Computer Graphics & Multimedia
Prof. Dr. Leif Kobbelt
RWTH Aachen University

Improving Image-Based Localization by Active Correspondence Search

Torsten Sattler, Bastian Leibe, Leif Kobbelt

RWTH AACHEN UNIVERSITY

1 Motivation

- **Structure-from-Motion** point cloud: every point has ≥ 2 SIFT descriptors [4]
- **2D-to-3D correspondences** needed for **pose estimation** for query image

- State-of-the-art **registration performance** obtained using a kd-tree for correspondence search, but kd-tree search is slow (>3 *seconds*) [6]
- Approaches with state-of-the-art **registration efficiency** (*0.25-0.4 seconds*) can register **significantly less images** [1,2,3,6]
- **Our approach:** State-of-the-art **registration performance & efficiency** using **Active Correspondence Search** and **Visibility Filtering**

2 Datasets

- Same datasets used in [1,2,3,5]
- Query images have dimensions ≤ 1600 pixels

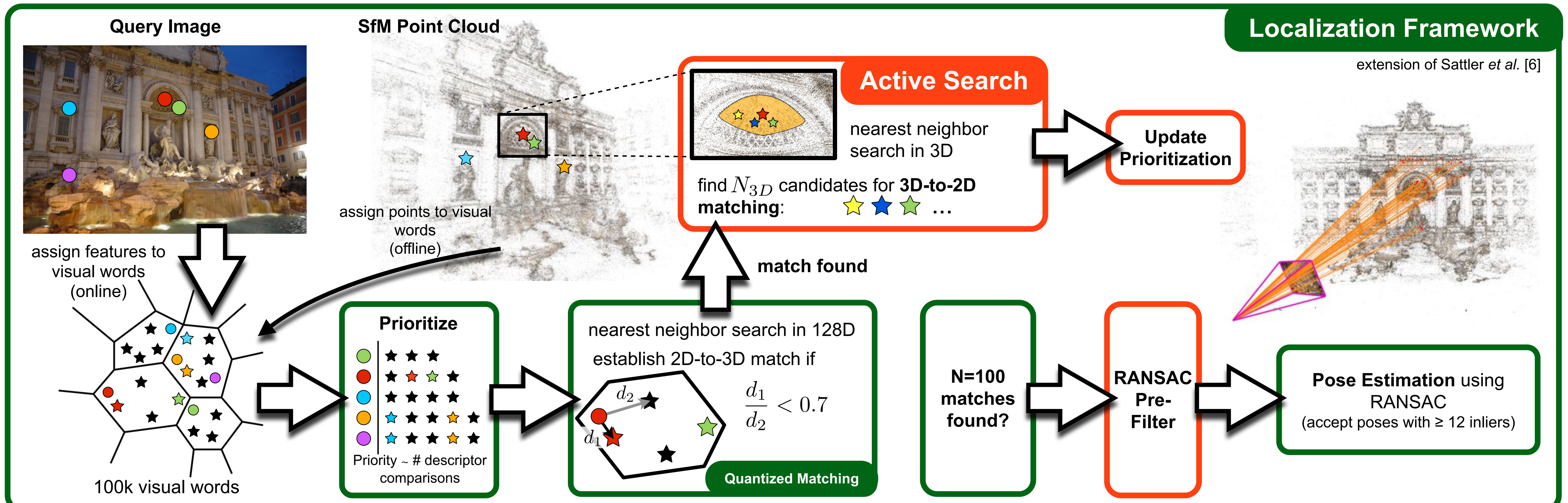
Dataset	# Cameras	# 3D Points	# Descriptors	# Query Images
Dubrovnik	6044	1.886.884	9.606.317	800
Rome	15.179	4.067.119	21.515.110	1000
Vienna	1324	1.123.028	4.854.056	266

3 2D-to-3D vs. 3D-to-2D matching

Complimentary Advantages	Method	Efficiency	Reject wrong matches	Accept correct matches
	3D-to-2D	✓	✗	✓
	2D-to-3D	✗	✓	✗

Localization Framework

extension of Sattler *et al.* [6]



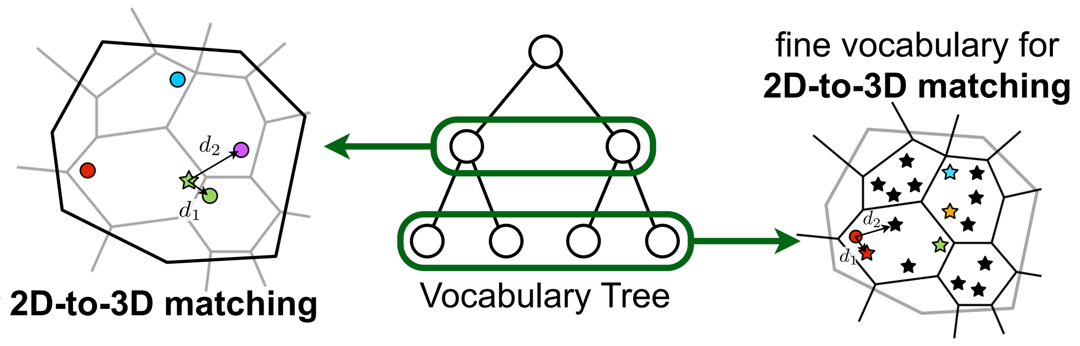
Source code available at http://www.graphics.rwth-aachen.de/localization

4 Active Correspondence Search

1. Find N_{3D} candidates for **3D-to-2D matching** through nearest neighbor search in 3D
2. Insert into **prioritization**

Use **2D-to-3D matches** to **seed 3D-to-2D matching**

Re-use structures for 3D-to-2D Matching



Computational Complexity

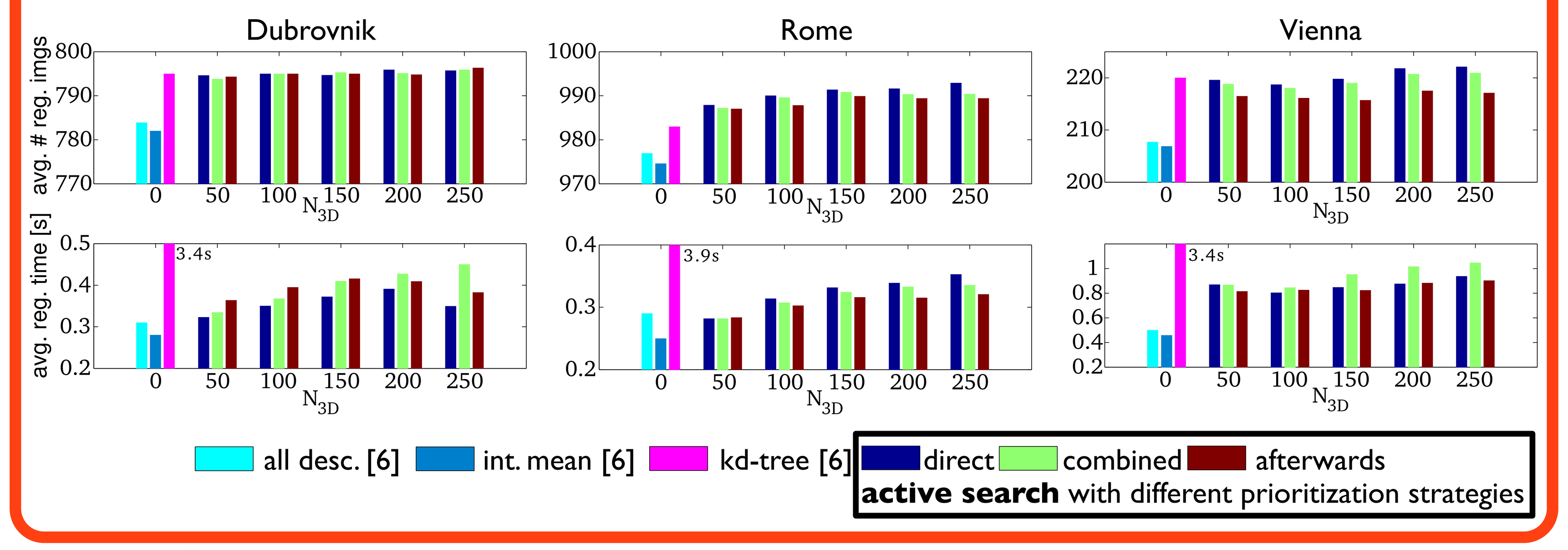
Soft assignments [5] $\mathcal{O}\left(c \cdot F/W \cdot P\right)$

active search $\mathcal{O}\left(N \cdot F/W' \cdot N_{3D} \cdot \log_2(P)\right)$ ⇒ **Better scalability**

constant

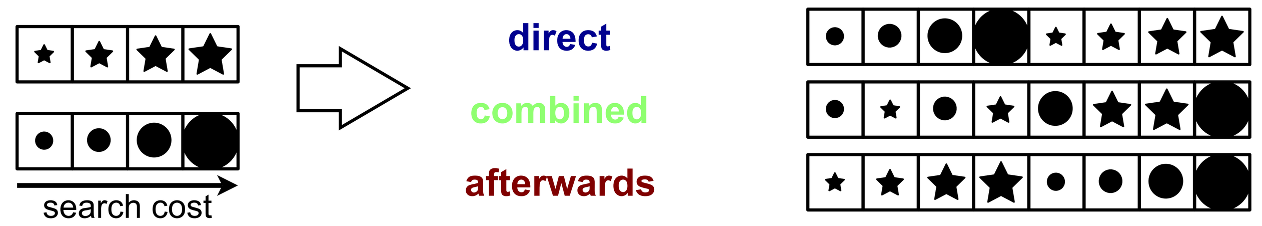
F = # image features, W = size fine voc., W' = size coarse voc., c+1 = # words a query feature is assigned to, **P = # 3D points**
N = # matches needed to terminate search, N_{3D} = # nearest neighbors in 3D

Active Search improves registration performance



5 Prioritization

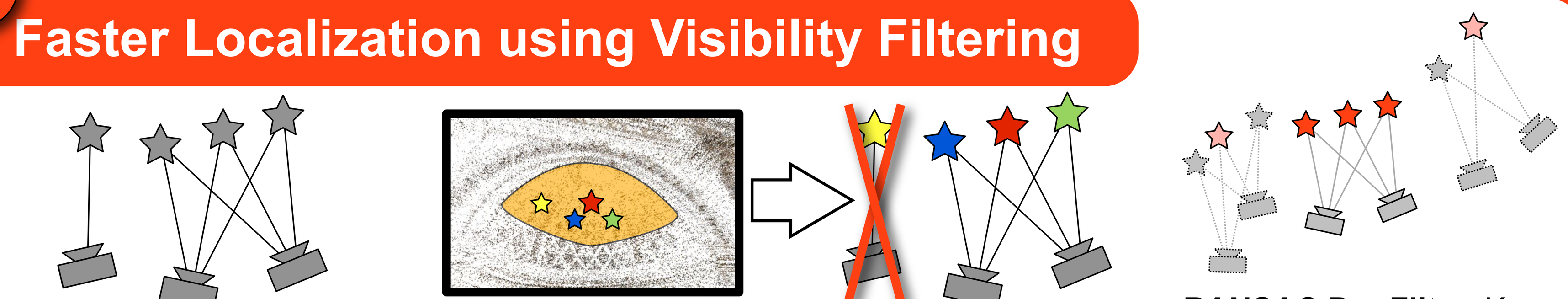
- Prioritize over **estimated search cost** ~ # descriptors inside a word [5]
- **Independent** of search direction!



Prioritization Scheme	Preferred Direction	Registration performance	Localization accuracy
direct	3D-to-2D	best	worst
combined	**cheaper direction**	**medium**	**good**
afterwards	2D-to-3D	worst	good

Use the *combined* strategy, prioritize cheaper direction

6 Faster Localization using Visibility Filtering

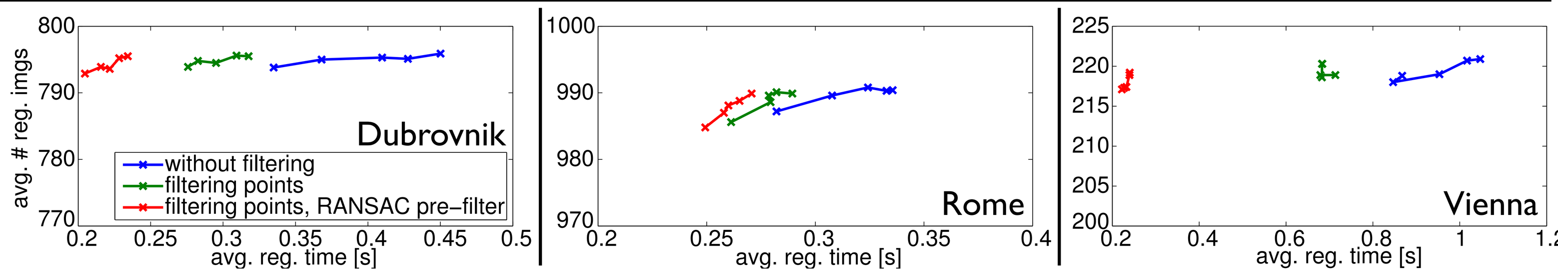


bipartite point-camera graph models co-visibility

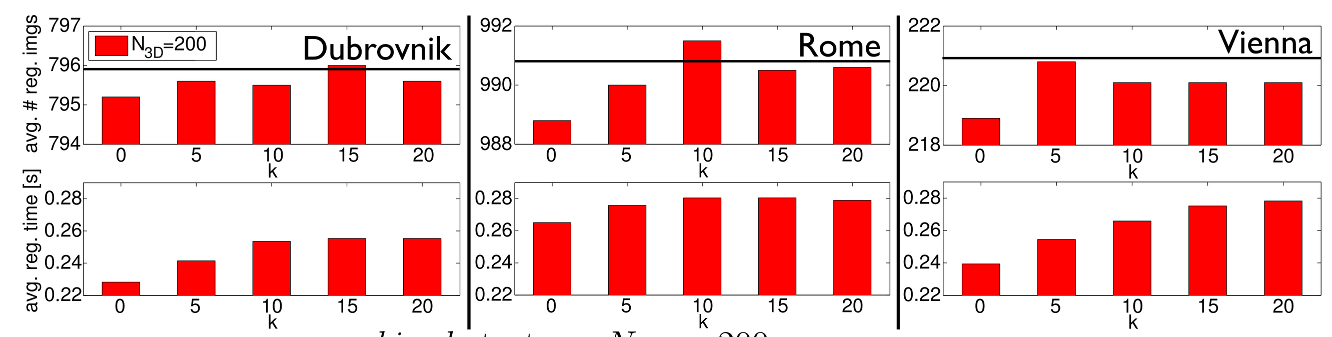
Filtering Points: Remove 3D-to-2D matching candidates not co-visible with 2D-to-3D match

RANSAC Pre-Filter: Keep only largest subgraph defined by matches

Active search + filtering
combined strategy



Cluster k nearest cameras to recover performance



combined strategy, $N_{3D} = 200$

7 Comparison with state-of-the-art

- **active search**: *combined* prioritization, both filters, $N_{3D}=200, k=10$ (CPU)
- all desc., int. mean: 3D-to-2D matching by Sattler *et al.* [6] without **active search** (CPU)
- P2F: Prioritized 3D-to-2D matching proposed by Li *et al.* [3] (CPU)
- P2F+F2P: Additional 2D-to-3D matching if P2F fails [3] (CPU)
- Vis. Prob.: 3D-to-2D matching based on visibility probabilities [1] **(ECCV'12)** (CPU)
- Image retrieval methods using a Vocabulary tree:
  - Irschara *et al.* [2] *(GPU)*
  - Li *et al.* [3] (CPU)

Method	Dubrovnik			Rome			Vienna		
	# reg. images	registr. time [s]	reject. time [s]	# reg. images	registr. time [s]	reject. time [s]	# reg. images	registr. time [s]	reject. time [s]
active search	**795.5**	**0.25**	0.56	**991.5**	0.28	2.14	**220.1**	**0.27**	0.52
all desc. [6]	783.9	0.31	2.22	976.9	0.29	1.90	207.7	0.50	2.40
int. mean [6]	782.0	0.28	1.70	974.6	**0.25**	1.66	206.9	0.46	2.43
Vis. Prob. [1]	*788*	***0.25***	***0.51***	*977*	*0.27*	***0.61***	*219*	*0.4*	***0.49***
P2F [3]	753	0.73	2.70	921	0.91	2.93	204	0.55	1.96
P2F+F2P [3]	753	0.70	3.96	924	0.87	4.67	205	0.54	3.62
Voc. tree (all) [3]	668	1.4	4.0	828	1.2	4.0	-	-	-
Voc. tree (points) [3]	677	1.3	4.0	815	1.2	4.0	-	-	-
Voc. tree GPU [2]	-	-	-	-	-	-	165	≤ **0.27** (worst case)	
kd-tree [4]	**795**	3.4	14.45	983	3.97	6.27	**220**	3.44	2.72

- Ground truth: Geo-registered version of Dubrovnik model
- Pose estimation using 6-point DLT algorithm

Method	# reg. images	Median [m]	Quartiles [m]		#images with error	
			1st	3rd	< 18.3m	> 400 m
active search	**795.5**	1.4	**0.4**	5.3	**704** (88.5%)	9 (1.1%)
all desc. [6]	783.9	1.4	**0.4**	5.9	685 (87.4%)	16 (2.0%)
int. mean [6]	782.0	**1.3**	0.5	**5.1**	675 (86.3%)	13 (1.7%)
Vis. Prob. [1]	*788*	*3.1*	*0.88*	*11.83*		
P2F [3]	753	9.3	7.5	13.4	655 (87%)	**0**

Related Work

[1] S. Choudhary and P. J. Narayanan. Visibility Probability Structure from SfM Datasets and Applications. ECCV'12
[2] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof. From structure-from-motion point clouds to fast location recognition. CVPR'09.
[3] Y. Li, N.Snavely, and D.P. Huttenlocher. Location recognition using prioritized feature matching. ECCV'10.
[4] D. Lowe. Distinctive image features from scale-invariant keypoints. IJCV, 60(2):91-110, 2004.
[5] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. CVPR'08.
[6] T. Sattler, B. Leibe, and L. Kobbelt. Fast Image-Based Localization using Direct 2D-to-3D Matching. ICCV'11