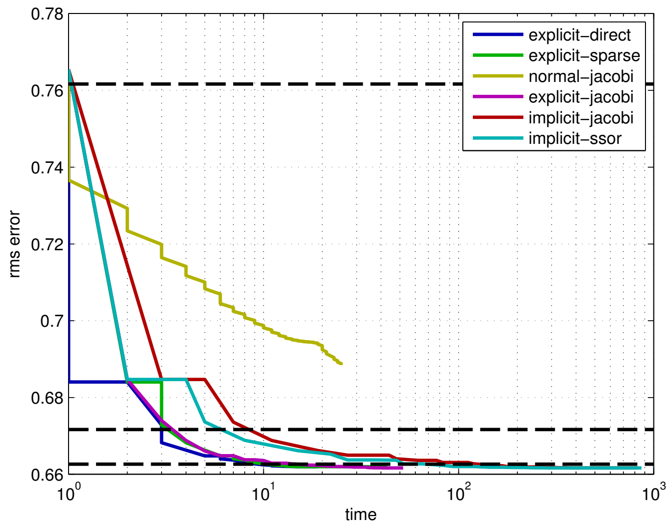
images = 201, points = 54427, observations= 199938



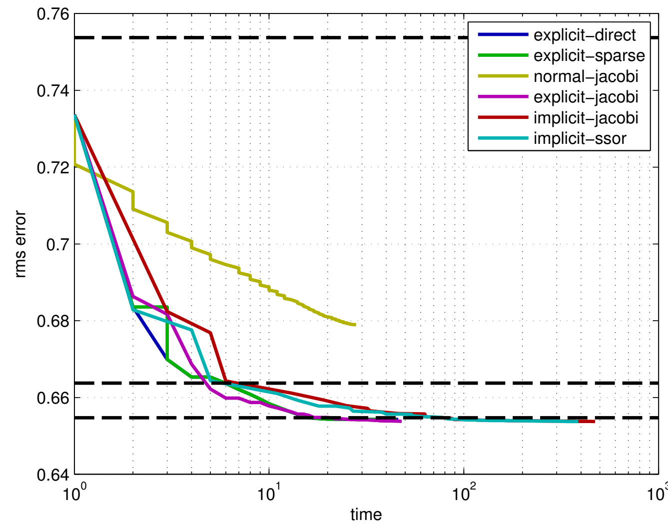
images = 206, points = 54562, observations= 200715



images = 215, points = 55910, observations= 204202



images = 225, points = 57665, observations= 208622



images = 257, points = 65132, observations= 225911

3 Venice

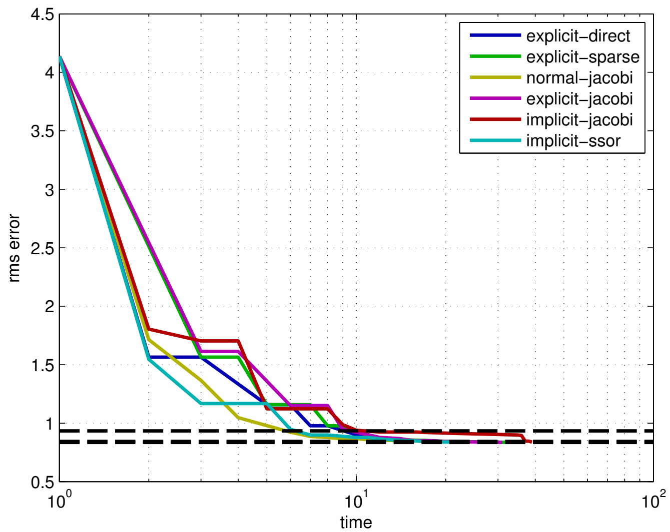
This section presents the results for bundle adjustment on problems generated during the incremental reconstruction of the skeletal set for the Venice dataset.

Table 5: Problem statistics and memory usages for the incremental reconstruction of the Venice Skeletal Sets

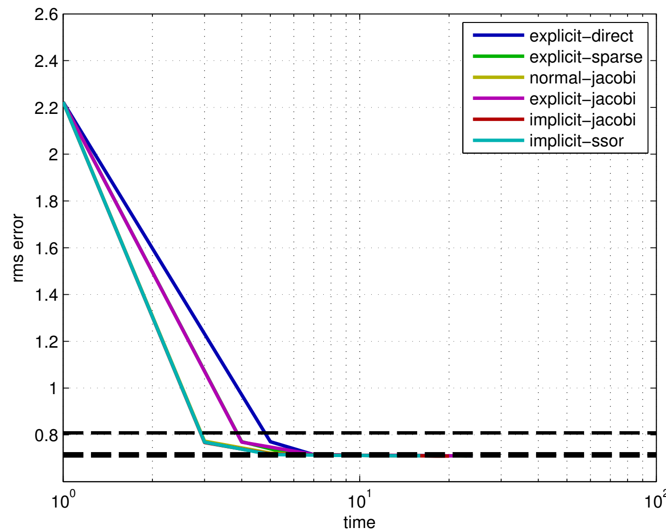
					Memory Usage (Bytes)					
images	points	obs	sparsity(S)	sparsity(L)	explicit-direct	explicit-sparse	explicit-jacobi	implicit-jacobi	implicit-ssor	normal-jacobi
52	64053	347173	0.02	0.00	2.0e+08	2.1e+08	2.0e+08	1.4e+08	1.4e+08	1.4e+08
89	110973	562976	0.07	0.00	3.4e+08	3.4e+08	3.3e+08	2.2e+08	2.3e+08	2.3e+08
245	198739	1091386	0.35	0.00	6.9e+08	7.3e+08	6.6e+08	4.3e+08	4.5e+08	4.5e+08
427	310384	1699145	0.46	0.01	1.2e+09	1.3e+09	1.1e+09	6.6e+08	7.0e+08	7.0e+08
744	543562	3058863	0.58	0.06	2.3e+09	2.6e+09	1.9e+09	1.2e+09	1.2e+09	1.2e+09
951	708276	3748892	0.66	0.11	3.0e+09	3.3e+09	2.4e+09	1.5e+09	1.6e+09	1.6e+09
1102	780462	4052340	0.70	0.10	3.4e+09	3.8e+09	2.6e+09	1.6e+09	1.7e+09	1.7e+09
1158	802917	4130503	0.71	0.11	3.5e+09	3.9e+09	2.7e+09	1.6e+09	1.7e+09	1.7e+09
1184	816583	4179047	0.72	0.14	3.6e+09	4.0e+09	2.7e+09	1.7e+09	1.7e+09	1.7e+09
1238	843534	4290503	0.74	0.17	3.8e+09	4.1e+09	2.8e+09	1.7e+09	1.8e+09	1.8e+09
1288	866452	4383006	0.75	0.19	3.9e+09	4.2e+09	2.9e+09	1.7e+09	1.8e+09	1.8e+09
1350	894716	4517126	0.77	0.20	4.1e+09	4.4e+09	3.0e+09	1.8e+09	1.9e+09	1.9e+09
1408	912229	4634530	0.78	0.19	4.3e+09	4.6e+09	3.0e+09	1.8e+09	1.9e+09	1.9e+09
1425	916895	4657314	0.78	0.20	4.3e+09	4.6e+09	3.0e+09	1.9e+09	2.0e+09	2.0e+09
1473	930345	4705873	0.79	0.21	4.5e+09	4.7e+09	3.1e+09	1.9e+09	2.0e+09	2.0e+09
1490	935273	4721817	0.79	0.23	4.5e+09	5.0e+09	3.1e+09	1.9e+09	2.0e+09	2.0e+09
1521	939551	4739031	0.80	0.22	4.6e+09	4.8e+09	3.1e+09	1.9e+09	2.0e+09	2.0e+09
1544	942409	4750193	0.80	0.22	4.6e+09	4.9e+09	3.1e+09	1.9e+09	2.0e+09	2.0e+09
1638	976803	4956814	0.82	0.66	5.0e+09	4.3e+09	3.3e+09	2.0e+09	2.1e+09	2.1e+09
1666	983911	4987145	0.82	0.68	5.0e+09	4.2e+09	3.3e+09	2.0e+09	2.1e+09	2.1e+09
1672	986962	5000110	0.82	0.67	5.1e+09	4.3e+09	3.3e+09	2.0e+09	2.1e+09	2.1e+09
1681	983415	4966839	0.82	0.68	5.1e+09	4.3e+09	3.3e+09	2.0e+09	2.1e+09	2.1e+09
1682	983268	4965018	0.82	0.68	5.1e+09	4.2e+09	3.3e+09	2.0e+09	2.1e+09	2.1e+09
1684	983269	4965728	0.82	0.68	5.1e+09	4.3e+09	3.3e+09	2.0e+09	2.1e+09	2.1e+09
1695	984689	4970942	0.82	0.68	5.1e+09	4.2e+09	3.3e+09	2.0e+09	2.1e+09	2.1e+09
1696	984816	4970895	0.82	0.66	5.1e+09	4.4e+09	3.3e+09	2.0e+09	2.1e+09	2.1e+09
1706	985529	4974631	0.82	0.68	5.1e+09	4.2e+09	3.3e+09	2.0e+09	2.1e+09	2.1e+09
1776	993909	5001859	0.84	0.69	5.3e+09	4.4e+09	3.3e+09	2.0e+09	2.1e+09	2.1e+09
1778	993923	5001946	0.84	0.70	5.3e+09	4.4e+09	3.3e+09	2.0e+09	2.1e+09	2.1e+09

Table 6: Perfomance statistics for the incremental reconstruction of the Venice Skeletal Sets

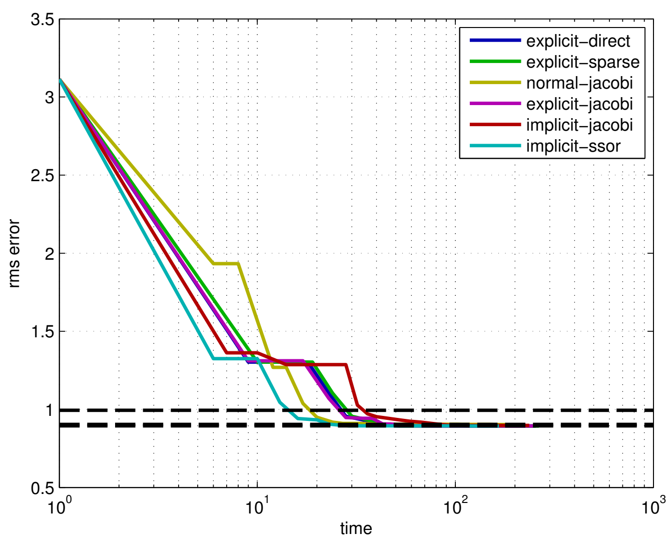
				explicit-direct		explicit-sparse		explicit-jacobi		implicit-jacobi		implicit-ssor		normal-jacobi	
images	points	obs	initial	time	error	time	error	time	error	time	error	time	error	time	error
52	64053	347173	4.137	49	0.835	50	0.835	44	0.834	35	0.837	52	0.834	32	0.834
89	110973	562976	2.223	57	0.709	57	0.709	32	0.709	37	0.709	24	0.709	23	0.709
245	198739	1091386	3.113	247	0.894	256	0.894	77	0.894	157	0.903	135	0.897	121	0.894
427	310384	1699145	2.642	105	0.789	111	0.789	96	0.789	171	0.789	158	0.789	975	0.785
744	543562	3058863	2.381	365	0.707	395	0.707	447	0.707	338	0.707	347	0.707	724	0.707
951	708276	3748892	1.632	659	0.646	698	0.646	1018	0.646	746	0.646	1960	0.647	573	0.647
1102	780462	4052340	1.110	814	0.686	869	0.686	893	0.689	169	0.623	1409	0.616	1263	0.616
1158	802917	4130503	0.833	559	0.589	594	0.589	600	0.589	283	0.591	947	0.589	1058	0.589
1184	816583	4179047	0.767	497	0.580	527	0.580	592	0.581	382	0.591	627	0.582	285	0.588
1238	843534	4290503	0.828	511	0.579	528	0.579	276	0.581	453	0.580	653	0.581	395	0.580
1288	866452	4383006	0.847	468	0.570	472	0.570	522	0.570	209	0.573	880	0.570	516	0.570
1350	894716	4517126	0.880	544	0.571	542	0.571	560	0.571	967	0.571	6129	0.571	1977	0.571
1408	912229	4634530	0.873	542	0.582	541	0.582	428	0.582	388	0.583	256	0.585	3911	0.582
1425	916895	4657314	0.682	478	0.578	464	0.578	848	0.578	2081	0.578	276	0.578	197	0.583
1473	930345	4705873	0.659	498	0.569	482	0.569	330	0.569	4913	0.569	3089	0.569	843	0.569
1490	935273	4721817	0.659	566	0.568	545	0.568	595	0.568	431	0.568	840	0.568	426	0.568
1521	939551	4739031	0.658	611	0.568	588	0.568	447	0.568	940	0.568	1046	0.568	2929	0.568
1544	942409	4750193	0.648	906	0.567	867	0.567	653	0.567	990	0.567	1952	0.567	420	0.568
1638	976803	4956814	1.127	2378	0.705	1450	0.705	1391	0.710	1789	0.719	2933	0.712	3664	0.719
1666	983911	4987145	1.042	683	0.663	413	0.663	1342	0.650	1322	0.650	4724	0.645	4744	0.646
1672	986962	5000110	1.002	1248	0.617	897	0.617	840	0.610	737	0.618	1543	0.606	879	0.607
1681	983415	4966839	0.693	2199	0.576	1310	0.576	1291	0.576	2157	0.578	4786	0.574	1293	0.577
1682	983268	4965018	0.664	2127	0.571	1249	0.571	1502	0.571	2371	0.575	3652	0.571	4031	0.571
1684	983269	4965728	0.667	2626	0.571	1554	0.571	1640	0.571	4084	0.574	3701	0.572	2993	0.572
1695	984689	4970942	0.674	1905	0.574	1107	0.574	1516	0.574	3243	0.577	2388	0.575	2957	0.573
1696	984816	4970895	0.665	1379	0.570	763	0.570	1127	0.571	2477	0.574	4763	0.571	3730	0.571
1706	985529	4974631	0.671	3745	0.575	1082	0.575	1449	0.575	3459	0.578	1839	0.576	1655	0.576
1776	993909	5001859	0.659	2871	0.569	1596	0.569	1475	0.571	8190	0.572	5766	0.571	4419	0.571
1778	993923	5001946	0.658	2875	0.569	1590	0.569	1525	0.571	6859	0.572	5377	0.571	4329	0.571



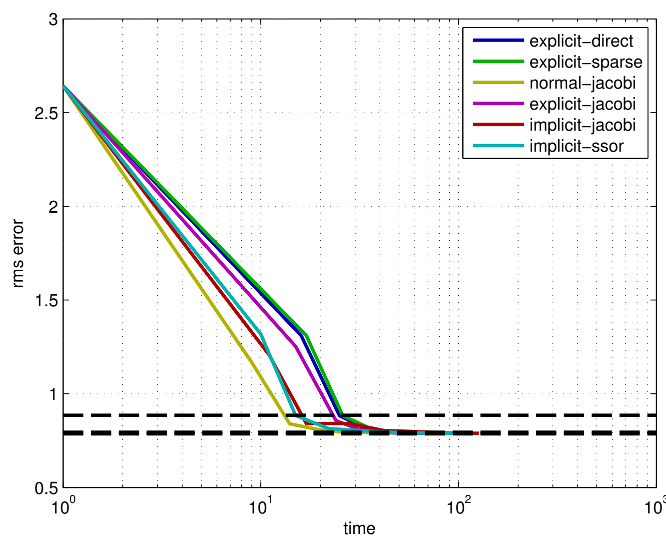
images = 52, points = 64053, observations= 347173



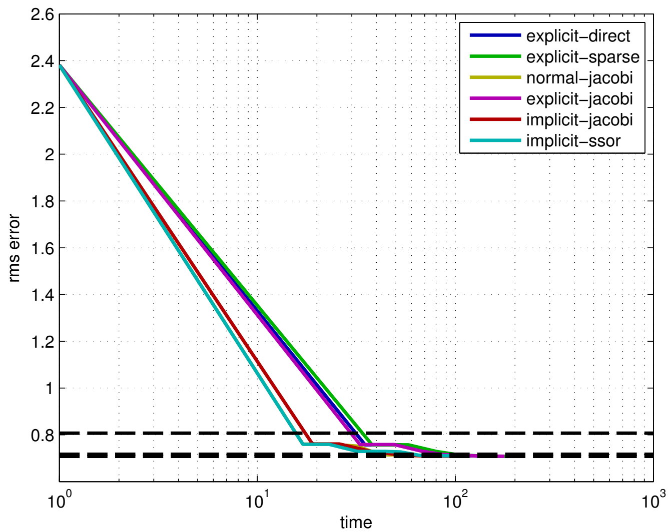
images = 89, points = 110973, observations= 562976



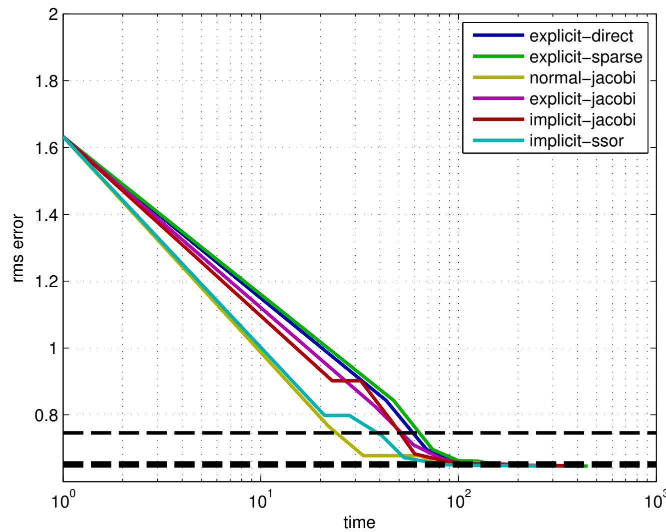
images = 245, points = 198739, observations= 1091386



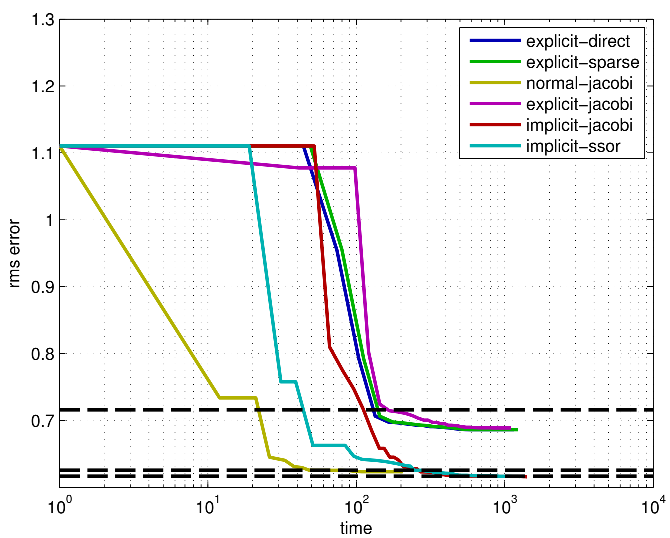
images = 427, points = 310384, observations= 1699145



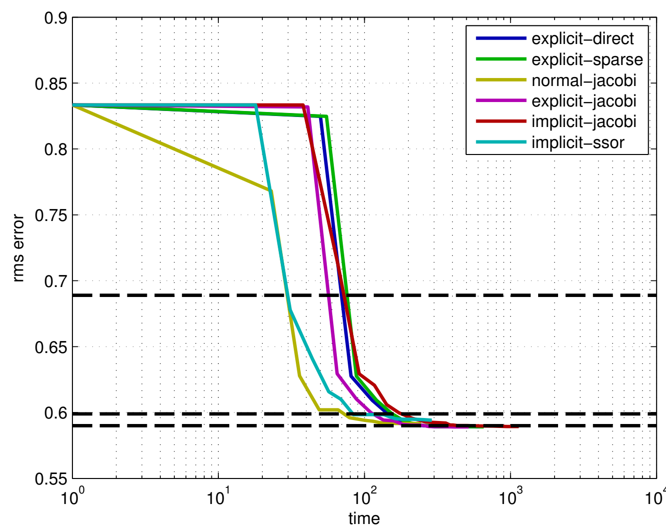
images = 744, points = 543562, observations= 3058863



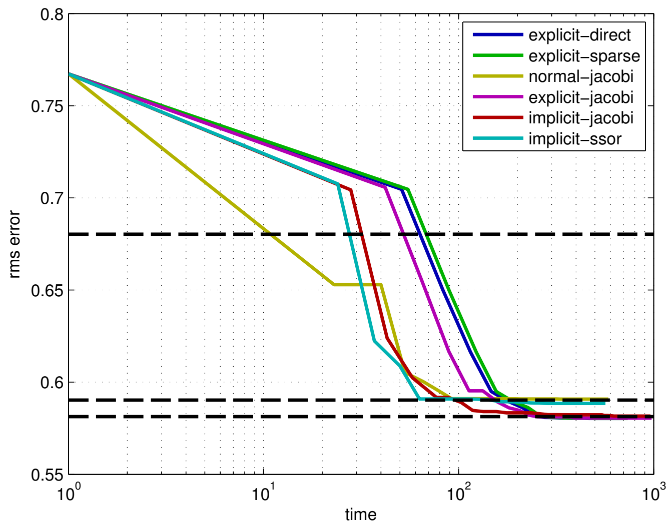
images = 951, points = 708276, observations= 3748892



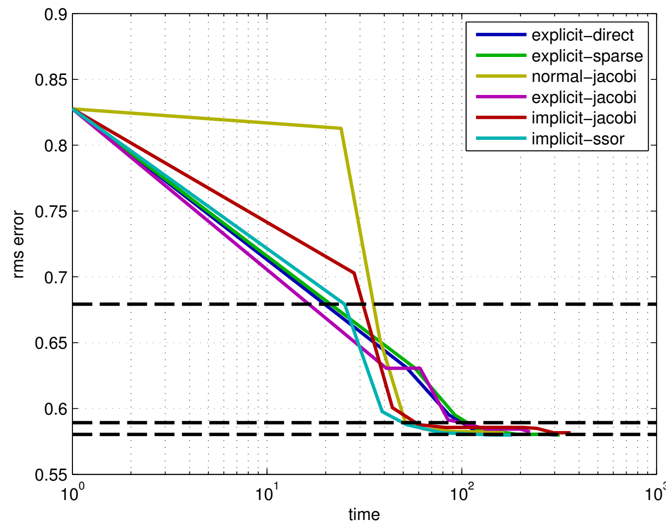
images = 1102, points = 780462, observations= 4052340



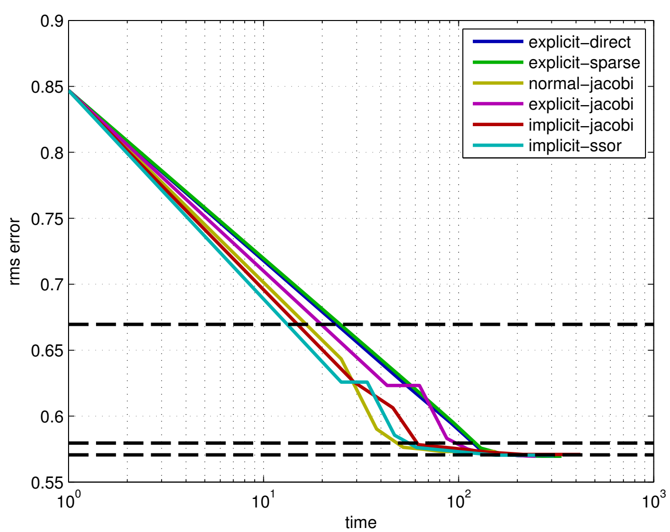
images = 1158, points = 802917, observations= 4130503



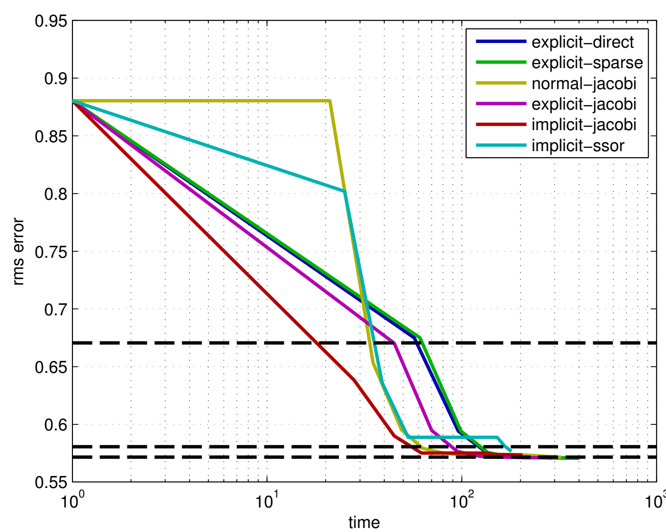
images = 1184, points = 816583, observations= 4179047



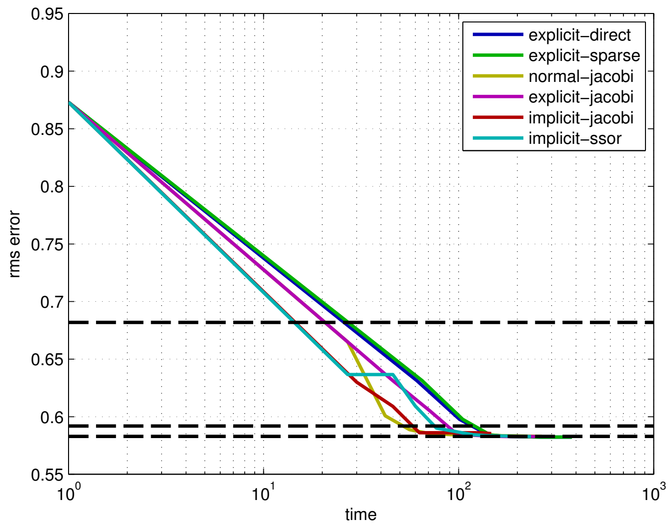
images = 1238, points = 843534, observations= 4290503



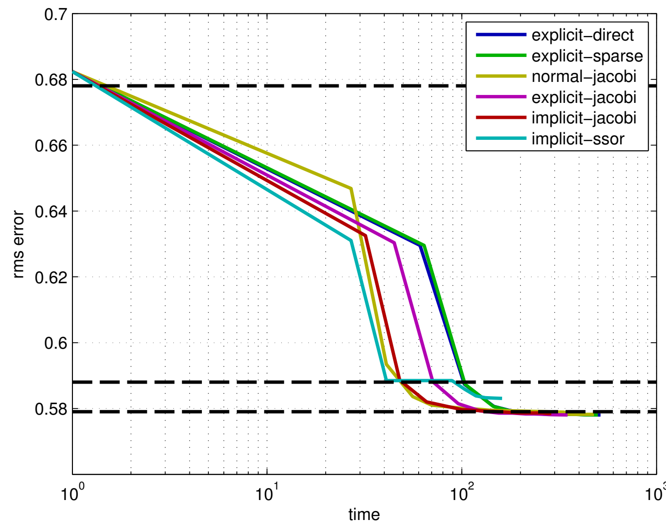
images = 1288, points = 866452, observations= 4383006



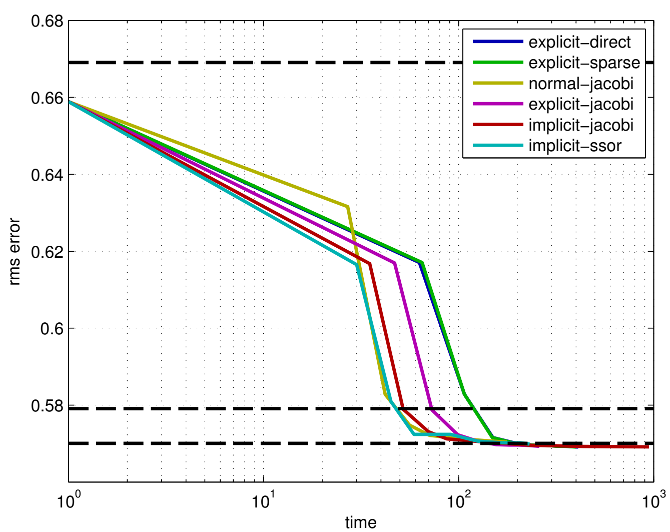
images = 1350, points = 894716, observations= 4517126



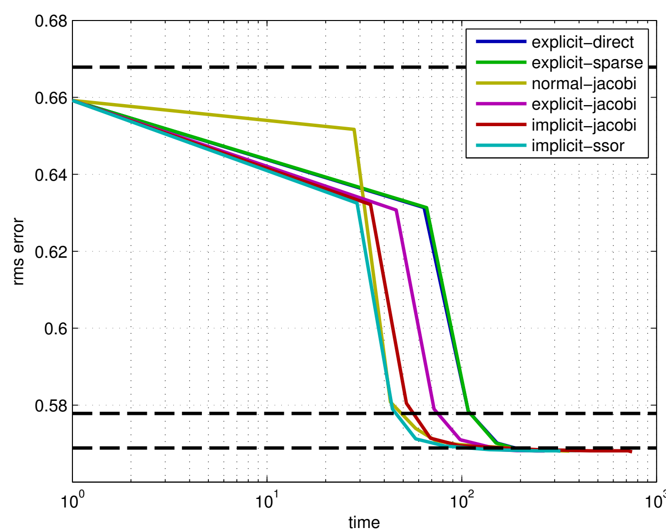
images = 1408, points = 912229, observations= 4634530



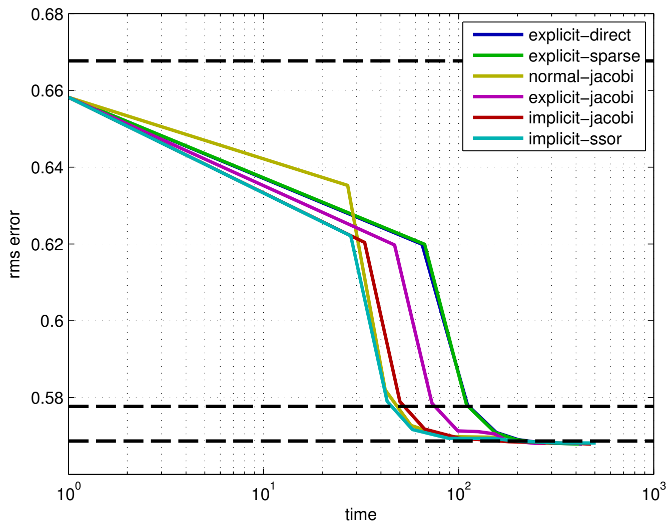
images = 1425, points = 916895, observations= 4657314



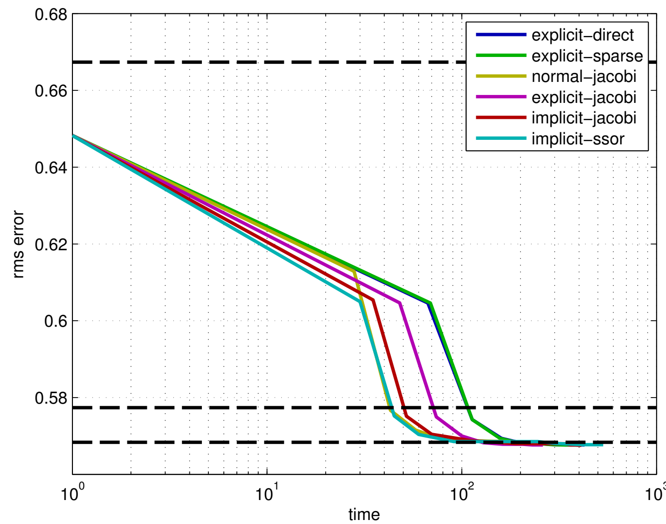
images = 1473, points = 930345, observations= 4705873



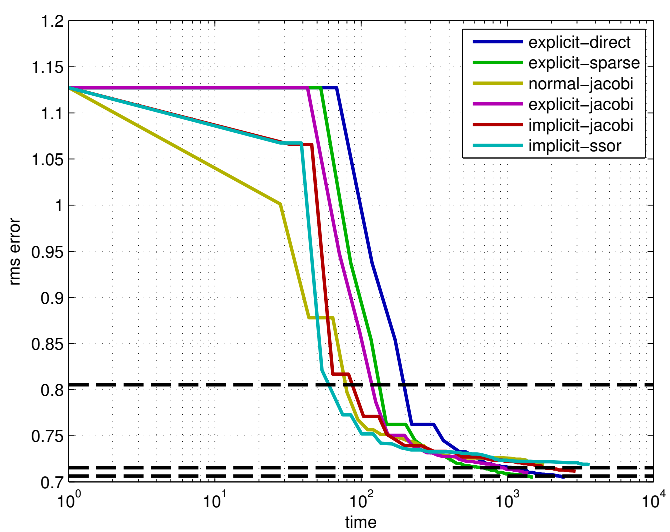
images = 1490, points = 935273, observations= 4721817



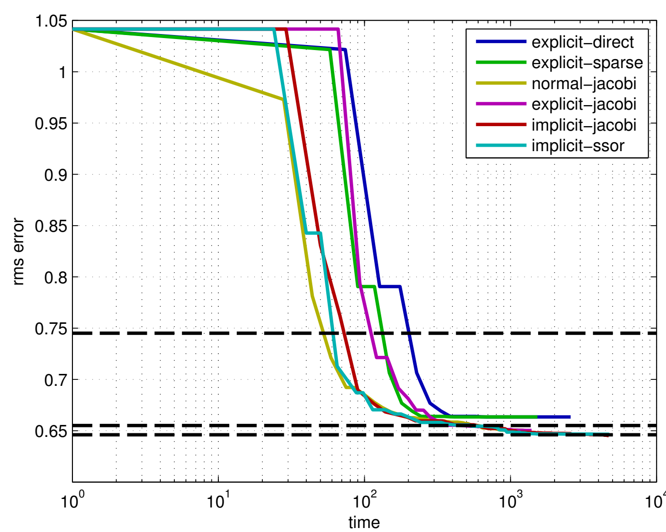
images = 1521, points = 939551, observations= 4739031



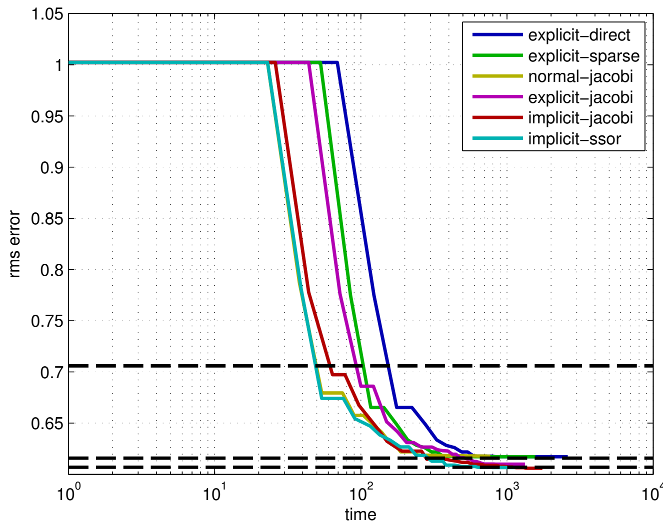
images = 1544, points = 942409, observations= 4750193



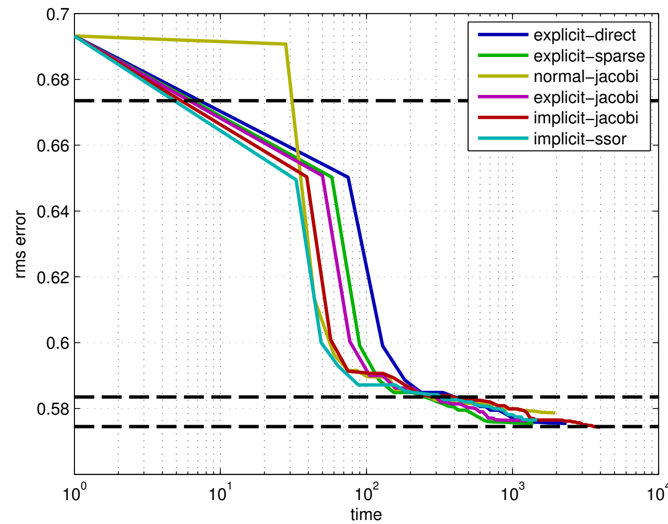
images = 1638, points = 976803, observations= 4956814



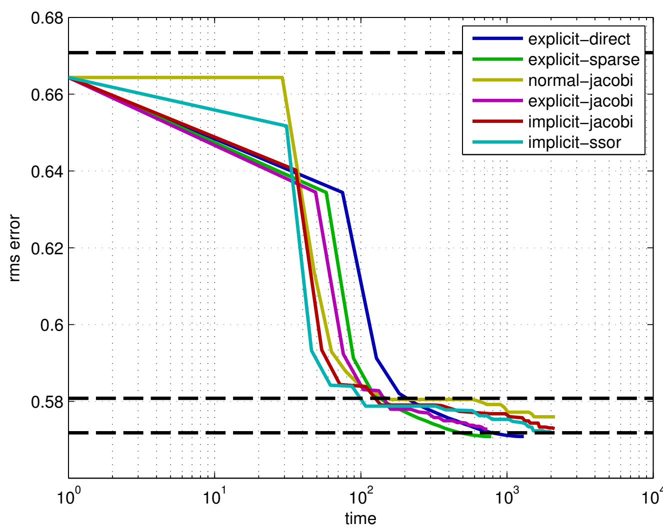
images = 1666, points = 983911, observations= 4987145



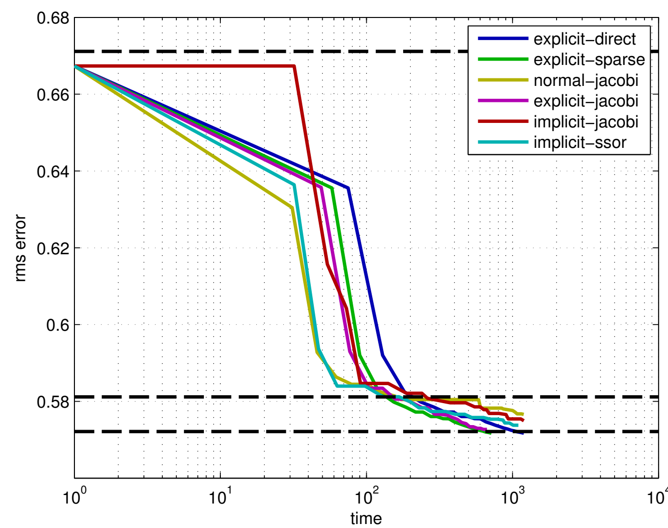
images = 1672, points = 986962, observations= 5000110



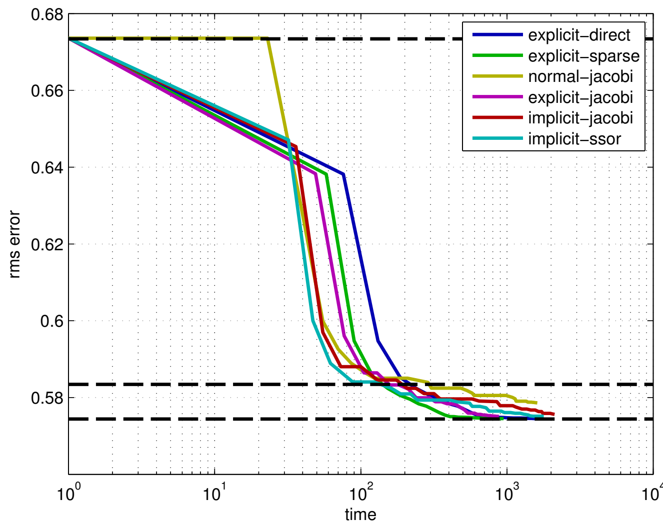
images = 1681, points = 983415, observations= 4966839



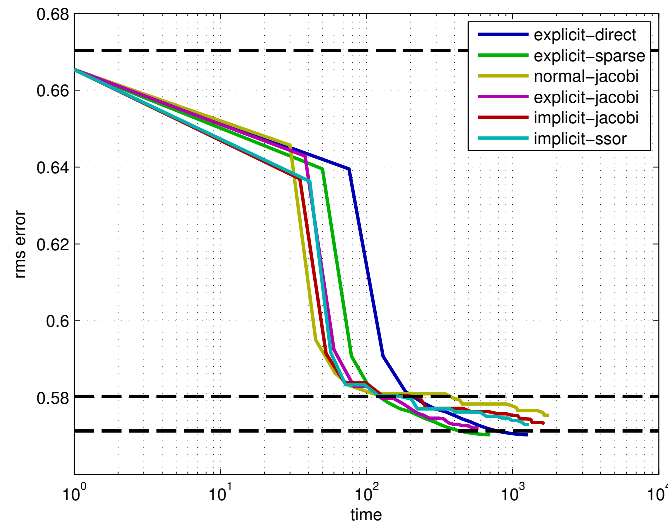
images = 1682, points = 983268, observations= 4965018



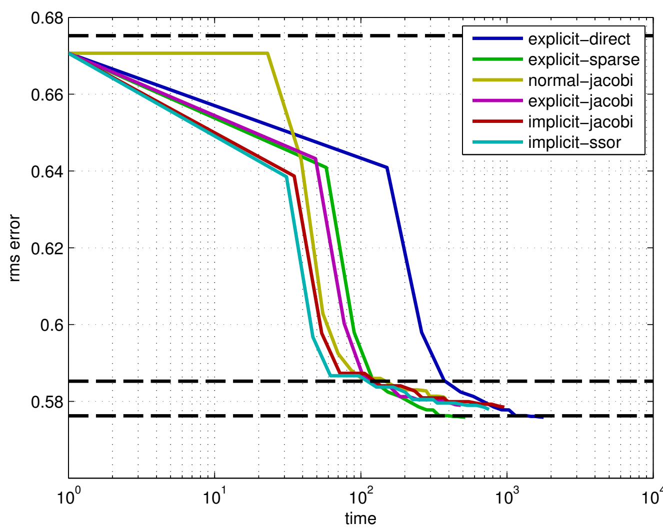
images = 1684, points = 983269, observations= 4965728



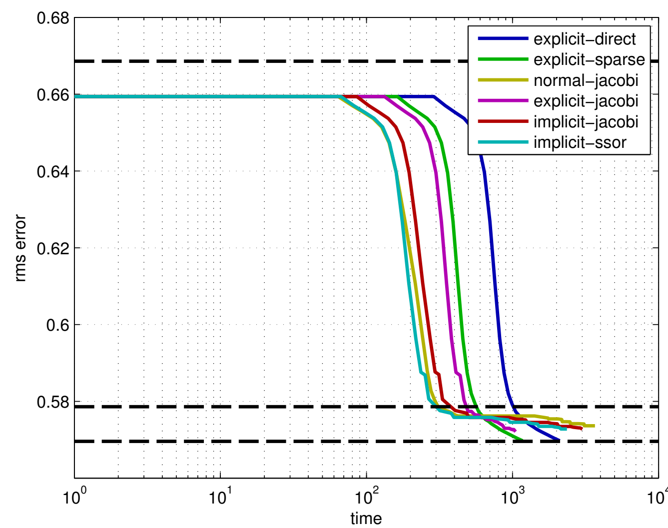
images = 1695, points = 984689, observations= 4970942



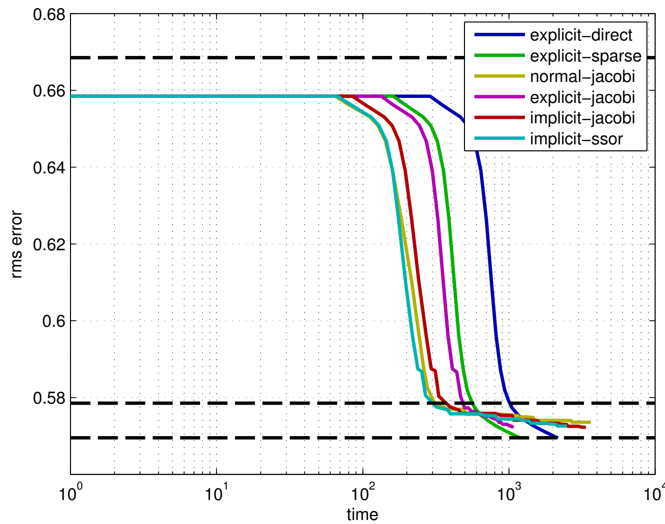
images = 1696, points = 984816, observations= 4970895



images = 1706, points = 985529, observations= 4974631



images = 1776, points = 993909, observations= 5001859



images = 1778, points = 993923, observations= 5001946

4 Final Bundles

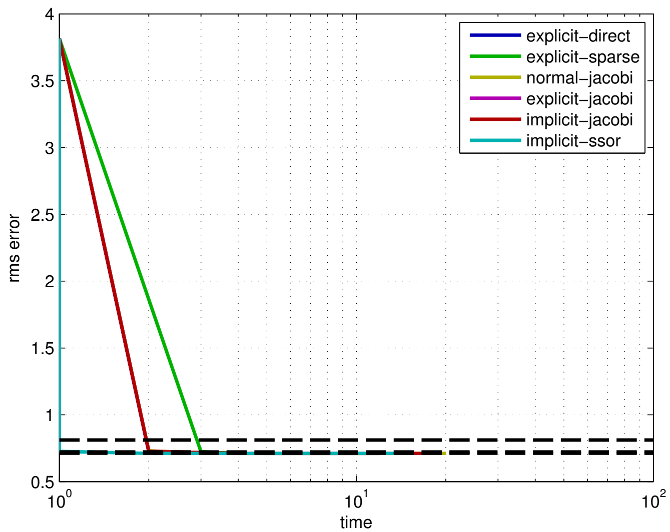
This section presents the results for bundle adjustment on problems generated by the skeletal sets algorithm when all the leaf images are added back to the skeleton and the remaining point tracks are triangulated. In cases when the algorithm ran out of memory or was unable to finish in a reasonable amount of time, we indicate the corresponding entry by NA.

Table 7: Problem statistics and memory usages for Bundle adjustment after Skeletal Sets

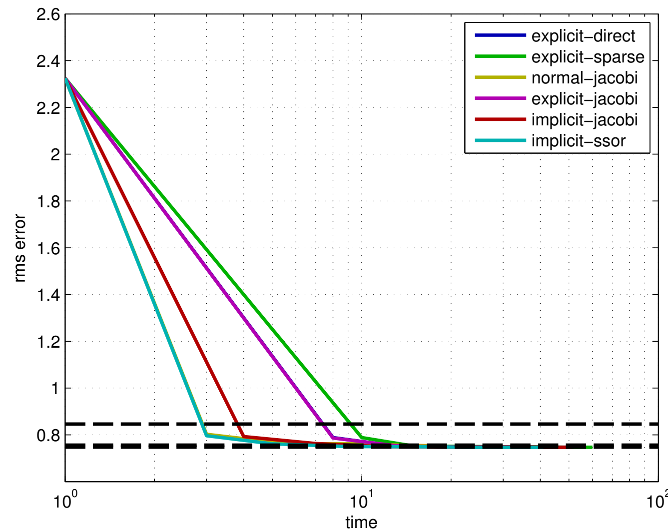
					Memory Usage (Bytes)					
images	points	obs	sparsity(S)	sparsity(L)	explicit-direct	explicit-sparse	explicit-jacobi	implicit-jacobi	implicit-ssor	normal-jacobi
93	61203	287451	0.00	0.00	1.8e+08	1.9e+08	1.8e+08	1.2e+08	1.2e+08	1.2e+08
394	100368	534408	0.06	0.00	5.1e+08	6.5e+08	4.2e+08	2.1e+08	2.3e+08	2.3e+08
871	527480	2785977	0.60	0.07	2.3e+09	2.6e+09	1.8e+09	1.1e+09	1.2e+09	1.2e+09
961	187103	1692975	0.01	0.00	2.2e+09	3.0e+09	1.6e+09	6.0e+08	6.4e+08	6.3e+08
1936	649673	5213733	0.03	0.00	7.8e+09	1.1e+10	5.4e+09	1.9e+09	2.0e+09	2.0e+09
3068	310854	1653812	0.79	0.19	8.4e+09	1.3e+10	2.4e+09	6.6e+08	7.6e+08	7.5e+08
4585	1324582	9125125	0.83	0.62	2.1e+10	1.6e+10	7.7e+09	3.4e+09	3.6e+09	3.6e+09
13682	4456117	28987644	0.86	NA	NA	NA	3.5e+10	1.1e+10	1.2e+10	1.2e+10

Table 8: Perfomance statistics for Bundle adjustment after Skeletal Sets

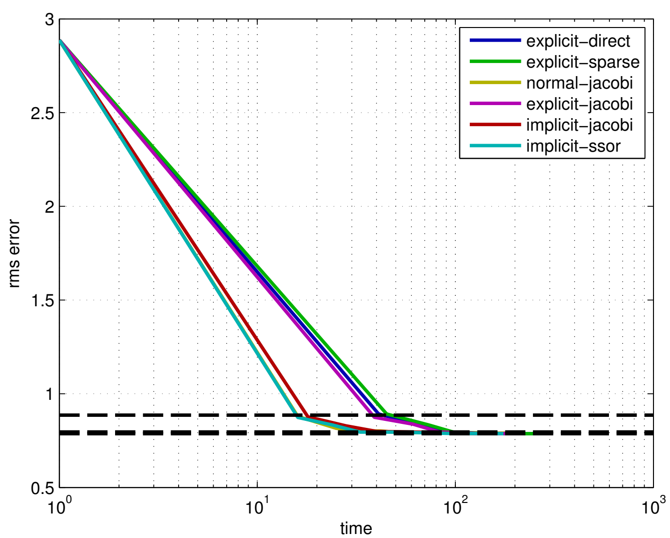
				explicit-direct		explicit-sparse		explicit-jacobi		implicit-jacobi		implicit-ssor		normal-jacobi	
images	points	obs	initial	time	error	time	error	time	error	time	error	time	error	time	error
93	61203	287451	3.815	4	0.712	4	0.712	5	0.712	5	0.712	6	0.712	2	0.712
394	100368	534408	2.325	38	0.747	44	0.747	68	0.746	56	0.747	48	0.747	27	0.746
871	527480	2785977	2.886	606	0.787	651	0.787	422	0.787	1299	0.787	424	0.787	309	0.787
961	187103	1692975	3.110	1551	0.968	1709	0.968	607	0.968	130	0.968	225	0.968	130	0.968
1936	649673	5213733	3.045	804	0.945	925	0.945	705	0.945	364	0.947	107	0.945	127	0.944
3068	310854	1653812	3.157	*NA*	*NA*	7448	0.969	1316	0.973	706	0.969	518	0.976	683	0.974
4585	1324582	9125125	2.160	*NA*	*NA*	4970	0.744	3494	0.748	5650	0.747	3473	0.745	2772	0.750
13682	4456117	28987644	2.543	*NA*	*NA*	*NA*	*NA*	21784	0.869	3668	0.864	5311	0.890	1644	0.889



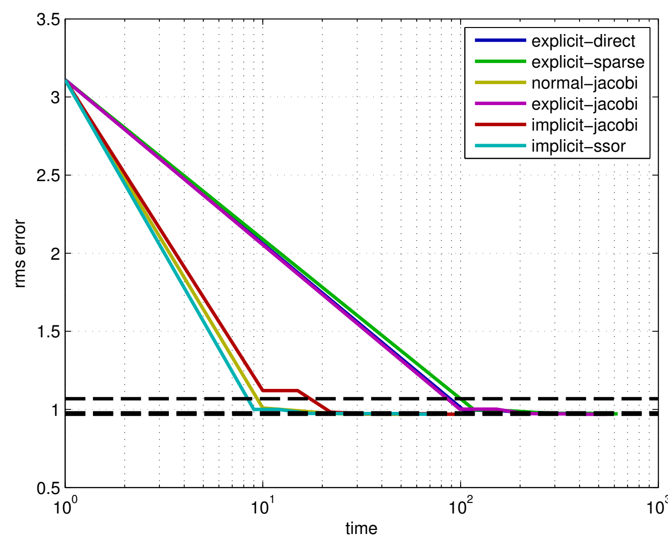
images = 93, points = 61203, observations= 287451



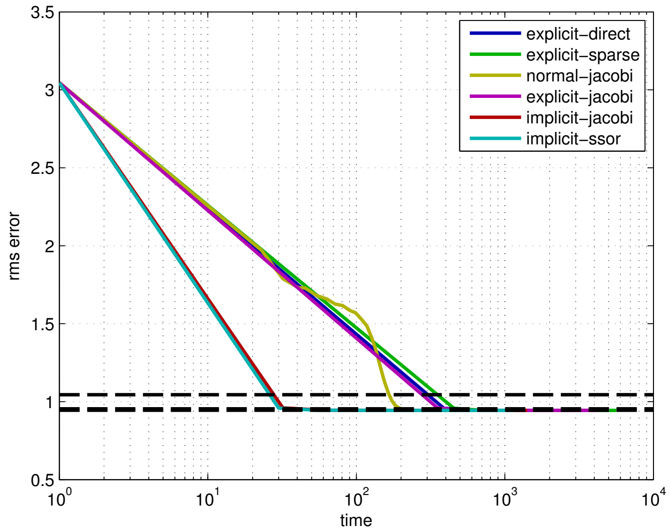
images = 394, points = 100368, observations= 534408



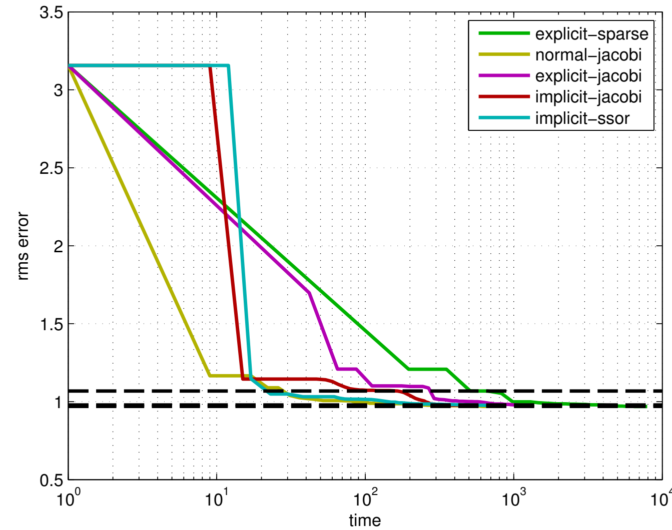
images = 871, points = 527480, observations= 2785977



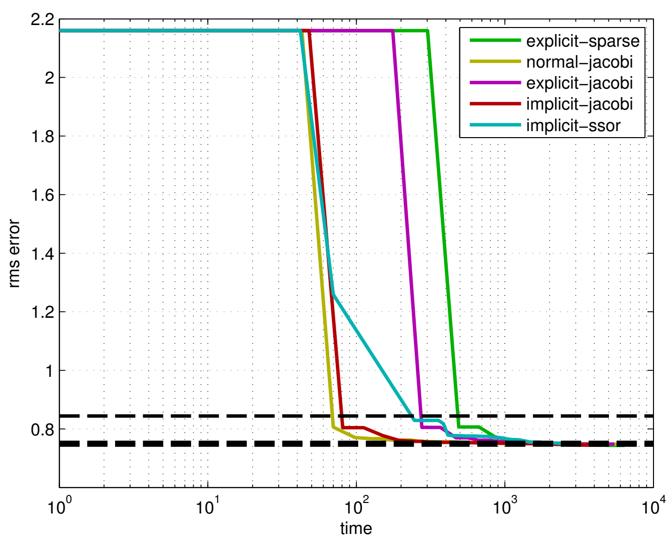
images = 961, points = 187103, observations= 1692975



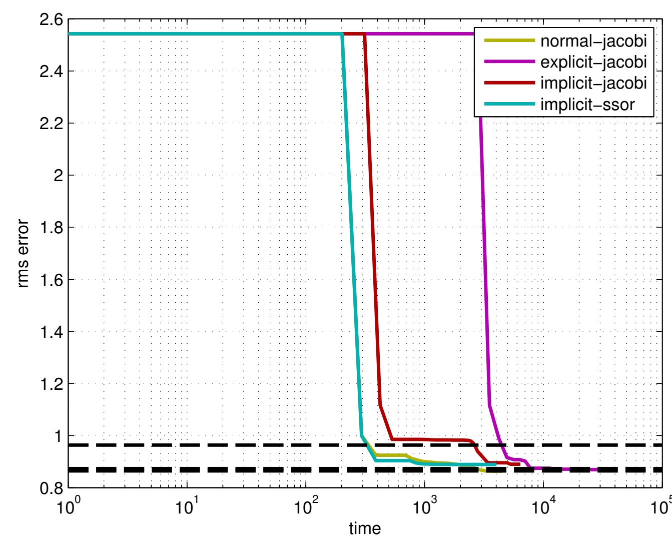
images = 1936, points = 649673, observations= 5213733



images = 3068, points = 310854, observations= 1653812



images = 4585, points = 1324582, observations= 9125125



images = 13682, points = 4456117, observations= 28987644

5 Streetview

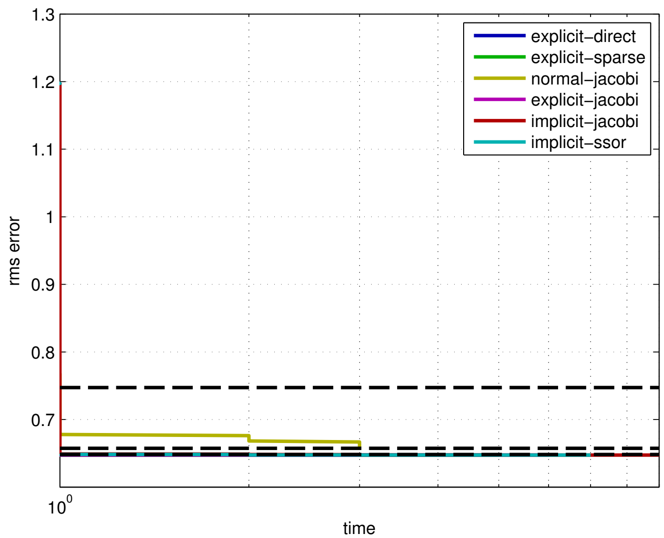
This section presents the results for bundle adjustment on problems generated during the incremental reconstruction of a dataset obtained by driving a Ladybug camera mounted on a truck down an urban street.

Table 9: Problem statistics and memory usages for the incremental reconstruction of the Streetview dataset

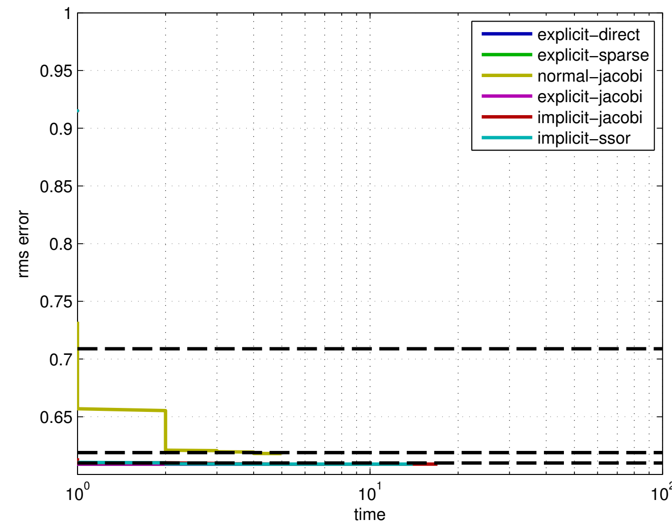
					Memory Usage (Bytes)					
images	points	obs	sparsity(S)	sparsity(L)	explicit-direct	explicit-sparse	explicit-jacobi	implicit-jacobi	implicit-ssor	normal-jacobi
49	7776	31843	0.16	0.00	2.2e+07	2.4e+07	2.1e+07	1.4e+07	1.5e+07	1.5e+07
73	11032	46122	0.28	0.00	3.4e+07	3.8e+07	3.1e+07	2.0e+07	2.1e+07	2.1e+07
138	19878	85217	0.43	0.01	7.1e+07	8.2e+07	6.0e+07	3.6e+07	3.9e+07	3.9e+07
318	41628	179919	0.56	0.05	2.0e+08	2.4e+08	1.5e+08	8.0e+07	8.6e+07	8.7e+07
372	47423	204472	0.60	0.09	2.5e+08	3.0e+08	1.7e+08	9.1e+07	9.7e+07	9.9e+07
412	52215	224242	0.63	0.13	2.9e+08	3.9e+08	1.9e+08	9.9e+07	1.0e+08	1.1e+08
460	56811	241877	0.67	0.20	3.3e+08	4.2e+08	2.0e+08	1.1e+08	1.1e+08	1.2e+08
539	65220	277273	0.71	0.25	4.1e+08	4.6e+08	2.4e+08	1.3e+08	1.4e+08	1.3e+08
598	69218	304170	0.74	0.29	4.8e+08	4.7e+08	2.6e+08	1.4e+08	1.5e+08	1.5e+08
646	73584	327297	0.75	0.31	5.4e+08	5.3e+08	2.8e+08	1.4e+08	1.5e+08	1.5e+08
707	78455	349940	0.78	0.37	6.1e+08	5.7e+08	3.0e+08	1.6e+08	1.7e+08	1.7e+08
783	84444	377052	0.80	0.28	7.1e+08	7.9e+08	3.3e+08	1.6e+08	1.9e+08	1.8e+08
810	88814	393775	0.81	0.29	7.5e+08	8.2e+08	3.4e+08	1.8e+08	1.9e+08	1.9e+08
856	93344	415769	0.82	0.59	8.1e+08	6.6e+08	3.6e+08	1.9e+08	2.0e+08	2.0e+08
885	97473	434905	0.83	0.58	8.6e+08	6.9e+08	3.7e+08	2.0e+08	2.0e+08	2.1e+08
931	102699	457460	0.84	0.61	9.3e+08	6.8e+08	3.8e+08	2.0e+08	2.2e+08	2.2e+08
969	105826	474627	0.85	0.61	9.9e+08	7.4e+08	3.9e+08	2.1e+08	2.3e+08	2.3e+08
1031	110968	500265	0.86	0.64	1.1e+09	8.0e+08	4.2e+08	2.3e+08	2.4e+08	2.4e+08
1064	113655	510211	0.86	0.66	1.1e+09	8.1e+08	4.4e+08	2.2e+08	2.4e+08	2.4e+08
1118	118384	528926	0.87	0.64	1.2e+09	8.9e+08	4.6e+08	2.3e+08	2.6e+08	2.6e+08
1152	122269	545819	0.88	0.67	1.3e+09	8.5e+08	4.6e+08	2.4e+08	2.5e+08	2.7e+08
1197	126327	563734	0.88	0.69	1.4e+09	8.7e+08	4.7e+08	2.6e+08	2.8e+08	2.7e+08
1235	129634	576286	0.89	0.68	1.5e+09	8.8e+08	4.9e+08	2.5e+08	2.8e+08	2.8e+08
1266	132593	587942	0.89	0.67	1.5e+09	9.5e+08	5.0e+08	2.6e+08	2.8e+08	2.9e+08
1340	137079	612593	0.89	0.68	1.7e+09	1.0e+09	5.4e+08	2.7e+08	3.0e+08	3.0e+08
1469	145199	641646	0.90	0.70	1.9e+09	1.1e+09	5.8e+08	3.0e+08	3.1e+08	3.2e+08
1514	147317	651480	0.90	0.71	2.0e+09	1.1e+09	5.9e+08	3.0e+08	3.1e+08	3.2e+08
1587	150845	663289	0.91	0.72	2.2e+09	1.1e+09	6.0e+08	3.0e+08	3.1e+08	3.3e+08
1642	153820	671268	0.91	0.73	2.3e+09	1.2e+09	6.1e+08	3.1e+08	3.4e+08	3.3e+08
1695	155710	676595	0.92	0.75	2.4e+09	1.1e+09	6.2e+08	3.1e+08	3.3e+08	3.4e+08

Table 10: Perfomance statistics for the incremental reconstruction of Streetview dataset

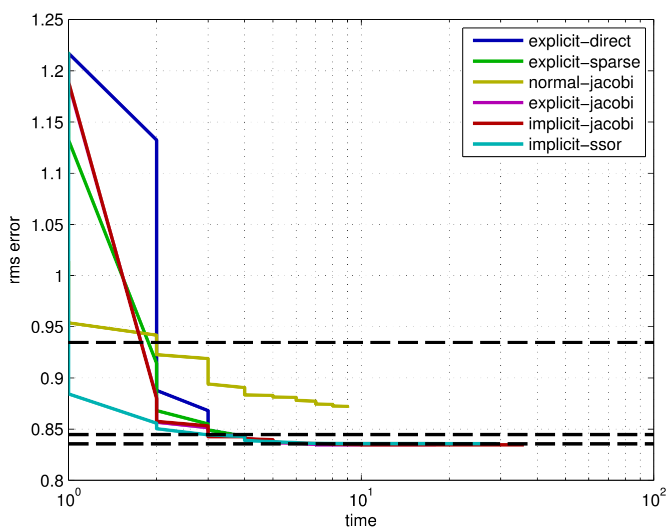
				explicit-direct		explicit-sparse		explicit-jacobi		implicit-jacobi		implicit-ssor		normal-jacobi	
images	points	obs	initial	time	error	time	error	time	error	time	error	time	error	time	error
49	7776	31843	1.197	2	0.647	2	0.647	1	0.647	3	0.659	4	0.647	3	0.647
73	11032	46122	0.915	1	0.609	1	0.609	1	0.609	5	0.618	6	0.609	4	0.609
138	19878	85217	1.217	8	0.836	9	0.836	8	0.835	9	0.871	12	0.835	9	0.836
318	41628	179919	0.715	36	0.693	44	0.693	40	0.693	20	0.693	29	0.693	37	0.693
372	47423	204472	0.833	55	0.743	65	0.743	45	0.743	20	0.745	41	0.743	33	0.743
412	52215	224242	0.740	61	0.703	73	0.703	35	0.703	24	0.702	46	0.703	33	0.703
460	56811	241877	0.826	60	0.730	66	0.730	51	0.730	26	0.734	92	0.730	65	0.730
539	65220	277273	0.899	17	0.742	19	0.742	11	0.742	30	0.743	16	0.742	11	0.742
598	69218	304170	0.861	80	0.739	72	0.739	110	0.738	33	0.740	379	0.738	521	0.738
646	73584	327297	0.813	100	0.743	86	0.743	164	0.743	33	0.745	370	0.743	278	0.742
707	78455	349940	1.041	89	0.751	75	0.751	105	0.751	39	0.758	292	0.750	465	0.750
783	84444	377052	0.733	168	0.725	156	0.725	177	0.725	39	0.728	1003	0.725	488	0.725
810	88814	393775	0.725	207	0.719	188	0.719	90	0.719	42	0.720	262	0.719	211	0.719
856	93344	415769	0.725	233	0.714	134	0.714	171	0.714	44	0.714	426	0.714	292	0.714
885	97473	434905	0.729	254	0.712	138	0.712	183	0.712	47	0.712	509	0.712	332	0.712
931	102699	457460	0.796	129	0.742	64	0.742	107	0.742	50	0.743	172	0.742	162	0.742
969	105826	474627	0.752	290	0.737	139	0.737	179	0.737	52	0.737	553	0.737	332	0.737
1031	110968	500265	0.747	388	0.742	174	0.742	190	0.742	56	0.742	482	0.742	367	0.742
1064	113655	510211	0.754	322	0.743	135	0.743	466	0.743	53	0.744	1470	0.743	1301	0.743
1118	118384	528926	0.809	232	0.752	94	0.752	225	0.752	56	0.754	765	0.752	628	0.752
1152	122269	545819	0.789	175	0.748	66	0.748	156	0.748	60	0.750	462	0.748	517	0.748
1197	126327	563734	0.843	581	0.752	196	0.752	114	0.752	56	0.753	169	0.752	142	0.752
1235	129634	576286	0.759	637	0.748	200	0.748	144	0.748	62	0.749	226	0.748	168	0.748
1266	132593	587942	0.765	648	0.748	204	0.748	208	0.749	65	0.749	648	0.749	434	0.749
1340	137079	612593	0.768	748	0.753	219	0.753	499	0.753	67	0.754	1113	0.753	1077	0.753
1469	145199	641646	0.767	872	0.749	217	0.749	202	0.749	70	0.749	435	0.749	205	0.749
1514	147317	651480	0.757	759	0.746	178	0.746	197	0.746	72	0.746	659	0.746	496	0.746
1587	150845	663289	0.799	1078	0.750	221	0.750	139	0.750	73	0.751	474	0.750	505	0.750
1642	153820	671268	0.752	1367	0.746	275	0.746	316	0.746	74	0.746	623	0.746	652	0.746
1695	155710	676595	0.771	1474	0.746	262	0.746	170	0.747	71	0.747	454	0.746	424	0.746



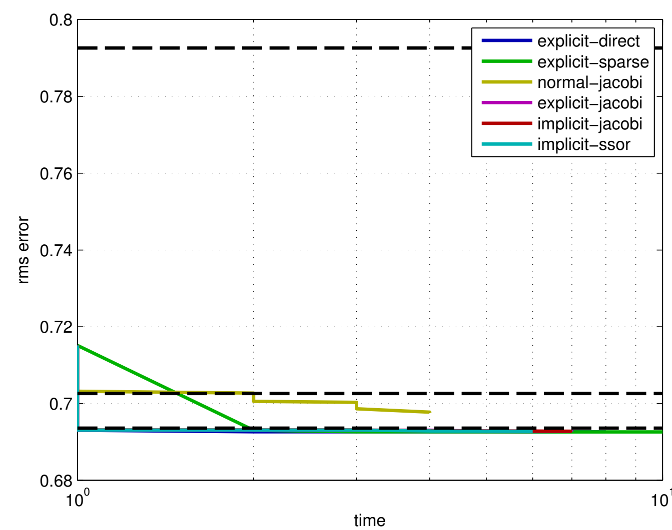
images = 49, points = 7776, observations= 31843



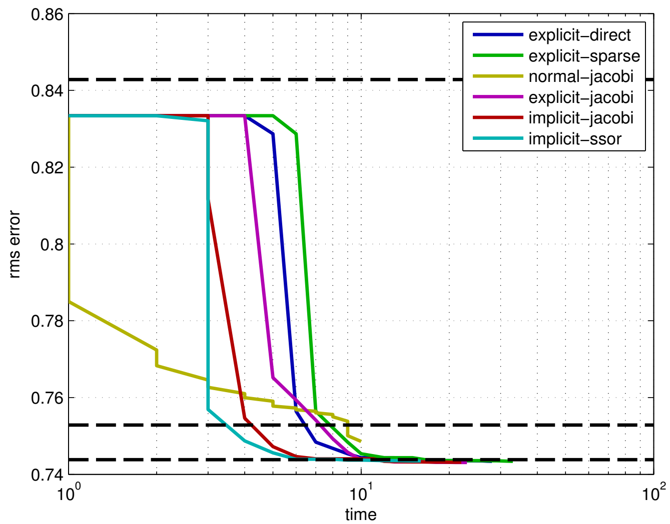
images = 73, points = 11032, observations= 46122



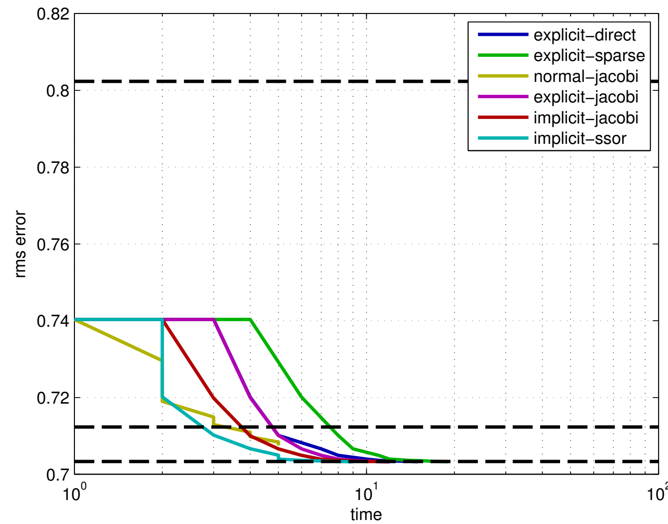
images = 138, points = 19878, observations= 85217



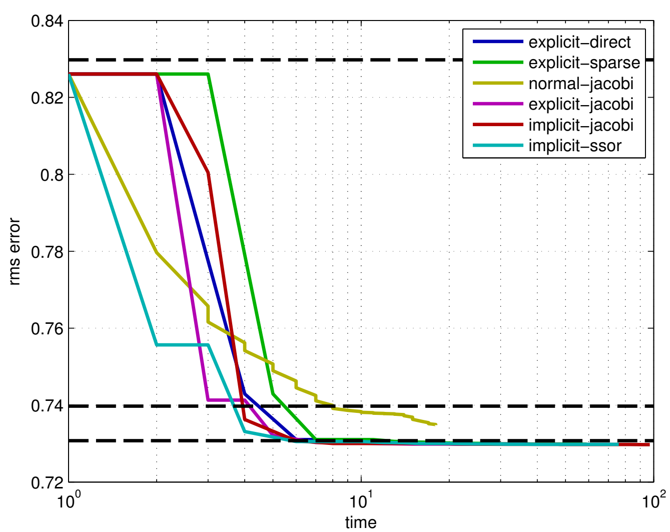
images = 318, points = 41628, observations= 179919



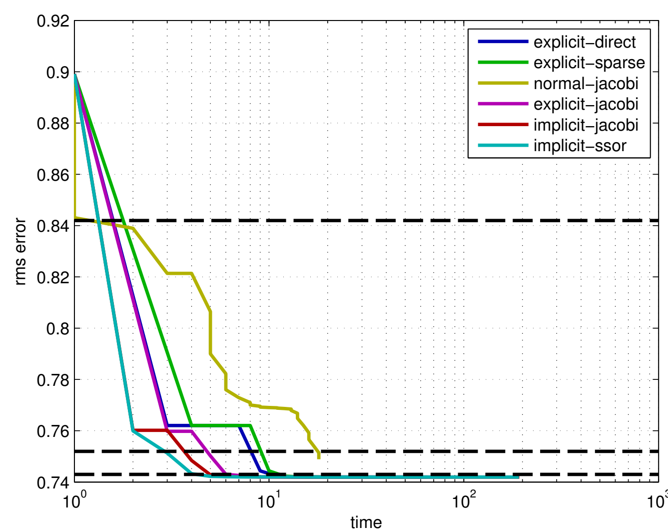
images = 372, points = 47423, observations= 204472



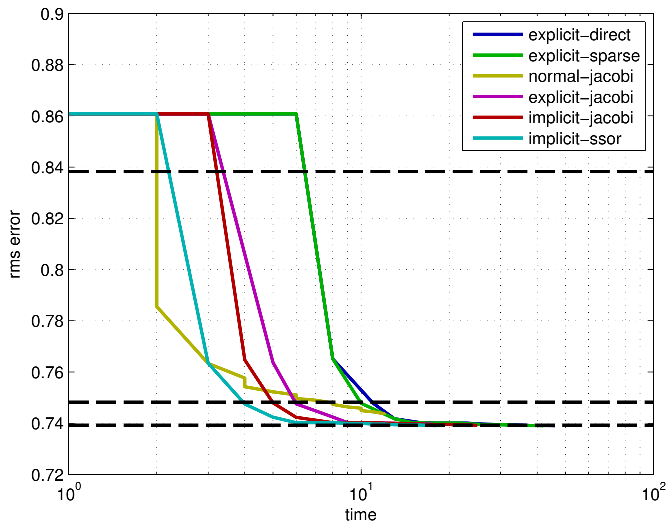
images = 412, points = 52215, observations= 224242



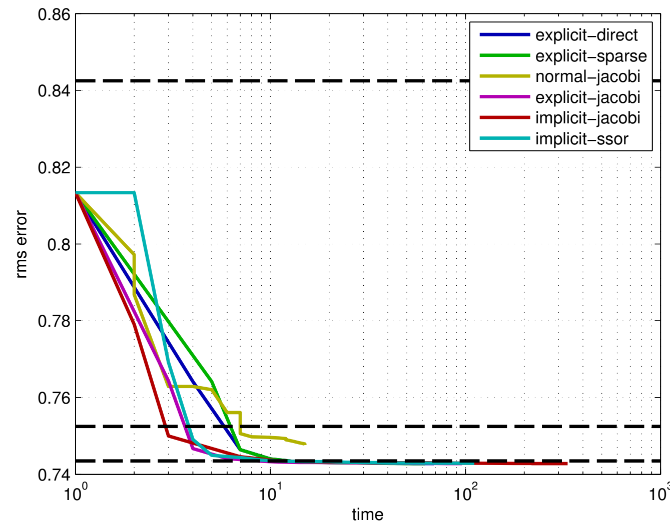
images = 460, points = 56811, observations= 241877



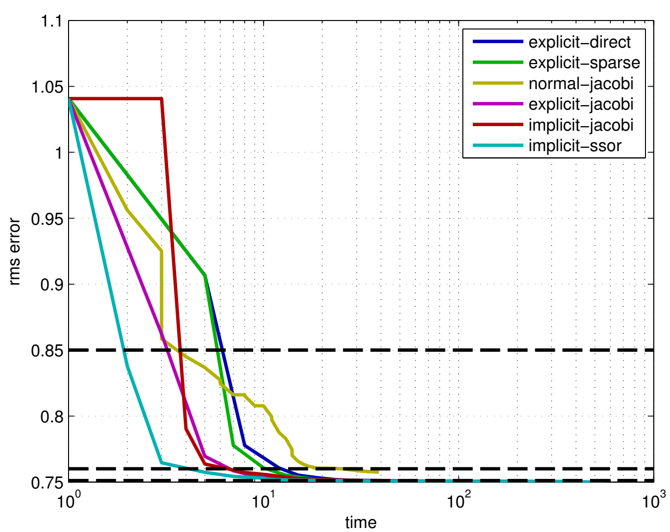
images = 539, points = 65220, observations= 277273



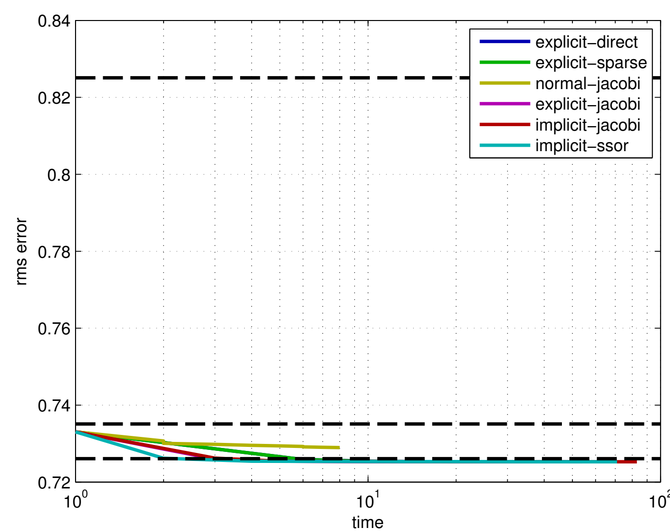
images = 598, points = 69218, observations= 304170



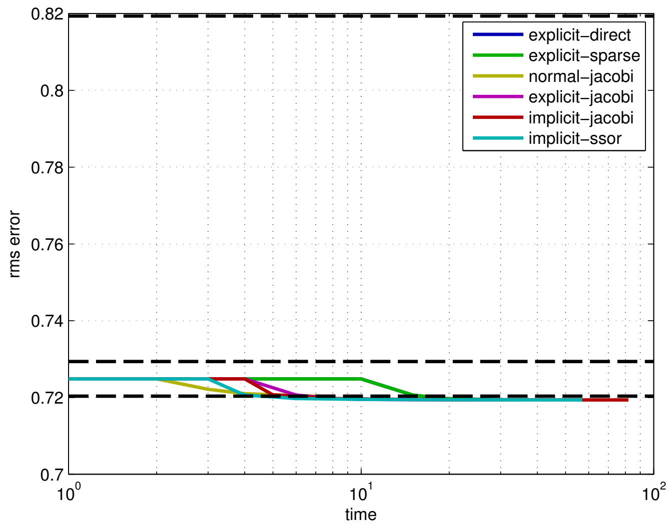
images = 646, points = 73584, observations= 327297



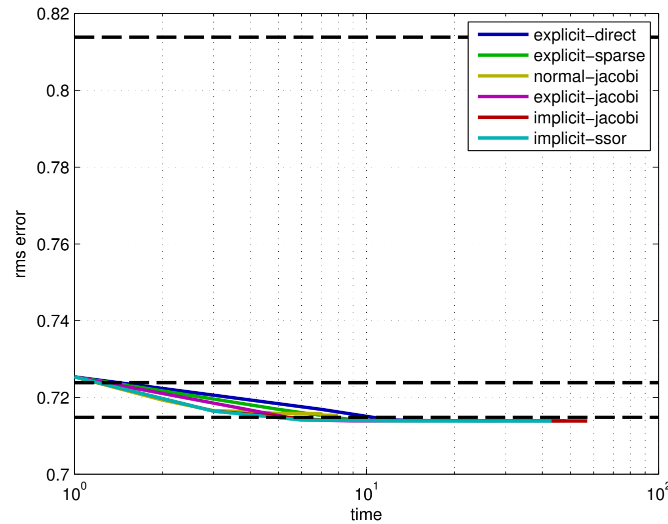
images = 707, points = 78455, observations= 349940



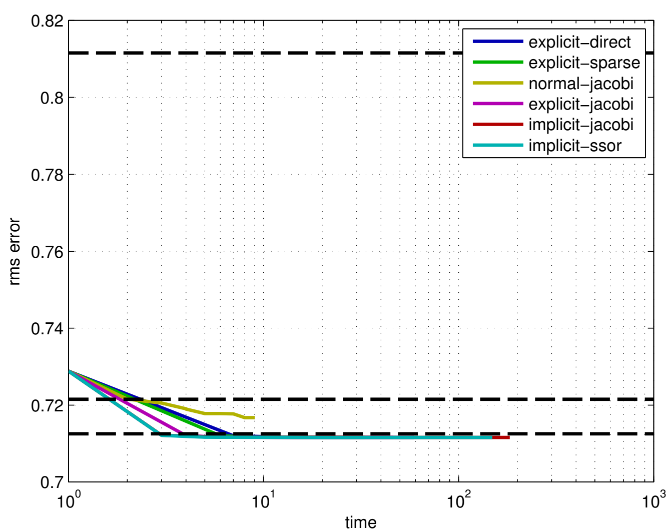
images = 783, points = 84444, observations= 377052



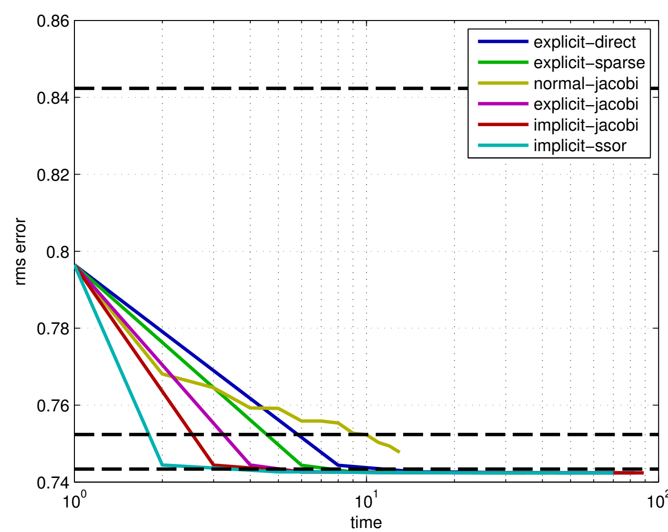
images = 810, points = 88814, observations= 393775



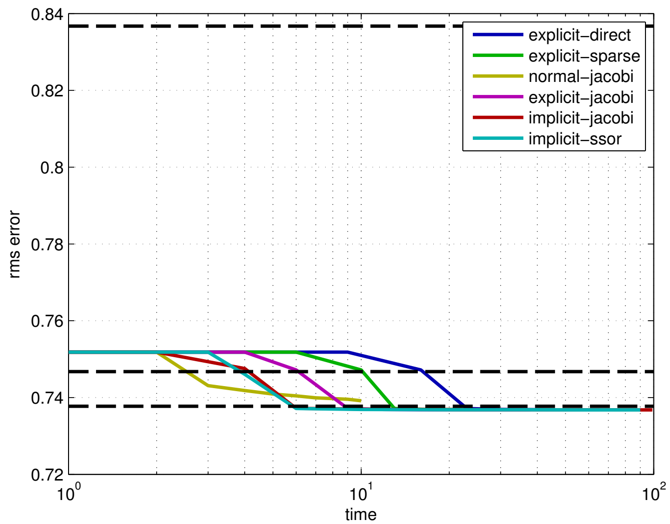
images = 856, points = 93344, observations= 415769



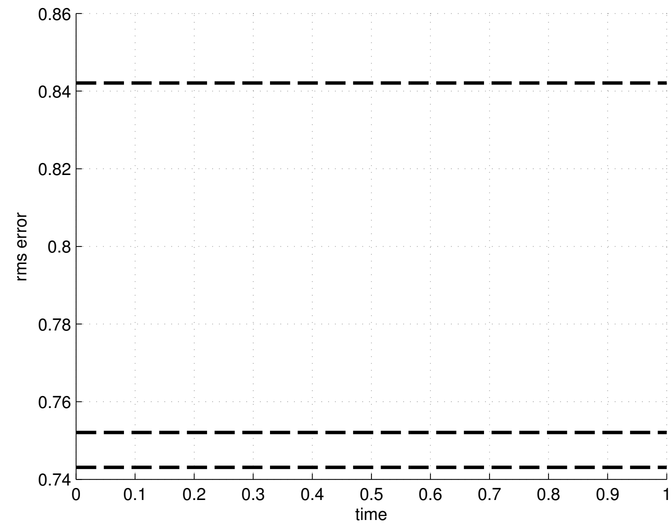
images = 885, points = 97473, observations= 434905



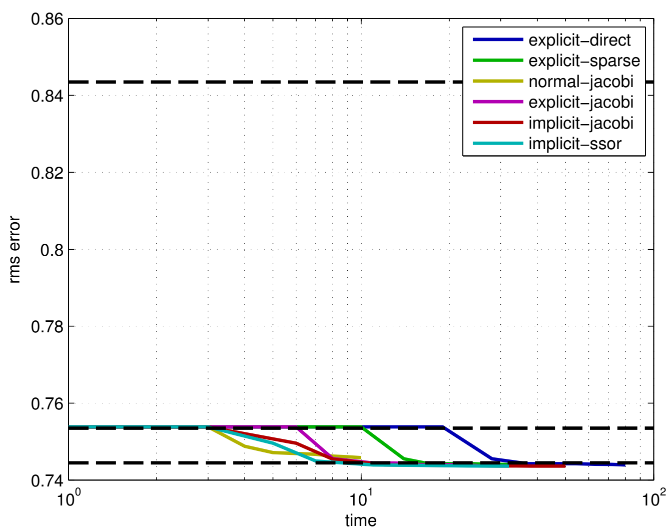
images = 931, points = 102699, observations= 457460



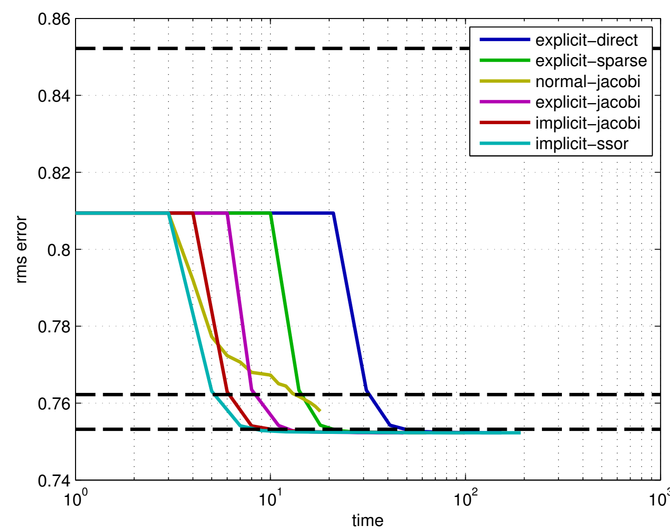
images = 969, points = 105826, observations= 474627



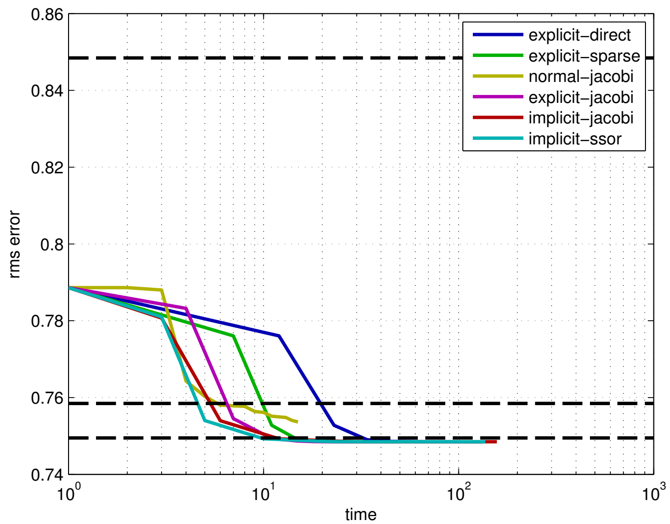
images = 1031, points = 110968, observations= 500265



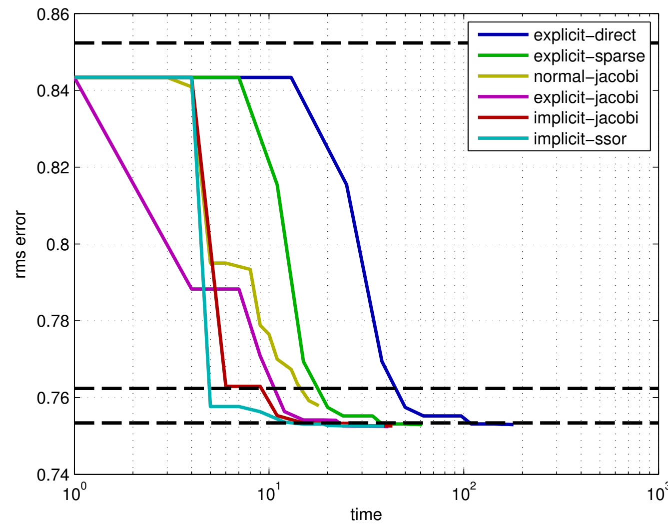
images = 1064, points = 113655, observations= 510211



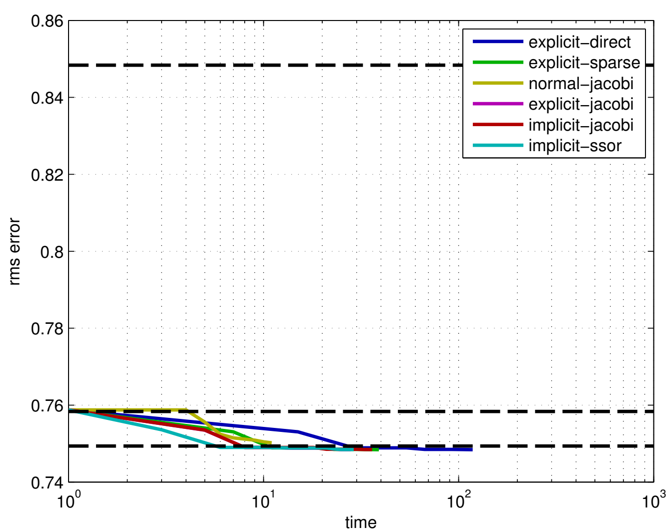
images = 1118, points = 118384, observations= 528926



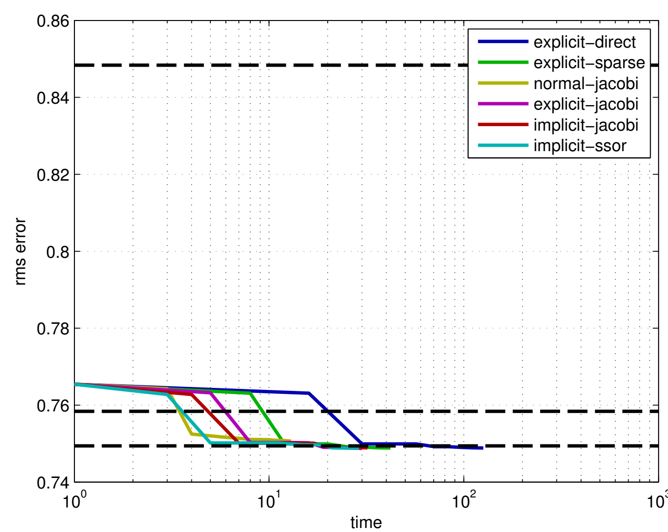
images = 1152, points = 122269, observations= 545819



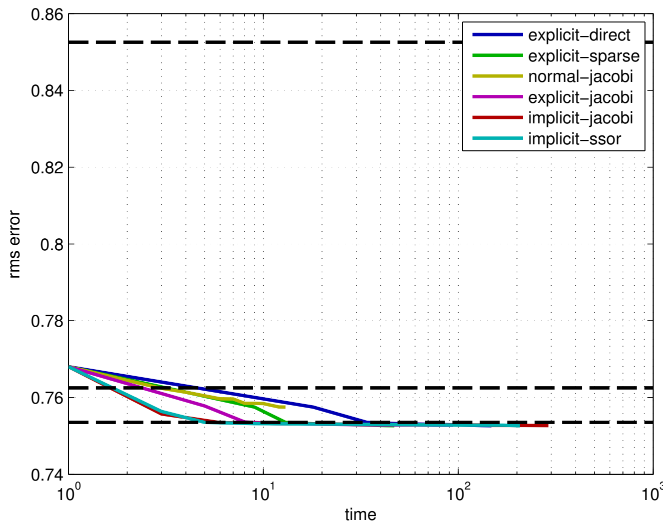
images = 1197, points = 126327, observations= 563734



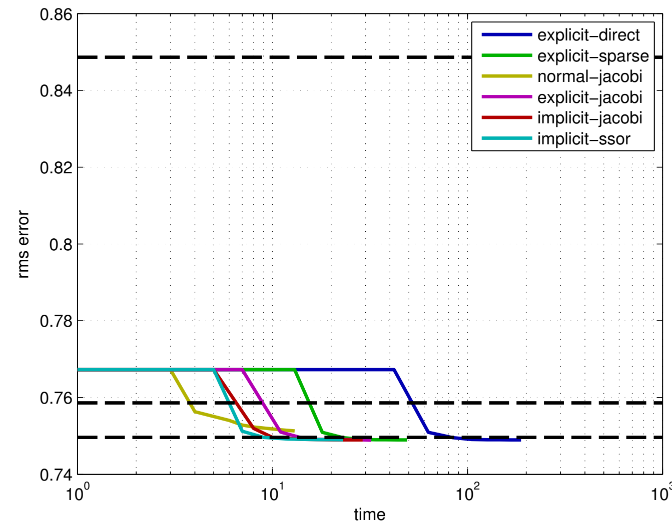
images = 1235, points = 129634, observations= 576286



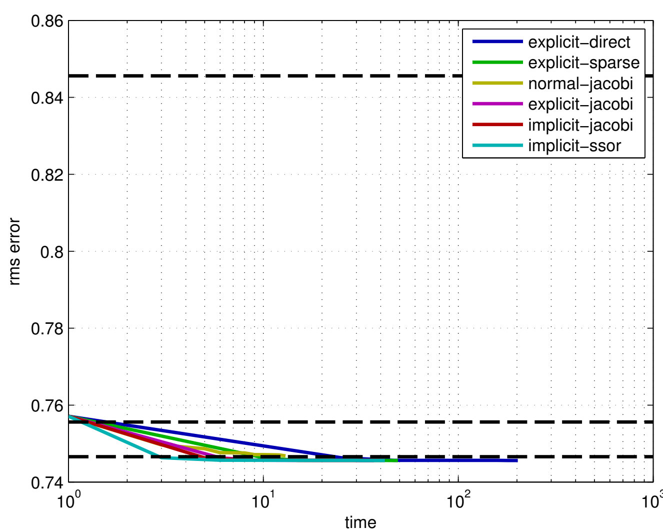
images = 1266, points = 132593, observations= 587942



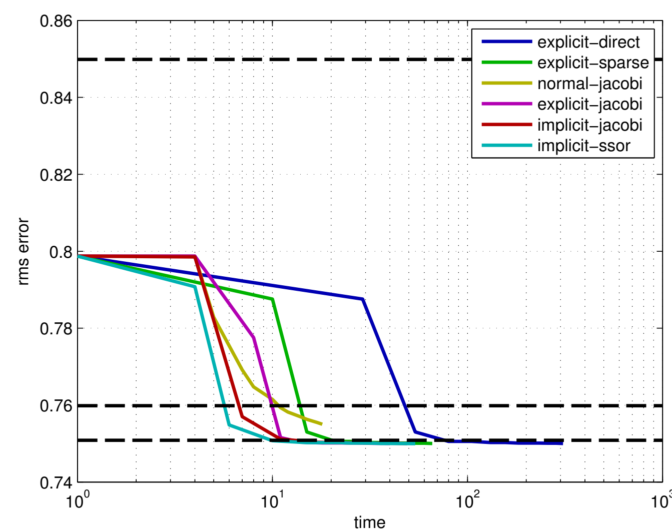
images = 1340, points = 137079, observations= 612593



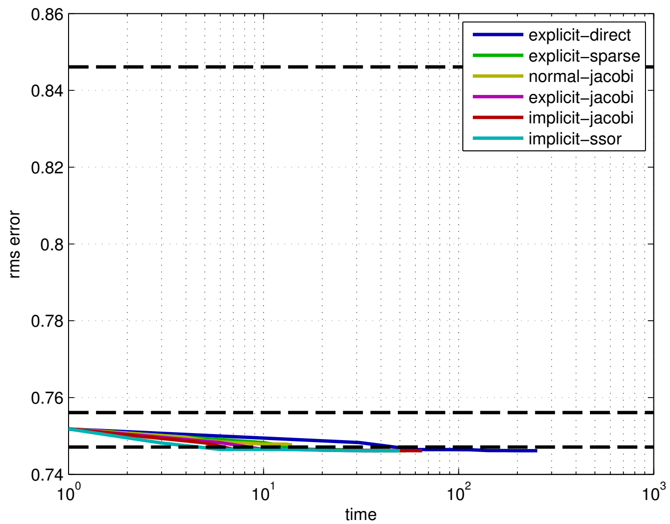
images = 1469, points = 145199, observations= 641646



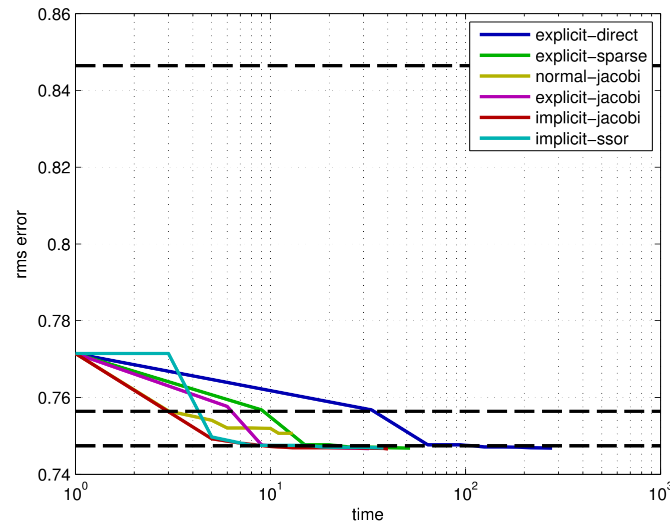
images = 1514, points = 147317, observations= 651480



images = 1587, points = 150845, observations= 663289



images = 1642, points = 153820, observations= 671268



images = 1695, points = 155710, observations= 676595

6 Overall Performance

We also reproduce here in Table 1 for reasons for clarity a larger version of the main figure from the paper.

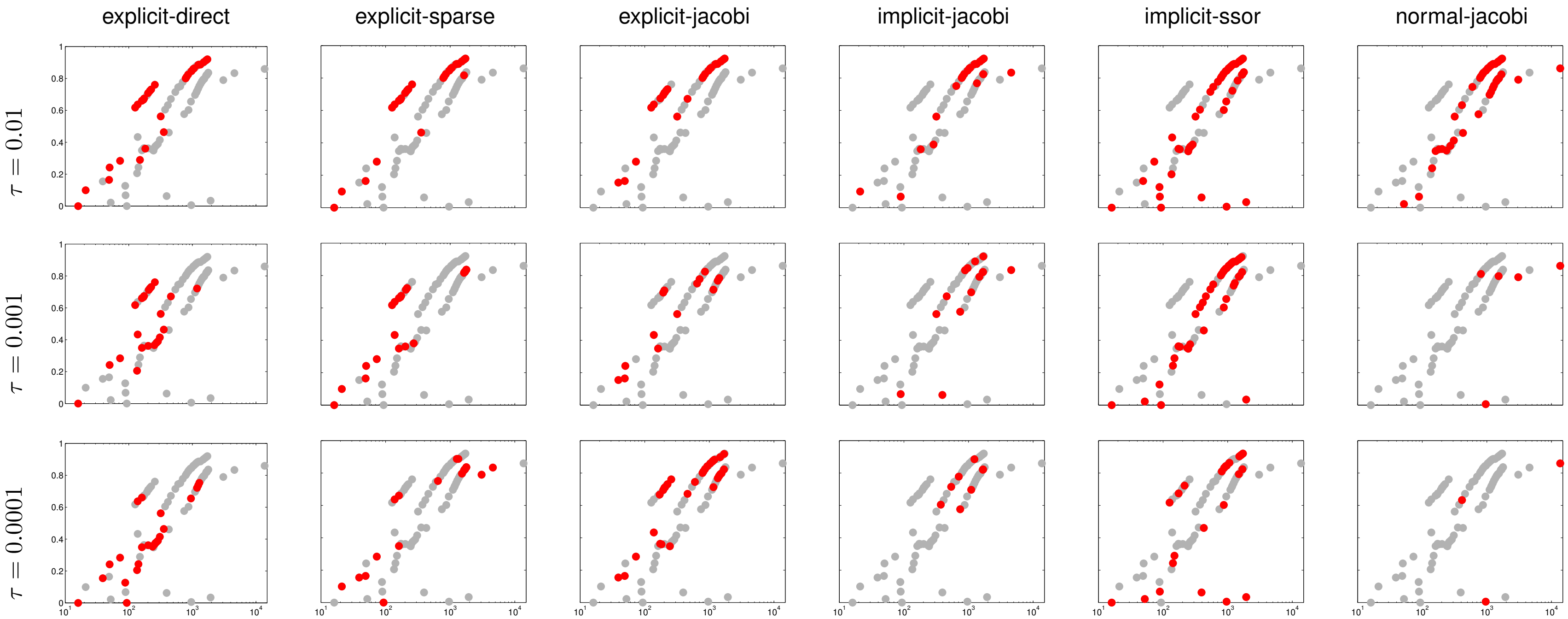


Figure 1: Performance analysis. Each column in this set of plots corresponds to one of six algorithms, and each row corresponds to one of three tolerances τ, where an algorithm gets to within tolerance when it is within τ of the best answer found over all algorithms. The axes of the individual plots are the same as in Figure 2 in the main paper. For each solver (column), a point is colored red if the solver was declared a winner for the given tolerance, and gray otherwise. See the text for discussion of these plots.

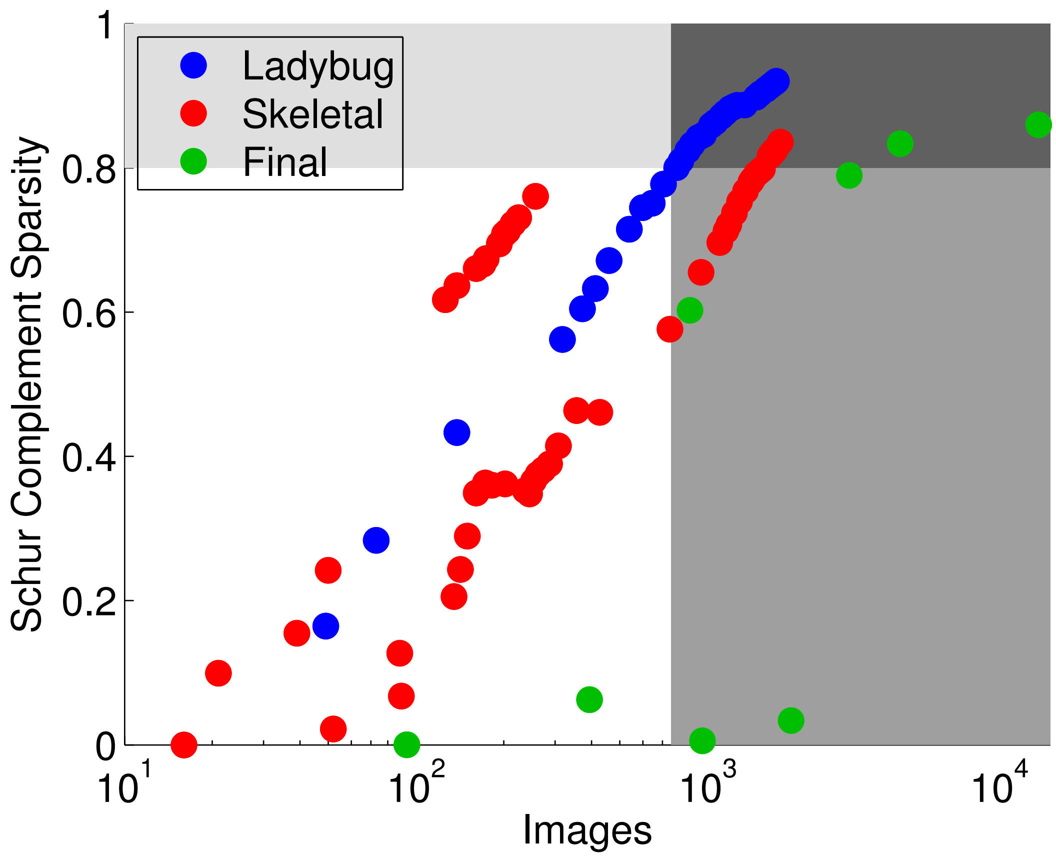


Figure 2: *Datasets. This scatter plot shows each of the datasets in our testbed, colored according to type (streetview, skeletal, and final), and plotted according to the number of images in the problem (x-axis, shown as a log-scale) and the sparsity of S (i.e., the fraction of entries that are zero). The plot is shaded according to the characteristics of the problem: small and dense (white), then in increasing gray-level, small and sparse, large and dense, and large and sparse.*

7 Proofs

We present here an elementary proof of the following of the fact that the reduced camera matrix is better conditioned than the regularized hessian. This results motivates the use of the implicit Schur complement based iterations.

Theorem 1. *Given a block structured matrix* $A \succ 0$*,*

$$A = \begin{bmatrix} B & E \\ E^\top & C \end{bmatrix} \tag{1}$$

then,

$$\kappa(A) \geq \kappa(B - EC^{-1}E^\top) \tag{2}$$

Proof. The maximum and the minimum eigenvalues of a matrix A are given by

$$\lambda_{\max}(A) = \max_x \frac{x^\top Ax}{x^\top x}, \quad \lambda_{\min}(A) = \min_x \frac{x^\top Ax}{x^\top x} \tag{3}$$

Now if the vector x is written in block structured form $[y, z]$, where the block sizes correspond to the block structure of A, it is easily seen than

$$\max_{x, z=0} \frac{x^\top Ax}{x^\top x} = \lambda_{\max}(B) \leq \lambda_{\max}(A) \tag{4}$$

Similarly, $\lambda_{\min}(A) \leq \lambda_{\min}(B)$. Thus,

$$\kappa(A) = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)} \geq \frac{\lambda_{\max}(B}{\lambda_{\min}(B} = \kappa(B) \tag{5}$$

Now, consider the inverse of A in block form,

$$A^{-1} = \begin{bmatrix} (B - EC^{-1}E^\top)^{-1} & \star \\ \star & (C - E^\top B^{-1}E)^{-1} \end{bmatrix} \tag{6}$$

Here, we use $\star$, to indicate entries that we do not care about right now. Then, from (5) we have that $\kappa(A^{-1}) \geq \kappa(S^{-1})$. Combining this inequality with the fact that for all invertible matrices X, $\kappa(X) = \kappa(X^{-1})$ proves the theorem. □