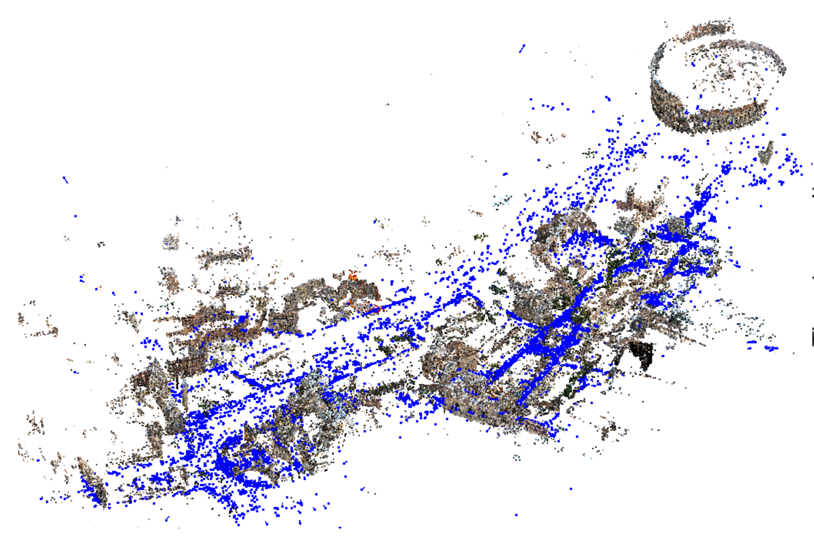
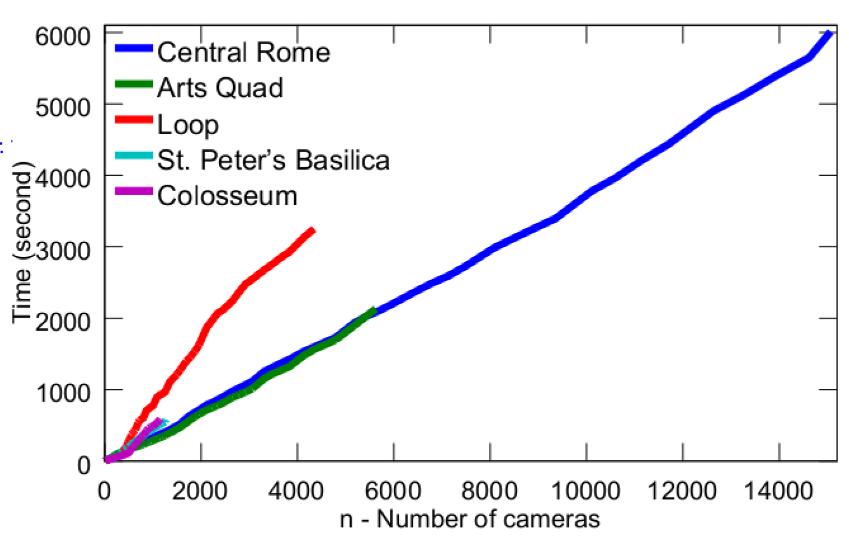
Towards Linear-time Incremental Structure from Motion

Changchang Wu

University of Washington



Reconstruction of Central Rome
(15065 cameras, 1.6M Points, 13M Observations)

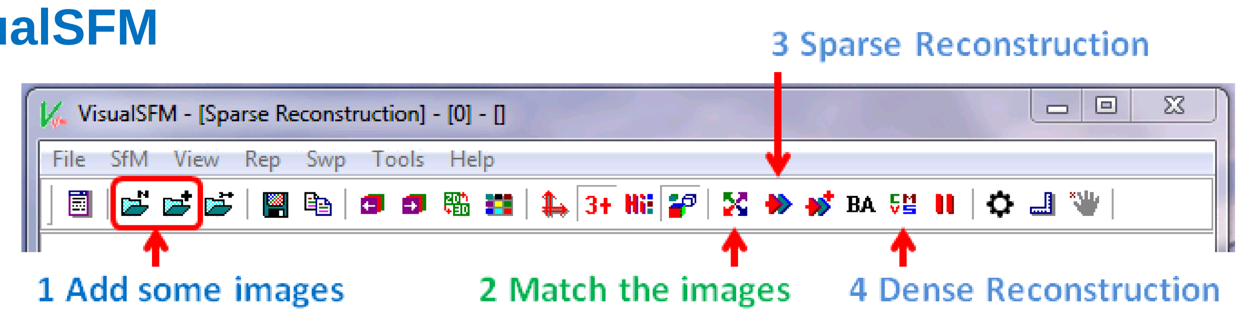


Time spent on incremental SfM stage
(not including feature matching)

Contributions

- We show that many sub-steps of incremental SfM, including BA and point filtering, require only O(n) time in practice when using a geometric BA strategy.
- Without sacrificing the time-complexity, we introduce a re-triangulation step to deal with the problem of accumulated drifts without explicit loop closing.
- A simple preemptive feature matching for reducing image matching cost.

VisualSFM

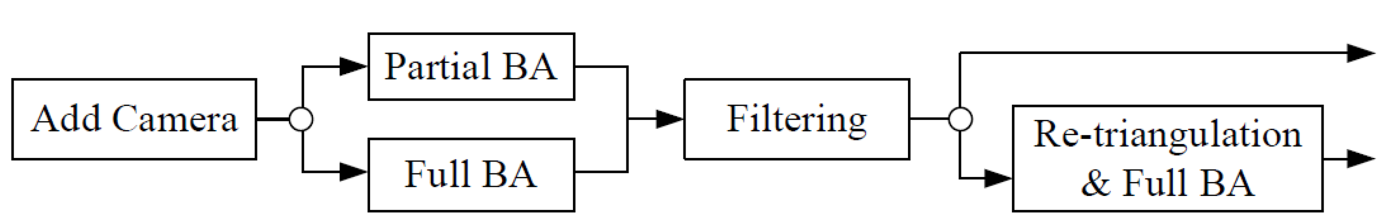


The proposed algorithms in this paper are available as part of VisualSFM: http://homes.cs.washington.edu/~ccwu/vsfm/ or http://ccwu.me/vsfm

Related "Truncations"

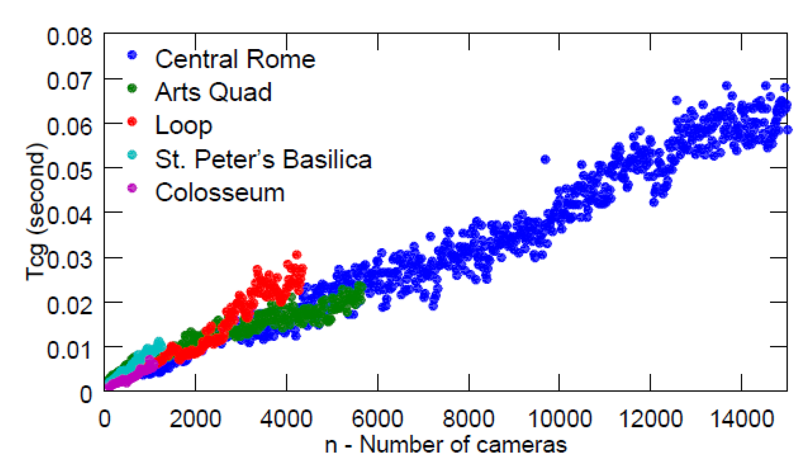
- Image matching using Vocabulary Tree and ANN, and filtering by GPS,
- Scene graph simplification (Skeletal graph or Iconic images),
- Bundle adjustment using Pre-conditioned Conjugated Gradient (PCG)

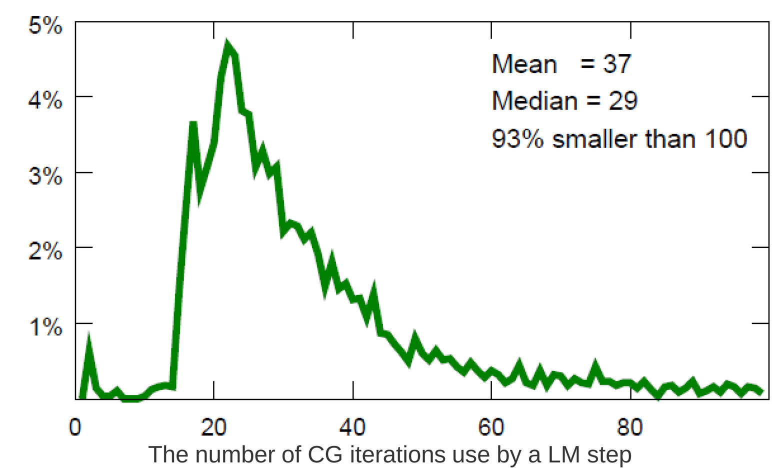
Incremental Structure from Motion



How Fast is Bundle Adjustment?

- PCG spent most of time on matrix-vector multiplication, where
  - The time complexity of a single CG iteration is *O(n)*;
  - The number of CG iterations depend on the condition number of the problems.





- BA can be done in linear time with truncation (max CG/LM iterations)

Bundle Adjustment Strategy

- There is no need to bundle adjust the entire model for ever image.

  With a linear sequence: full BA when *n* increases by α

$$\sum_i^{\lfloor n/\alpha \rfloor} T_{BA}(i*\alpha) = O\Big(\sum_i^{\lfloor n/\alpha \rfloor}(i*\alpha)\Big) = O\Big(\frac{n^2}{\alpha}\Big)$$

  With a geometric sequence: full BA when *n* increases relatively by *r*

$$\sum_i^{\infty} T_{BA}\Big(\frac{n}{(1+r)^i}\Big) = O\Big(\sum_i^{\infty}\frac{n}{(1+r)^i}\Big) = O\Big(\frac{n}{r}\Big)$$

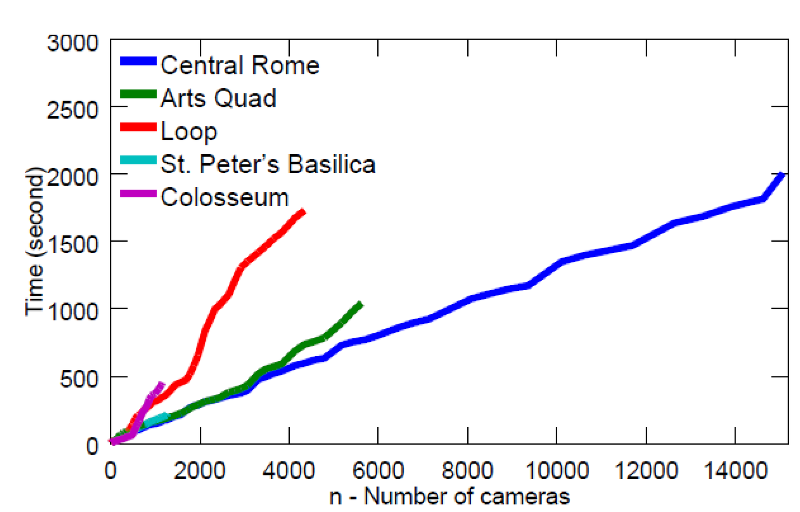
- Perform a partial BA when not performing full BA, which adds to *O(n).*
- Point filtering also requires only *O(n)* thanks to the geometric sequence.

Re-Triangulation (RT)

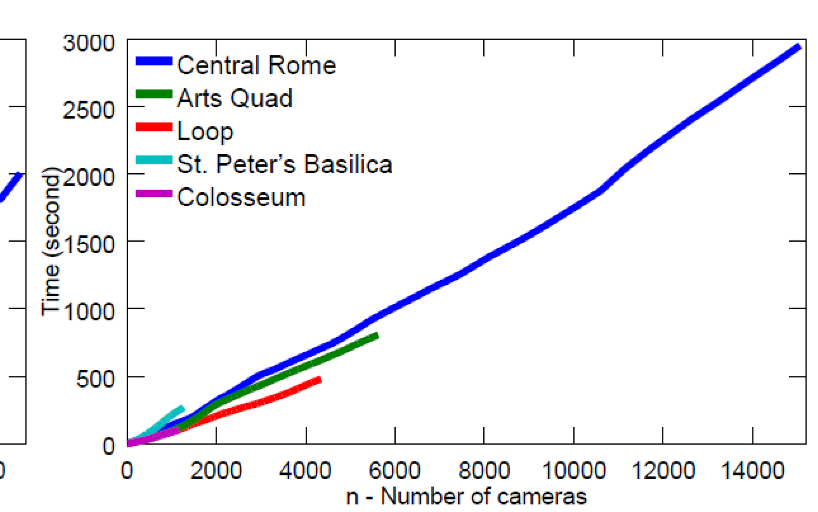
- Revisits feature matches in a geometric sequence, which takes *O(n)* time.
- Reduces accumulated drifts and allows for implicit loop-closing.

Time Complexity

- Still *O(n^2)* due to linear scan for the partial BA and filtering.
- Large constant factor for the *O(n)* portion for 15K cameras.



(b) Accumulated time of full BA.

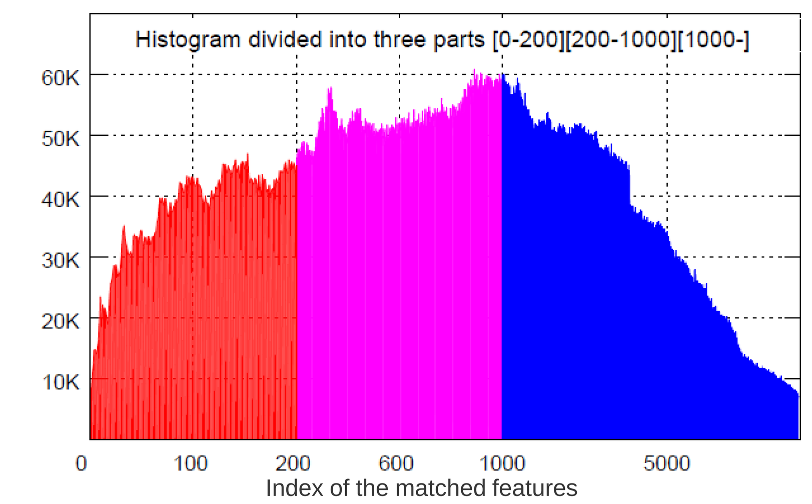


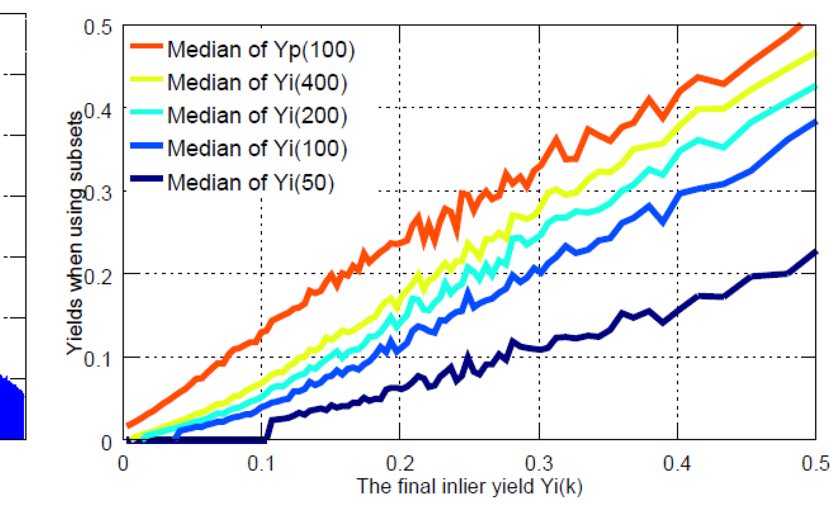
(c) Accumulated time of partial BA.

Preemptive Feature Matching

1. Sort the features in decreasing-scale order for each image.
2. Find the image pairs that need to be matched (VT, GPS, etc...).
3. Match the first *h* features for each pair. Let m_p *(h)* be the number of matches.
4. Standard feature matching for pairs that satisfy $m_p(h) \geq t_h$.

How reliable is the feature matching of just top h?





Significant reduction of the matching cost

Dataset	Without Preemptive Matching: Pairs to Match	Pairs With 15+ Inliers	Feature Matches	n	Using Preemptive Matching ($h = 100$): t_h	Pairs to Match	Pairs With 15+ Inliers	Feature Matches	n
Central Rome	N/A	N/A	N/A	N/A	4	13551K	540K	67M	15065
Arts Quad	15402K	192K	32M	5624	4	521K, 3%	62K, 32%	25M, 78%	4272
					2	4308K, 28%	121K, 63%	29M, 91%	5393
Loop	709K	329K	158M	4342	4	269K, 38%	235K, 71%	150M, 95%	4342
					8	151K, 21%	150K, 46%	135M, 85%	4342

Reconstruction Summary

Dataset	Input Images	Cameras n	Points p	Observations q	Time t Overall	Time Full BA	Time Partial BA	Time Adding	Time Filtering
Central Rome	32768	15065	1660415	12903348	6010	2008	2957	549	247
Arts Quad	6514	5624	819292	5838784	2132	1042	807	122	57
Loop	4342	4342	1101515	7195960	3251	1731	478	523	47
St. Peter's	1275	1267	292379	2706250	583	223	268	48	20
Colosseum	1164	1157	293724	1759136	591	453	100	19	9

Evaluation and Comparisons

Dataset	Full BA	Partial BA	RT	n	q	t	t/n	t/q
Central Rome	$r = 5\%$	Every Image	$r' = 25\%$	15065	12903K	1.67hour	0.40s	0.47ms
	$r = 25\%$	Every Image	$r' = 50\%$	15113	12958K	1.32hour	0.31s	0.37ms
	$r = 5\%$	Every 3 Images	$r' = 25\%$	14998	12599K	1.03hour	0.25s	0.29ms
	DISCO result of [4]			14754	21544K	13.2hour	3.2s	2.2ms
	Bundler result of [4]			13455	5411K	82hour	22s	54ms
Arts Quad	$r = 5\%$	Every Image	$r' = 25\%$	5624	5839K	0.59hour	0.38s	0.37ms
	$r = 25\%$	Every Image	$r' = 50\%$	5598	5850K	0.42hour	0.27s	0.26ms
	$r = 5\%$	Every 3 Images	$r' = 25\%$	5461	5530K	0.53hour	0.35s	0.35ms
	DISCO result of [4]			5233	9387K	7.7hour	5.2s	2.9ms
	Bundler result of [4]			5028	10521K	62hour	44s	21ms
Loop	$r = 5\%$	Every Image	$r' = 25\%$	4342	7196K	3251s	0.75s	0.45ms
	$r = 25\%$	Every Image	$r' = 50\%$	4342	7574K	1985s	0.46s	0.26ms
	$r = 5\%$	Every 3 Images	$r' = 25\%$	4341	7696K	3207s	0.74s	0.41ms

[4] D. Crandall, A. Owens, N. Snavely, and D. P. Huttenlocher. Discrete-continuous optimization for large-scale structure from motion. In CVPR, 2011.

Thanks to Noah Snavely for providing the Rome data and the statistics.