GraphMatch: Efficient Large-Scale Graph Construction for Structure from Motion

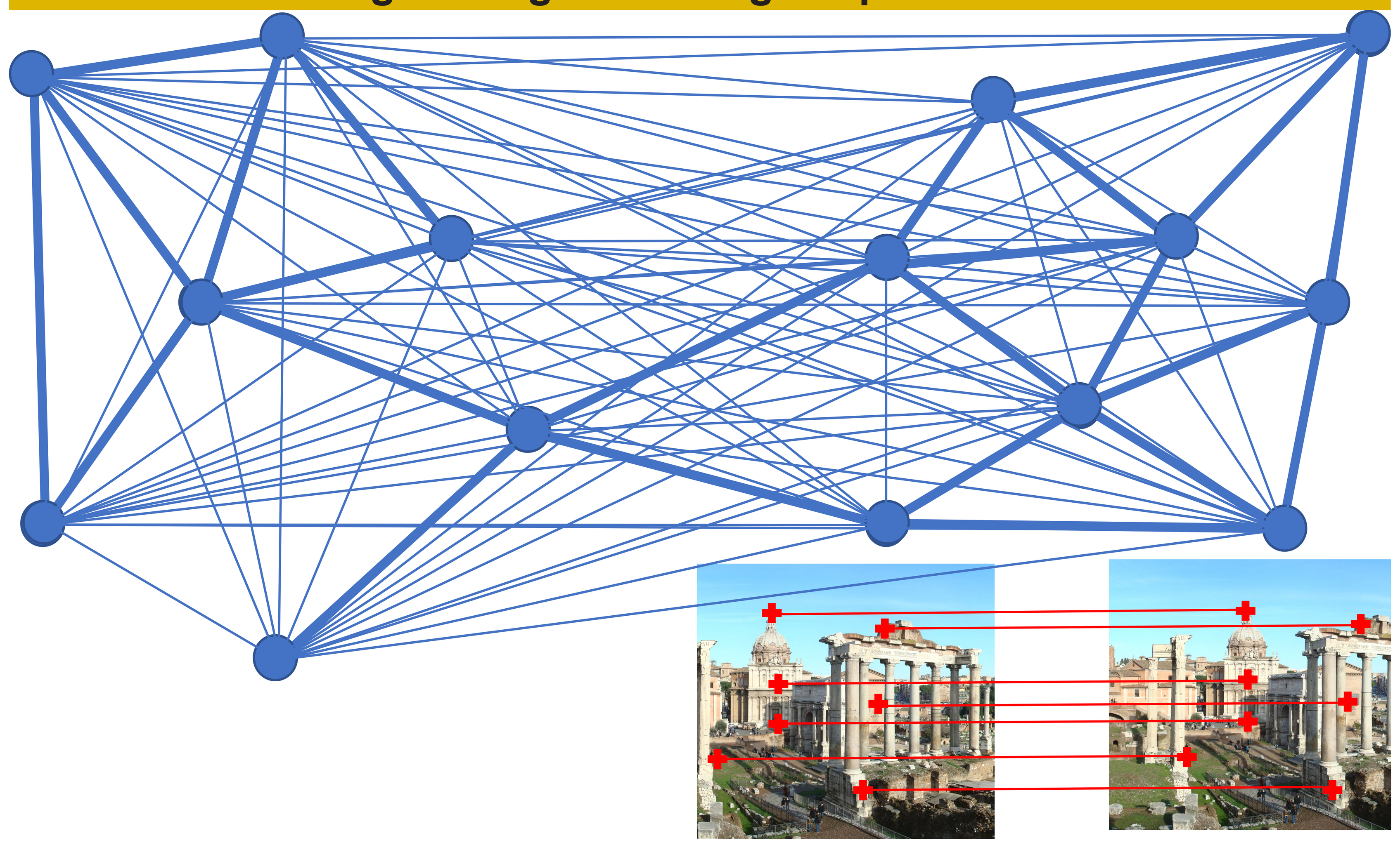
Qiaodong Cui[1], Victor Fragoso[2], Chris Sweeney[3] and Pradeep Sen[1]

[1]University of California Santa Barbara [2]West Virginia University [3]University of Washington

{qiaodong@umail, psen@ece}.ucsb.edu victor.fragoso@mail.wvu.edu csweeney@cs.washington.edu

Introduction

Fig. 1 Image Matching Graph for SfM



- Modern SfM pipelines require the building of a matching graph.
- Vast amount (75-95%) of image pairs do not match in large collections.
- Building and creating an image index from Voc. Trees are expensive. To combat the problem of creating matching graph for large datasets, we present a new matching algorithm called GraphMatch.

Contributions

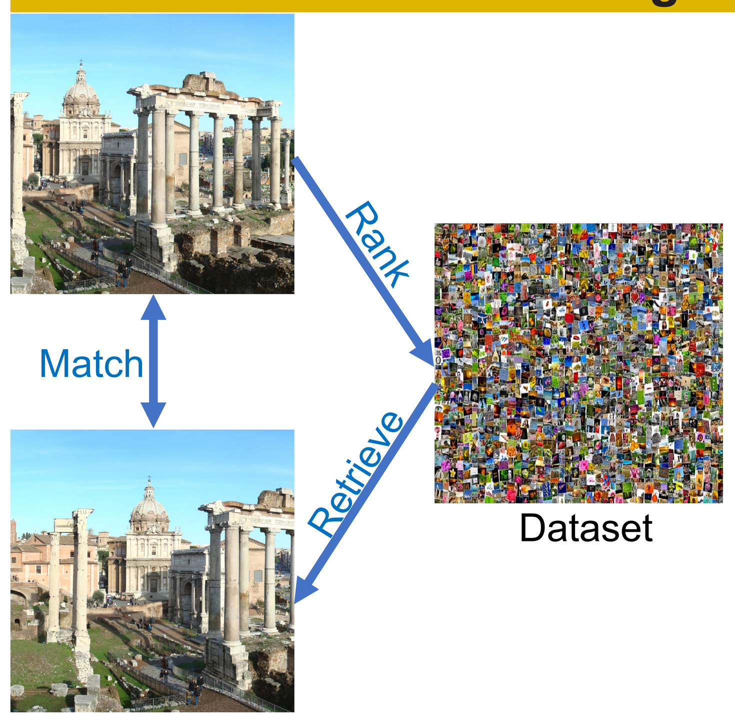
- We use priors computed from the distance between Fisher vectors, which is faster to compute and more indicative of a possible match than Voc. Tree similarity scores.
- We extend the "Propagation" step of PatchMatch algorithm to image matching, which maximizes the fraction of matching pairs.
- We use an iterative "sample-and-propagate" scheme which recover 80-90% of the total good matches in ground truth matching graph.

Graph Match Algorithm

1. Extract sift features and fisher vector from images.
2. Compute fisher distance pairs of image, obtain a ranked list for all images.
3. While the algorithm is not converged do
   Sampling step based on the fisher priors
   Propagation step to match between neighbors
4. Runs the reconstruction algorithm on resulting matching graph

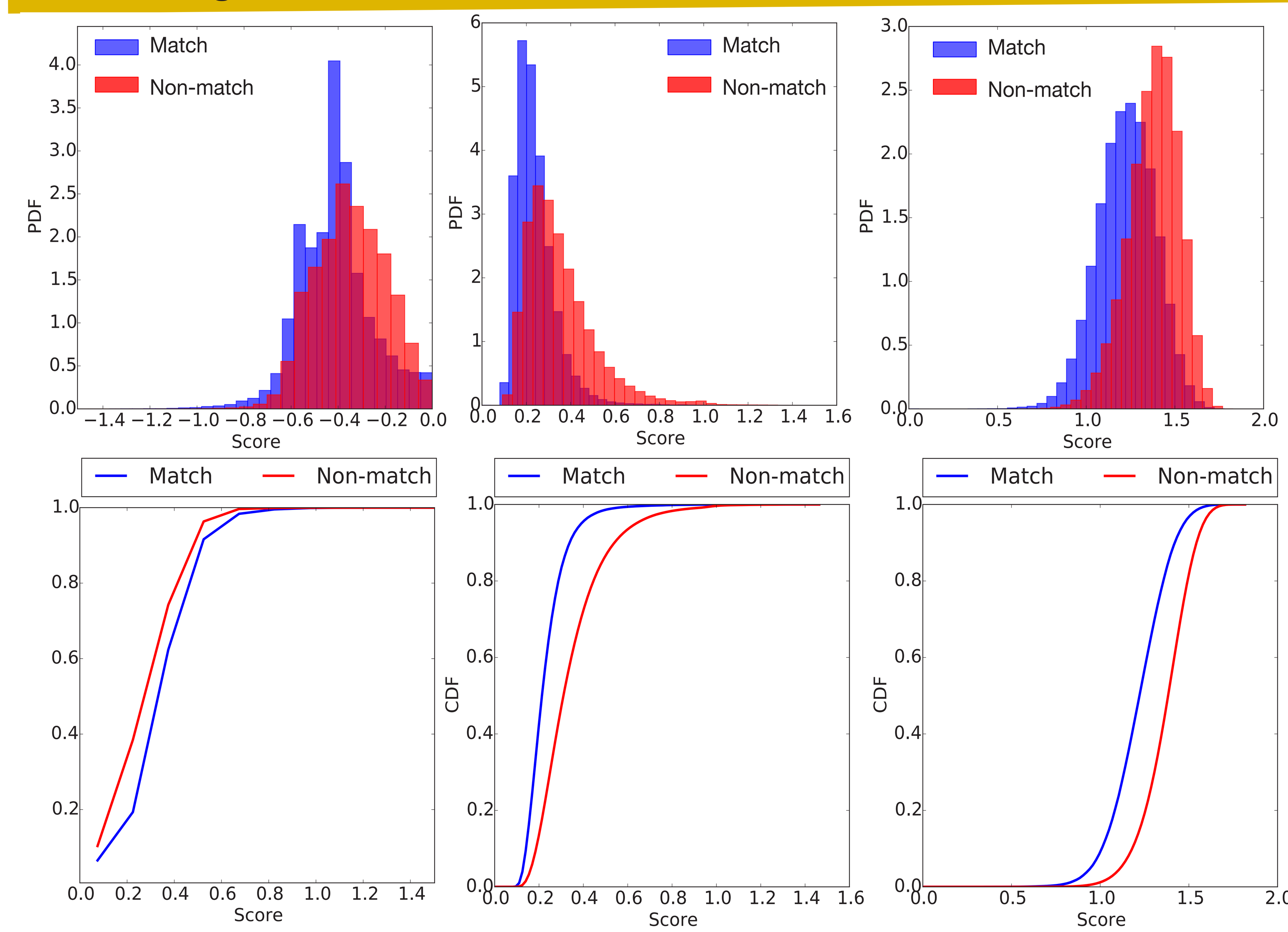
Sampling and Propagation

Fig. 2 Sampling Step



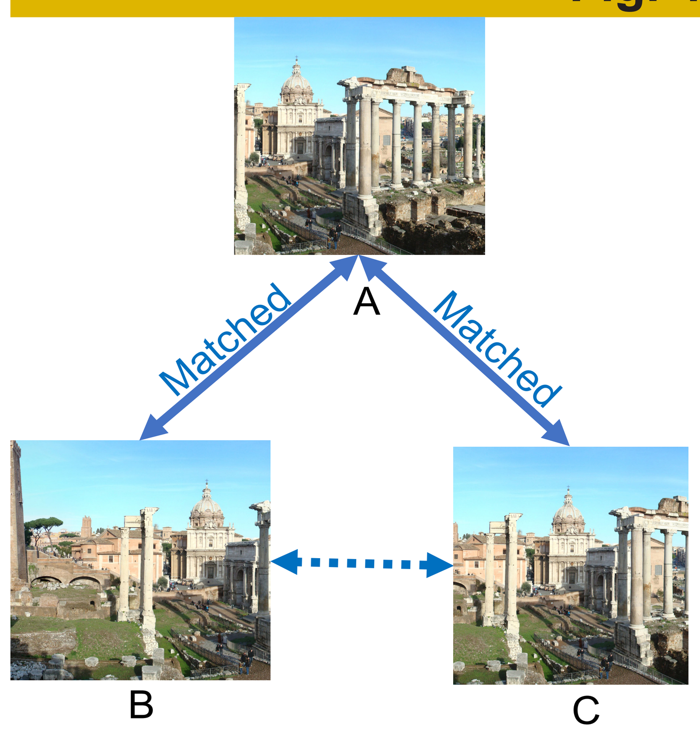
Sampling Step attempts to find connections between new regions. First, Scores are computed from appearance based image descriptors between all image pairs. Next highly ranked images are retrieved and then match against an image. We compared vocabulary tree, VLAD and Fisher vectors and found Fisher vectors are best at predicting matching pairs.

Fig. 3 PDFs for Voc. Trees, VLAD and Fisher vectors



(a) Voc. Trees Scores PDF and CDF (b) VLAD PDF and CDF (b) Fisher PDF and CDF

Fig. 4 Propagation Step



Propagation Step identifies new edges by leveraging the spatial coherence. For two neighbors B and C of image A, we concludes that B and C are also likely to match since images that are geometrically verified tend to be spatially correlated with each other. We use the propagation step to densify our matching graph.

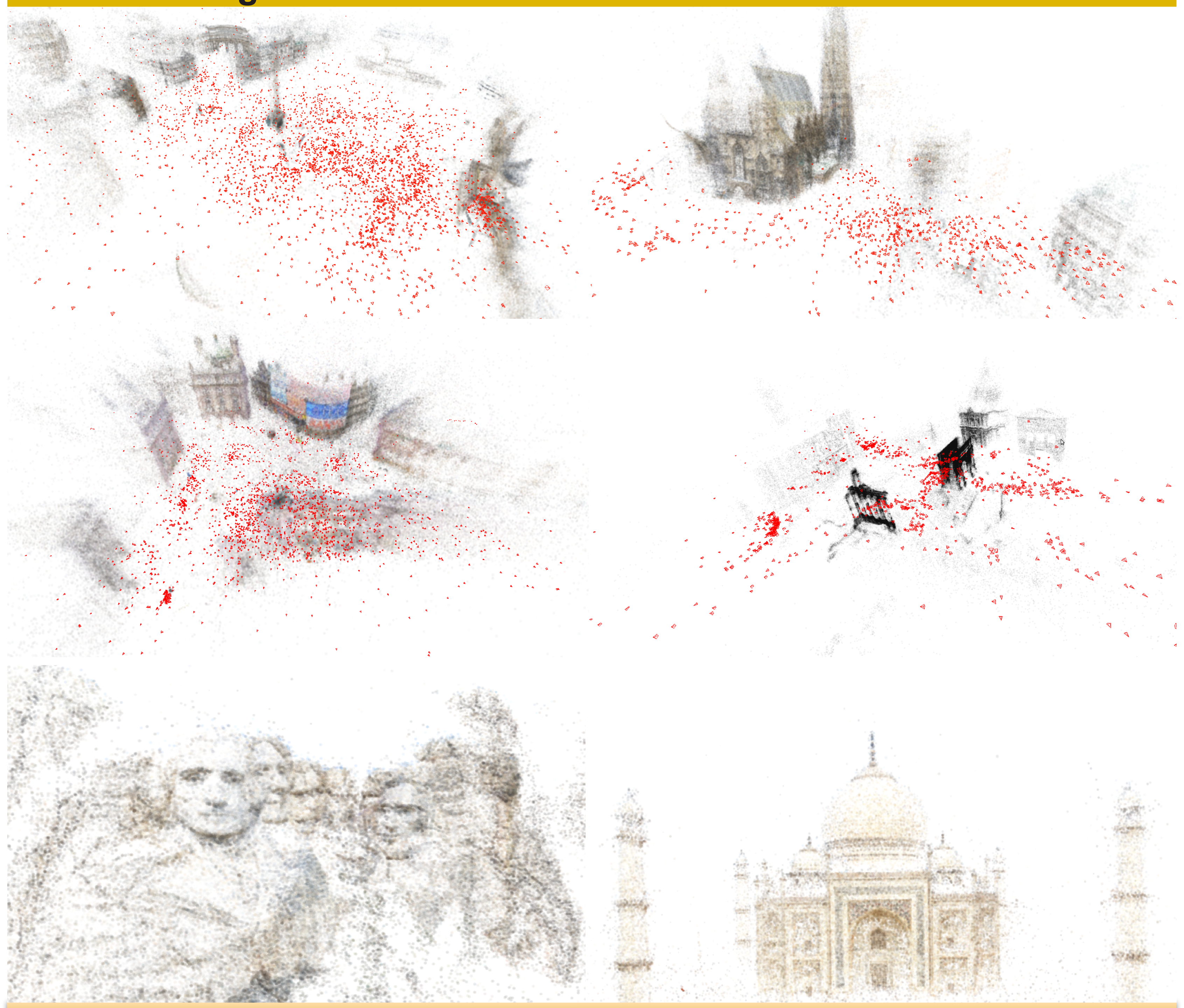
Experiment Results

Table 1. Results of timing and camera reconstructed for each method

Dataset	# images / graph density	Alg.	# recon cameras	# edges	Time (min) pre	match	recon	total
Taj Mahal	1497	baseline	576	63,474	0	696.62	38.81	754.62
		voc. tree	413	9,368	10.04	37.39	53.60	109.23
	0.0437	BRIAD	319	21,332	10.04	26.98	13.14	58.19
		ours	615	48,949	0.87	41.78	57.62	108.54
Montreal N.D.	2298	baseline	486	33,836	0	1556.23	51.30	1643.39
		voc. tree	431	18,060	18.35	56.66	37.16	123.04
	0.0100	BRIAD	430	8,241	18.35	32.51	52.50	113.82
		ours	460	31968	1.39	48.90	48.62	109.65
Roman Forum	2364	baseline	1247	52,155	0	1839.29	58.78	1939.45
		voc. tree	951	15,795	20.48	91.88	71.50	196.11
	0.0159	BRIAD	452	26,730	20.48	61.47	35.68	129.37
		ours	1142	50,216	1.39	78.80	52.81	145.04
NYC Library	2550	baseline	261	15,241	0	1065.78	24.98	1119.26
		voc. tree	232	7,639	20.18	56.94	18.37	107.18
	0.0039	BRIAD	135	4,327	20.18	39.44	9.02	80.03
		ours	245	13,427	1.40	41.46	24.53	78.82
Alamo	2915	baseline	726	62,793	0	2506.32	85.31	2646.70
		voc. tree	573	19,932	27.64	73.37	44.93	160.47
	0.0122	BRIAD	251	12,490	27.64	41.28	6.33	89.25
		ours	648	50,943	1.61	61.29	65.82	143.04
Vienna Cathedral	6288	voc. tree	273	10,578	117.36	450.80	20.05	624.96
		BRIAD	242	17,578	117.36	216.60	22.94	389.75
	0.004	ours	794	79,394	3.37	367.58	44.28	450.60

BRIAD: Building Rome in a day of Agarwal et al.

Figure 5. Reconstructions of several datasets



Acknowledgments

This work was supported in part by NSF grants IIS-1657179, IIS-1342931, and IIS-1321168. The authors would like to thank Atieh Taheri, who worked on an early version of the algorithm and conducted some preliminary experiments.