华中科技大学自动化学院，图像识别与人工智能研究所，
多谱信息处理国家重点实验室，
图像信息处理与智能控制教育部重点实验室

Efficient Large Scale 3D Reconstruction

陶文兵 (Wenbing Tao)

School of Automation,
Institute for Pattern Recognition and Artificial Intelligence
National Key Laboratory of Science and Technology on Multi-spectral Information Processing,
Key Laboratory of Ministry of Education for Image Processing and Intelligence Control,
Huazhong University of Science & Technology,

主要合作者：Qingshan Xu(徐青山)，Kun Sun(孙琨)，Tao Xu(徐涛)

目录

01 Background

02 GPU accelerated large scale image matching

03 Large scale Structure from Motion

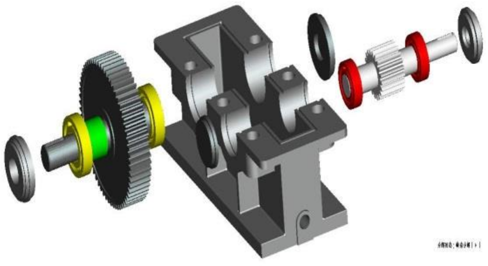
04 Multi-view stereo for 3D dense reconstruction

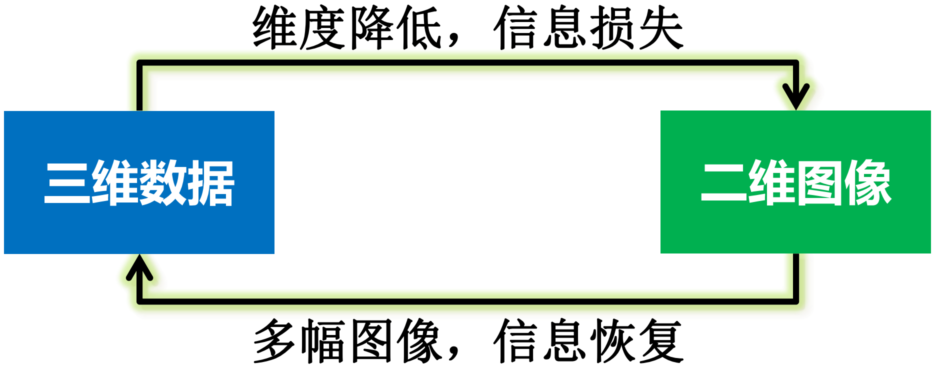
PART **1**

Background

Background

1 The three-dimensional model can provide the most true perception of the world





Background

2 The three-dimensional city model has extensive application

市政规划

灾后救援

虚拟景观

数字校园

三维导航

公共安全

交通管理

地图查询

Existing 3D modeling method

1. 利用几何造型技术建模

优 点

技术成熟，有很多流行的商业软件

Autodesk Maya

AutoCAD

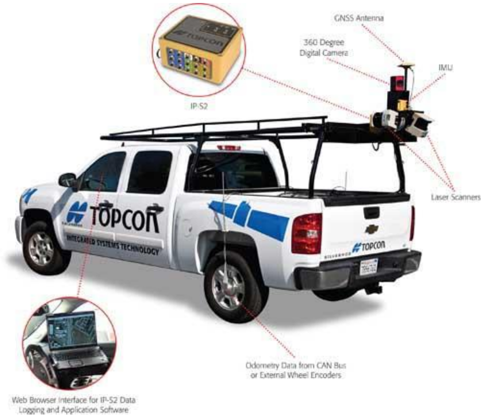
3ds max

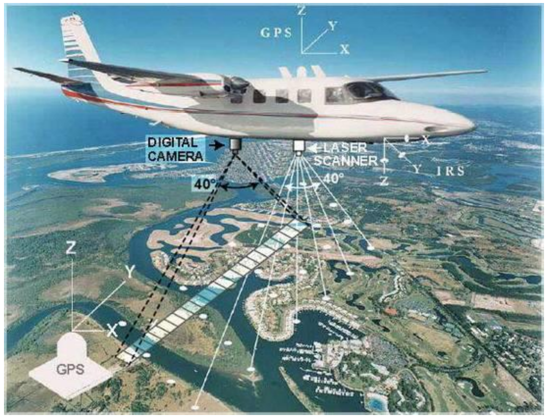
缺 点

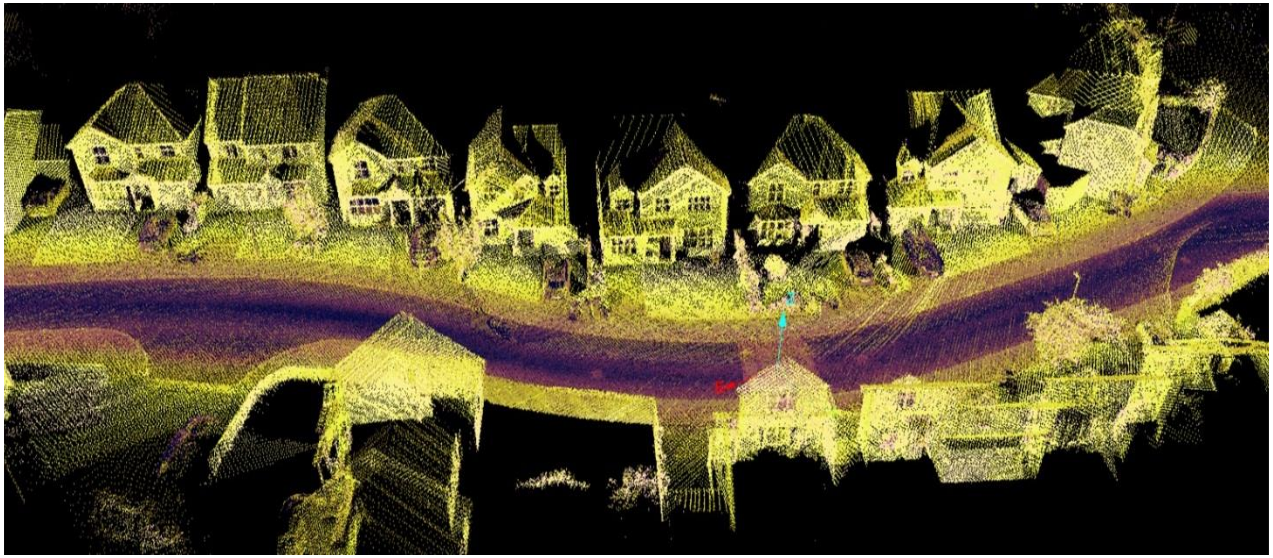
- ◆ 重建精度差，不能反映真实尺寸
- ◆ 重建真实感差，技术过于虚拟化

Existing 3D modeling method

2.主动接触式三维建模(激光雷达扫描仪、结构光扫描仪、红外测距仪)


GNSS Antenna
360 Degree Digital Camera
IMU
IP-S2
TOPCON
INTEGRATED SYSTEMS TECHNOLOGY
Laser Scanners
Odometry Data from CAN Bus or External Wheel Encoders
Web Browser Interface for IP-S2 Data Logging and Application Software





优 点

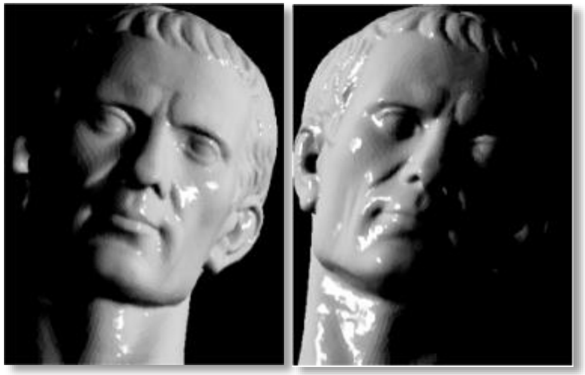
主动测量，直接得到三维点云信息，不需要复杂的后续计算和处理

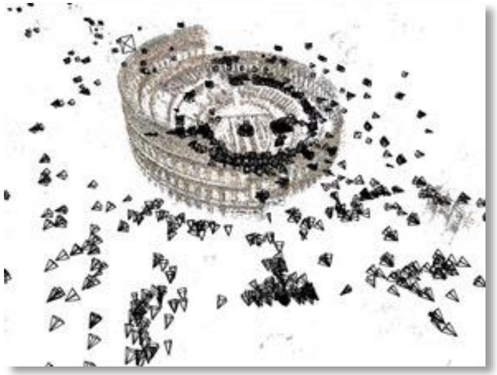
缺 点

- ◆ 设备操作复杂
- ◆ 重建成本很高
- ◆ 远距离精度差
- ◆ 重建真实感差

Existing 3D modeling method

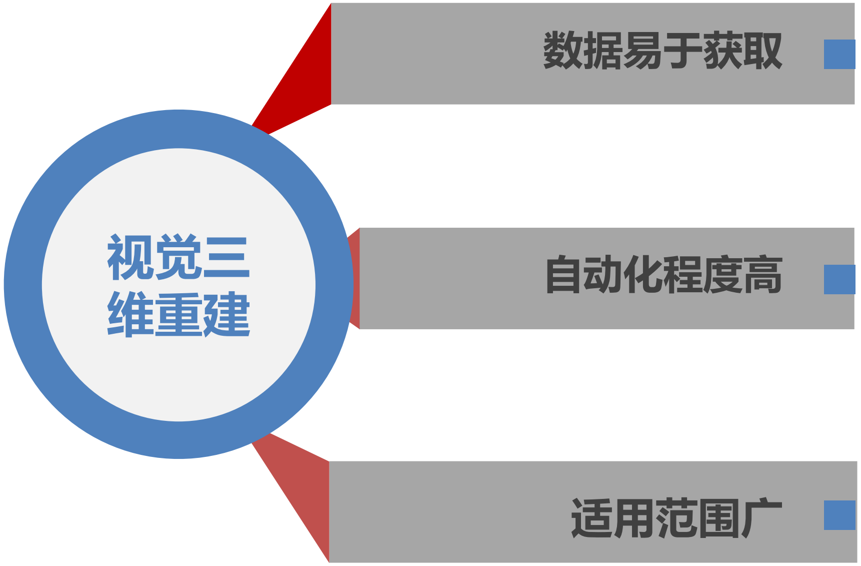
3. 被动式三维建模(视觉算法)





- Shape from X（阴影、纹理、遮挡等）
- 双目立体视觉（Binocular Stereo）
- 运动恢复结构（Structure from Motion，SfM）

Multiple-view 3D reconstruction

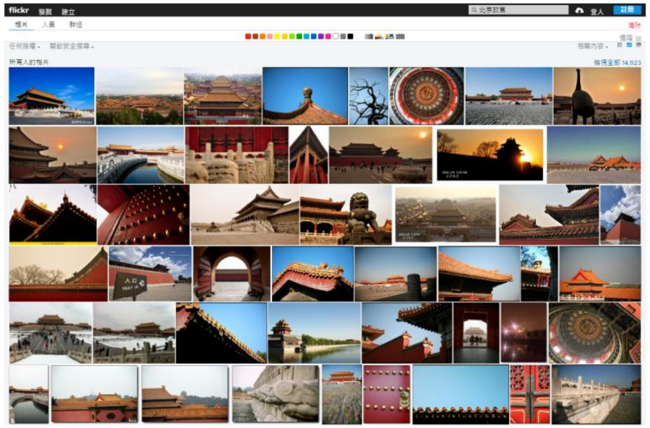


视觉三维重建

数据易于获取

自动化程度高

适用范围广



2014 年全球有大约 8800 亿张新的图片产生
2017 年这一数字达到 1.3 万亿

The basic procedure

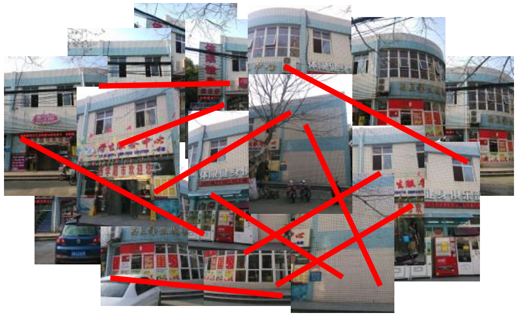
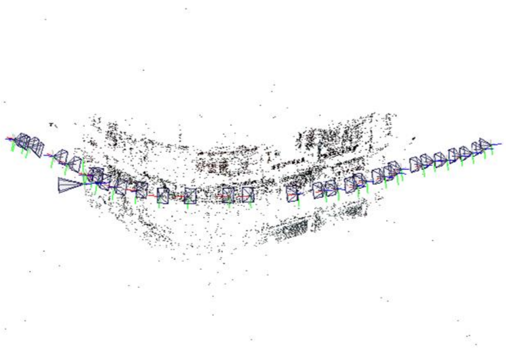
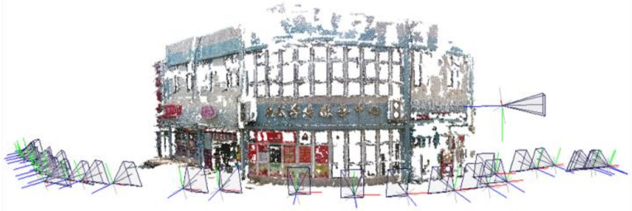


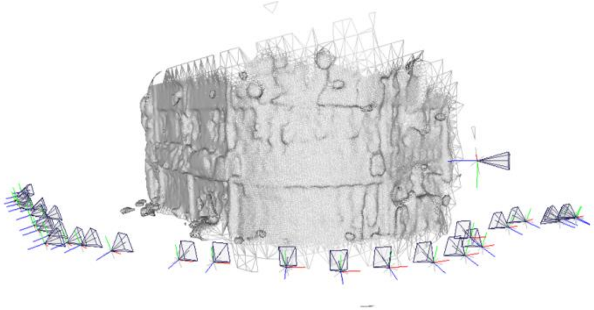
Image matching



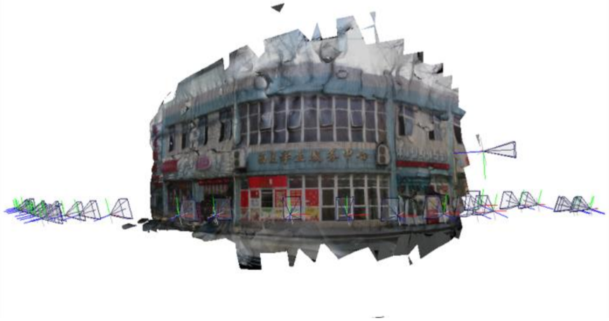
Structure from Motion



Dense representation



Surface reconstruction

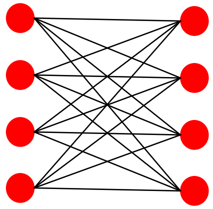


Texture mapping

PART 2

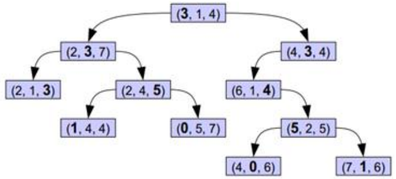
GPU Accelerated Cascade Hashing Image Matching

SIFT, Kd-Tree, CasHash and siftGPU



SIFT Matching (Lowe1999)：
Brute search
Find the smallest Euclidean distance and significant point

$O(N^2)$， a pair of images costs 4-5 seconds



Kd-Tree (Muja2009)：
Binary search tree
Approximate nearest neighbor (ANN) search

O(log N)， 2-4 pairs / s

10^4 SIFT points

Hashing lookup

Hashing remapping

<10

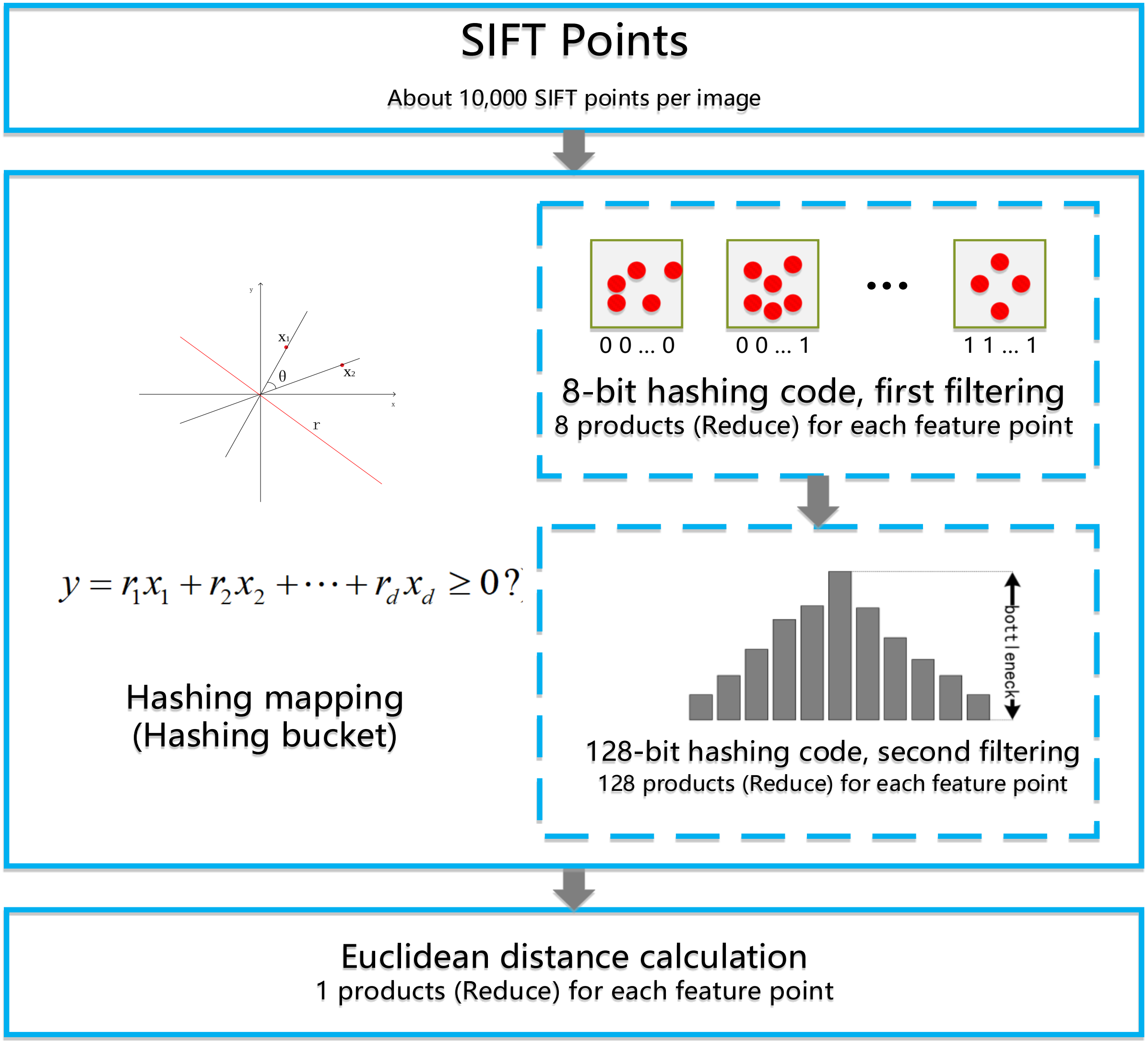
Cascade Hashing (Cheng2014)：
Two-level hashing filtering
ANN search

Lower algorithm complexity
10-20 pairs / s

siftGPU(Wu 2013)

40-50pair/s

Cascade Hashing



SIFT Points

About 10,000 SIFT points per image

$$y = r_1x_1 + r_2x_2 + \cdots + r_dx_d \geq 0?$$

Hashing mapping
(Hashing bucket)

8-bit hashing code, first filtering
8 products (Reduce) for each feature point

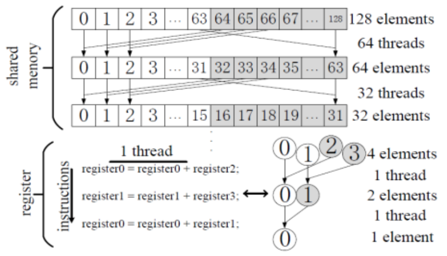
128-bit hashing code, second filtering
128 products (Reduce) for each feature point

Euclidean distance calculation
1 products (Reduce) for each feature point

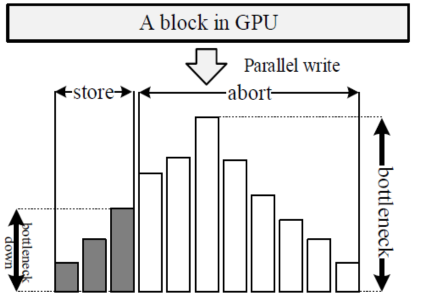
GPU Accelerated CasHash

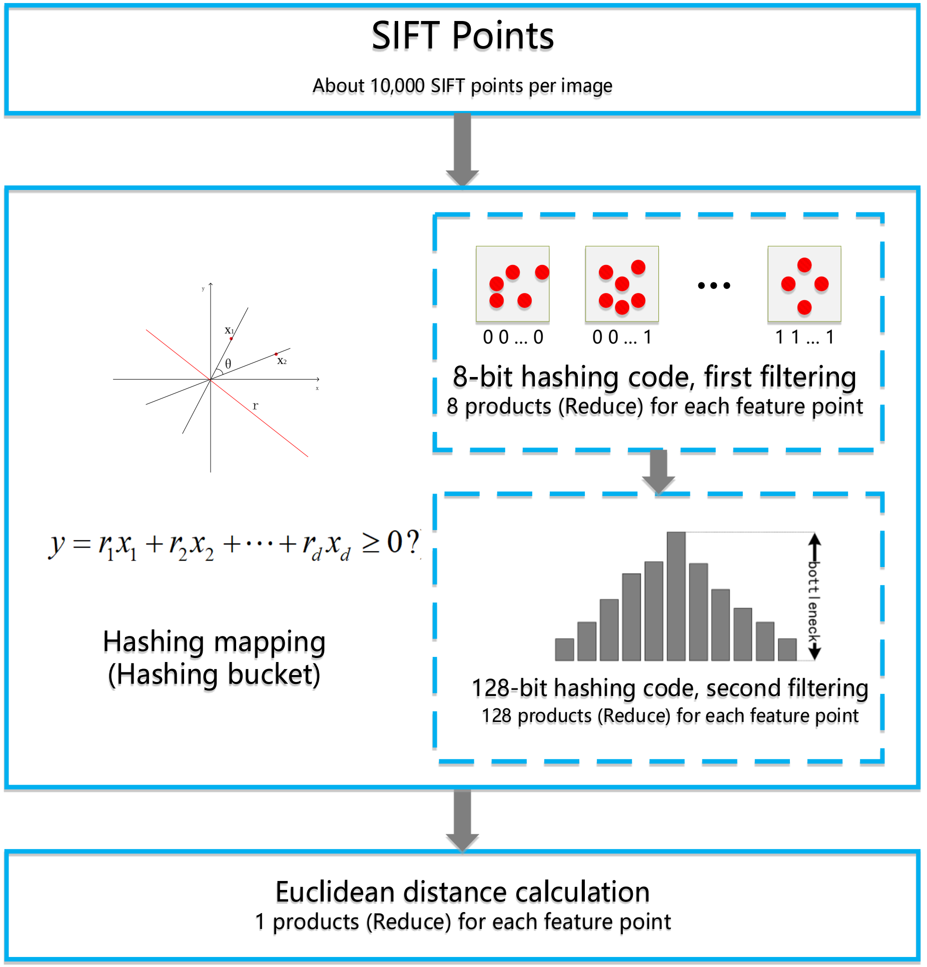
GPU algorithms

Fast Computation of Reduction



Improved Parallel Hashing Ranking





SIFT Points

About 10,000 SIFT points per image

$$y = r_1x_1 + r_2x_2 + \cdots + r_dx_d \geq 0?$$

Hashing mapping
(Hashing bucket)

8-bit hashing code, first filtering

8 products (Reduce) for each feature point

128-bit hashing code, second filtering

128 products (Reduce) for each feature point

Euclidean distance calculation

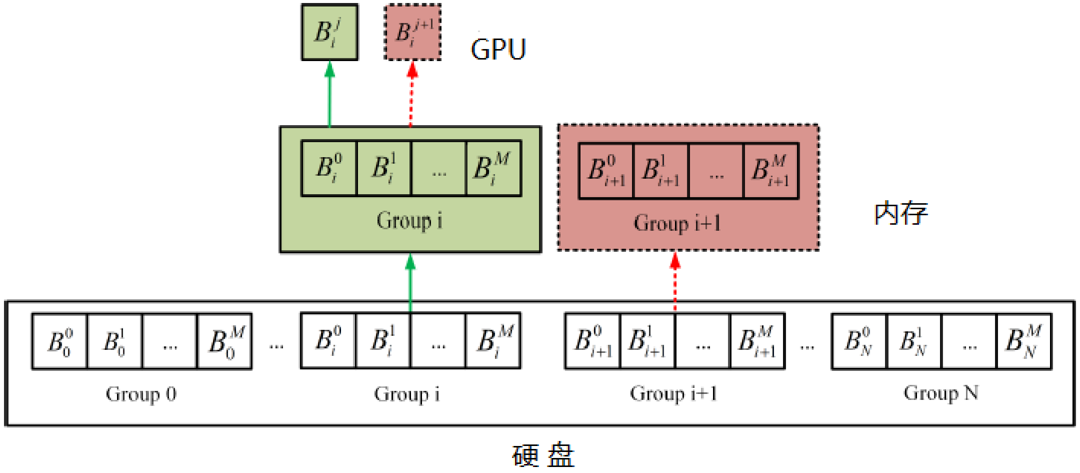
1 products (Reduce) for each feature point

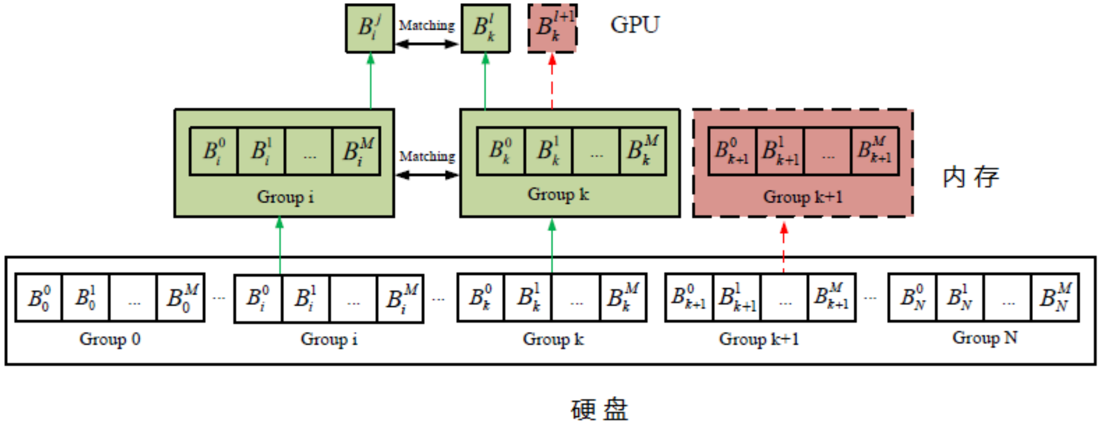
GPU-Memory-Disk
Data Exchange Strategy

Tao Xu, Kun Sun and Wenbing Tao*, GPU Accelerated Cascade Hashing Image Matching for Large Scale 3D reconstruction, arXiv:1805.08995

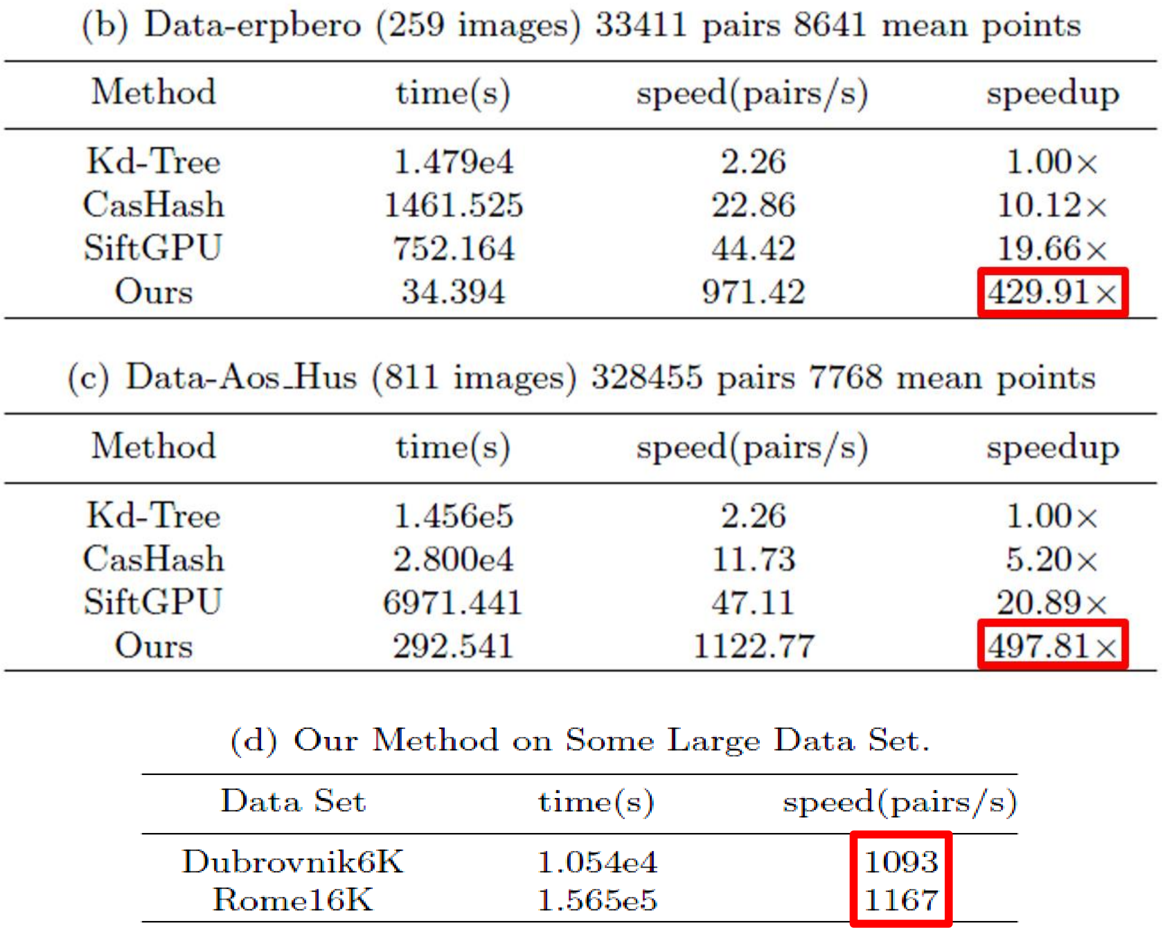
GPU Accelerated CasHash

Data Scheduling Strategy


B_i^j
B_i^{j+1}
GPU
B_i^0 B_i^1 ... B_i^M
Group i
B_{i+1}^0 B_{i+1}^1 ... B_{i+1}^M
Group i+1
内存
B_0^0 B_0^1 ... B_0^M
Group 0
...
B_i^0 B_i^1 ... B_i^M
Group i
B_{i+1}^0 B_{i+1}^1 ... B_{i+1}^M
Group i+1
...
B_N^0 B_N^1 ... B_N^M
Group N
硬 盘


B_i^j
Matching
B_k^l
B_k^{l+1}
GPU
B_i^0 B_i^1 ... B_i^M
Group i
Matching
B_k^0 B_k^1 ... B_k^M
Group k
B_{k+1}^0 B_{k+1}^1 ... B_{k+1}^M
Group k+1
内 存
B_0^0 B_0^1 ... B_0^M
Group 0
...
B_i^0 B_i^1 ... B_i^M
Group i
...
B_k^0 B_k^1 ... B_k^M
Group k
B_{k+1}^0 B_{k+1}^1 ... B_{k+1}^M
Group k+1
...
B_N^0 B_N^1 ... B_N^M
Group N
硬 盘

Results on Public Available Datasets



(b) Data-erpbero (259 images) 33411 pairs 8641 mean points

Method	time(s)	speed(pairs/s)	speedup
Kd-Tree	1.479e4	2.26	1.00×
CasHash	1461.525	22.86	10.12×
SiftGPU	752.164	44.42	19.66×
Ours	34.394	971.42	429.91×

(c) Data-Aos_Hus (811 images) 328455 pairs 7768 mean points

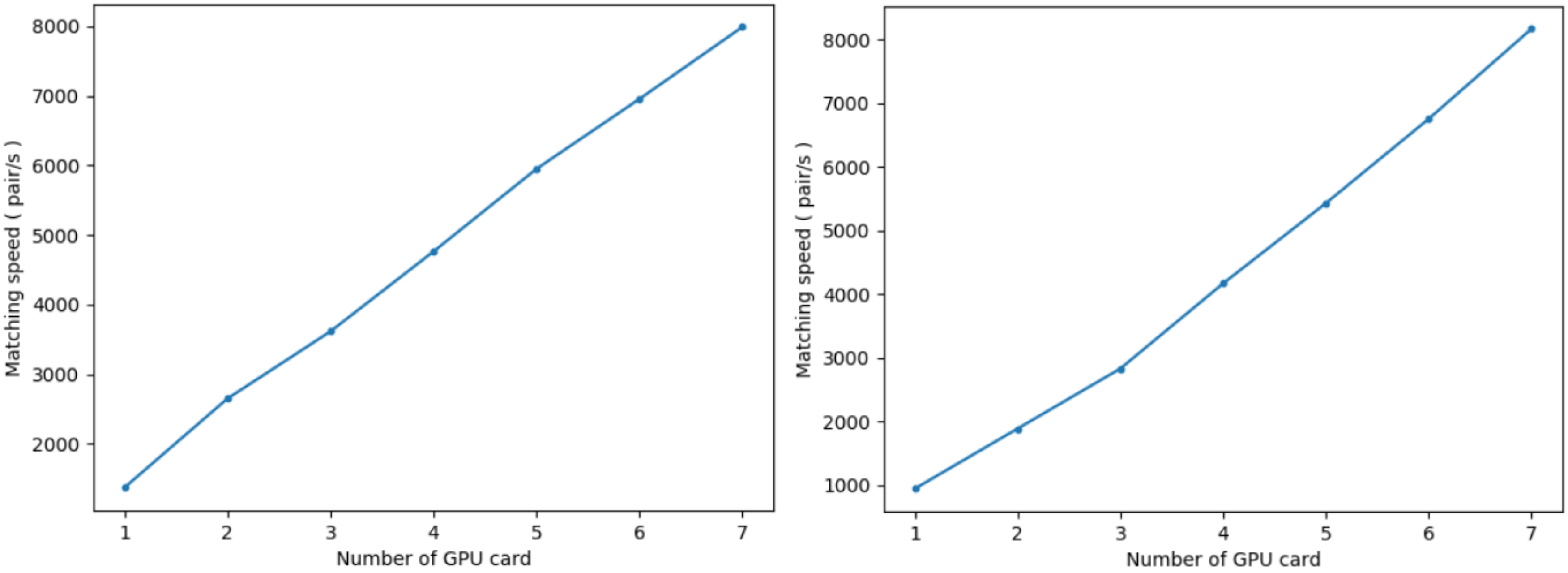
Method	time(s)	speed(pairs/s)	speedup
Kd-Tree	1.456e5	2.26	1.00×
CasHash	2.800e4	11.73	5.20×
SiftGPU	6971.441	47.11	20.89×
Ours	292.541	1122.77	497.81×

(d) Our Method on Some Large Data Set.

Data Set	time(s)	speed(pairs/s)
Dubrovnik6K	1.054e4	1093
Rome16K	1.565e5	1167

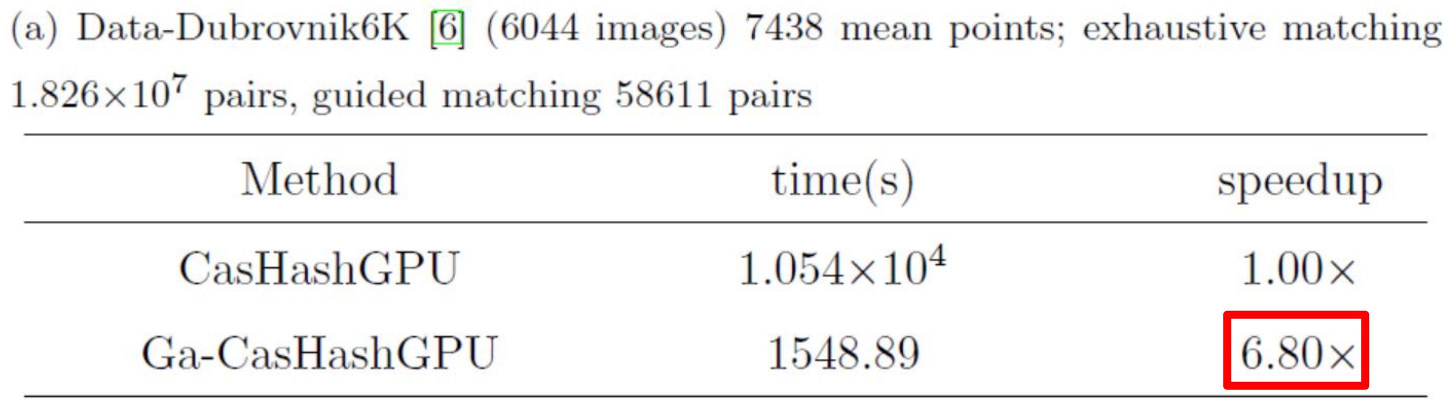
Experiments on large image set

Multiple GPU acceleration



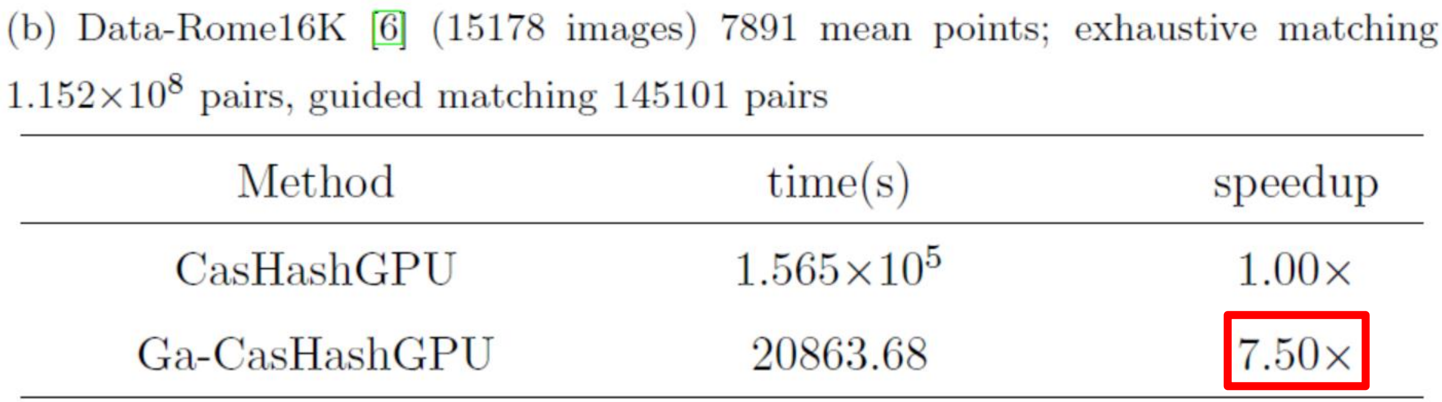
The relationship between the number of GPU card and matching speed. The experiment on Data-Dubrovnik(6K) time is showed in left. The experiment on Data-Rome(16K) time is showed in right.

Geometry-aware CasHashGPU



(a) Data-Dubrovnik6K [6] (6044 images) 7438 mean points; exhaustive matching 1.826×10^7 pairs, guided matching 58611 pairs

Method	time(s)	speedup
CasHashGPU	1.054×10^4	1.00×
Ga-CasHashGPU	1548.89	6.80×

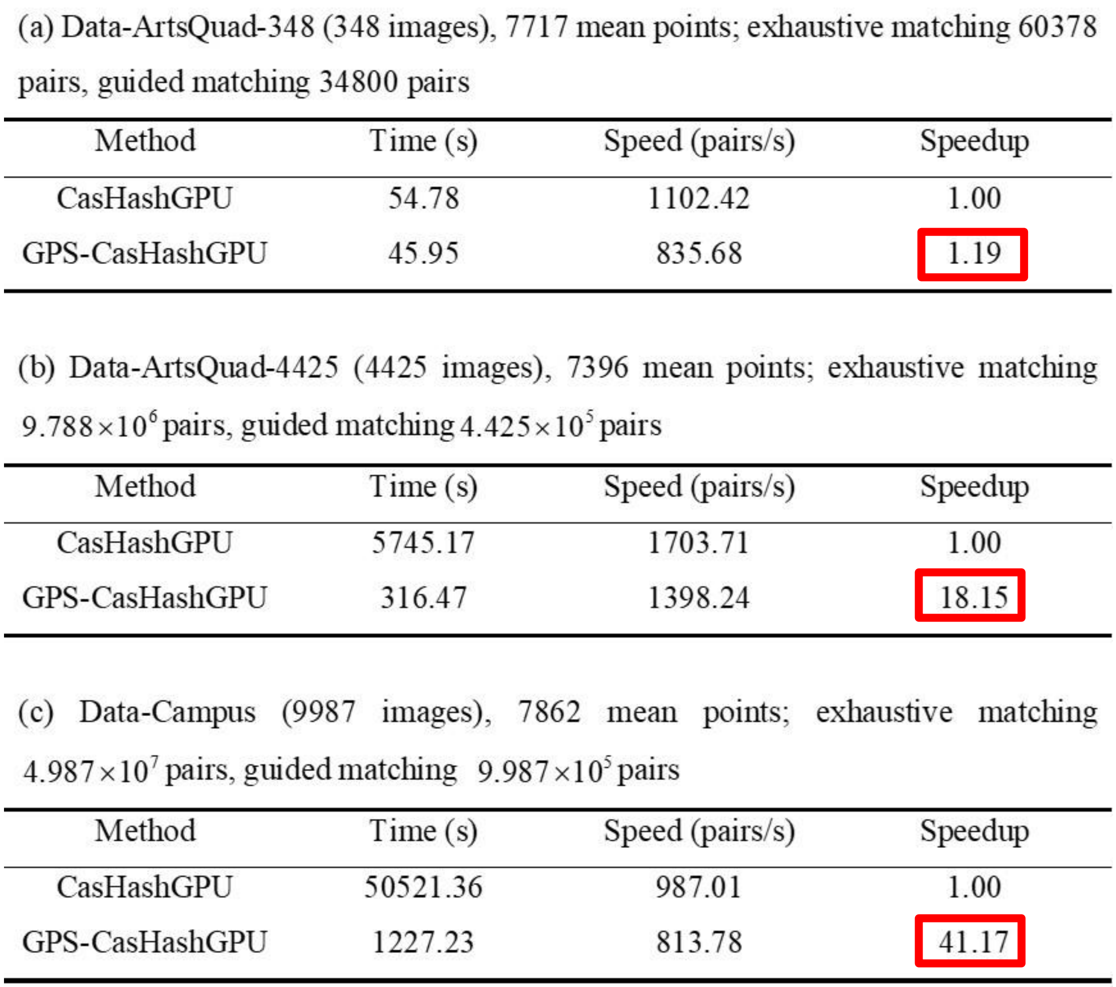


(b) Data-Rome16K [6] (15178 images) 7891 mean points; exhaustive matching 1.152×10^8 pairs, guided matching 145101 pairs

Method	time(s)	speedup
CasHashGPU	1.565×10^5	1.00×
Ga-CasHashGPU	20863.68	7.50×

- ➢ The top 20% scale SIFT features is used to do exhaustive image matching (Wu 2013) by CasHashGPU
- ➢ The information is used to guide the remaining matching procedure

GPS-aware CasHashGPU



(a) Data-ArtsQuad-348 (348 images), 7717 mean points; exhaustive matching 60378 pairs, guided matching 34800 pairs

Method	Time (s)	Speed (pairs/s)	Speedup
CasHashGPU	54.78	1102.42	1.00
GPS-CasHashGPU	45.95	835.68	1.19

(b) Data-ArtsQuad-4425 (4425 images), 7396 mean points; exhaustive matching 9.788×10^{6} pairs, guided matching 4.425×10^{5} pairs

Method	Time (s)	Speed (pairs/s)	Speedup
CasHashGPU	5745.17	1703.71	1.00
GPS-CasHashGPU	316.47	1398.24	18.15

(c) Data-Campus (9987 images), 7862 mean points; exhaustive matching 4.987×10^{7} pairs, guided matching 9.987×10^{5} pairs

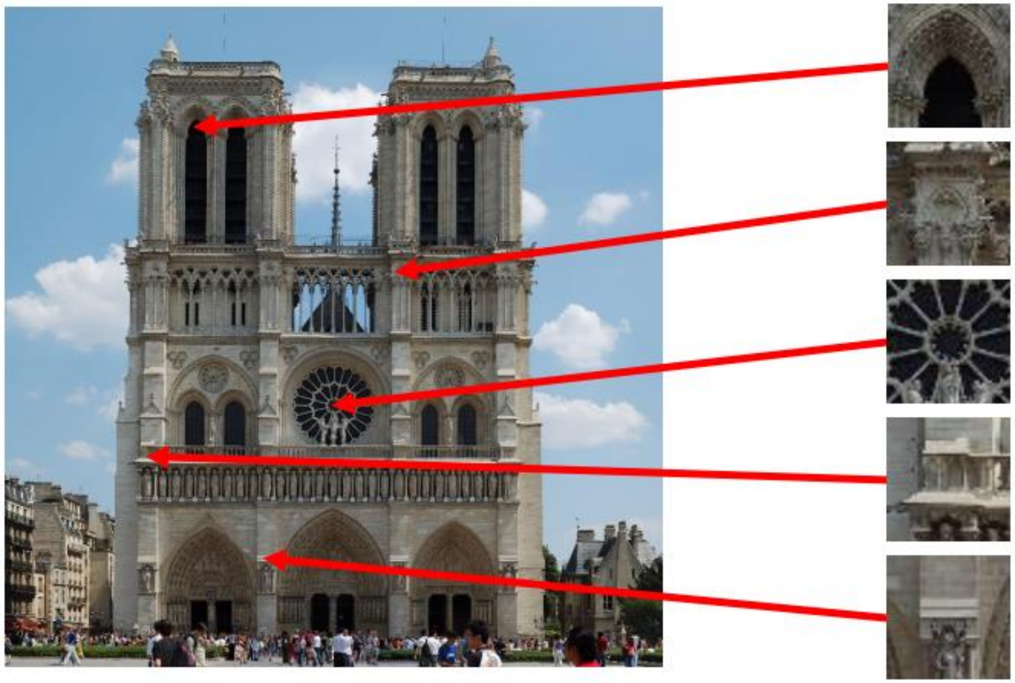
Method	Time (s)	Speed (pairs/s)	Speedup
CasHashGPU	50521.36	987.01	1.00
GPS-CasHashGPU	1227.23	813.78	41.17

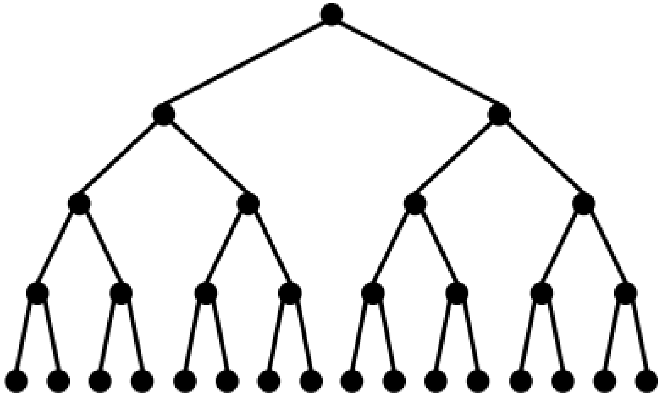
Related works

Vocabulary tree

Fast searching for nearest neighbors.

Bag of words





Vocabulary tree

Our improvement on overlap detection

A fast GPU vocabulary indexing implementation

1DSfM_Roman_Forum, 2360 images			
Stage	**GPU Time(s)**	**CPU Time(s)**	**Speedup factor**
Pre-Process	**0.782**	**0**	-
Search(+Sparse)	**7.854**	**267.478**	34.0
Weight	**0.005**	**0.220**	-
Normalize	**0.182**	**0.544**	-
Score	**0.506**	**1.027**	-
Data Copy	**2.444**	**0**	-
Others	**0.501**	**0.242**	-
Total	**12.274**	**269.511**	**21.9**

1DSfM_Vienna_Cathedral, 6280 images			
Stage	**GPU Time(s)**	**CPU Time(s)**	**Speedup factor**
Pre-Process	**0.892**	**0**	-
Search(+Sparse)	**29.317**	**837.375**	28.5
Weight	**0.023**	**0.346**	-
Normalize	**0.466**	**1.284**	-
Score	**5.821**	**19.399**	-
Data Copy	**6.852**	**0**	-
Others	**1.910**	**0.930**	-
Total	**45.281**	**859.334**	**18.9**

All the tests are performed on a machine with 256GB RAM, one Intel Xeon E5-2630 v3 @ 2.40GHz CPU and one NVIDIA GeForce GTX Titan X GPU card

Expect to process 10000 images within 1 minute.

GPU-based F-matrix and H-matrix estimation

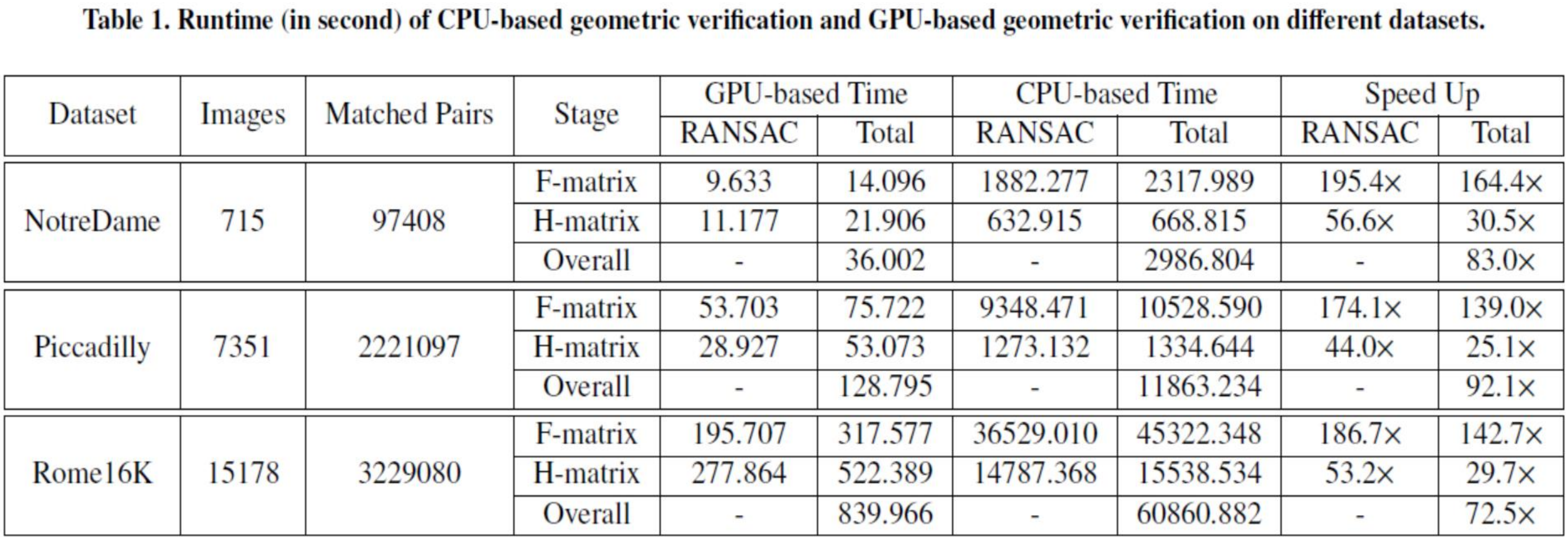


Table 1. Runtime (in second) of CPU-based geometric verification and GPU-based geometric verification on different datasets.

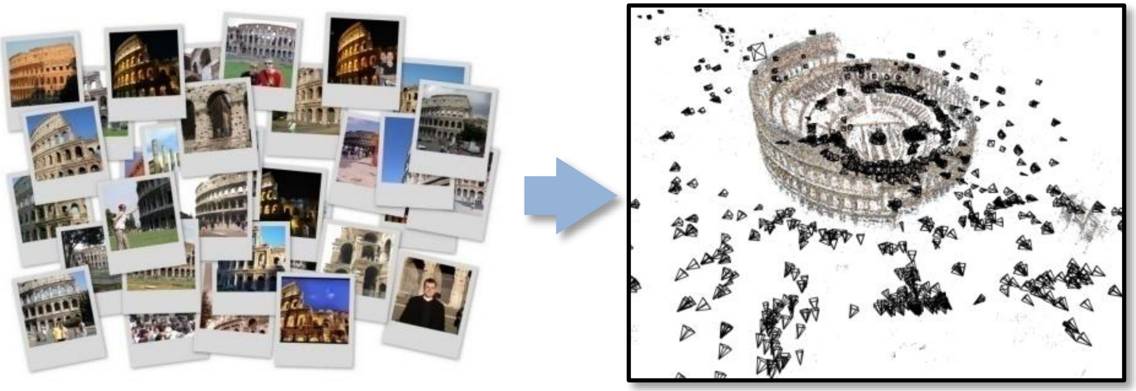
Dataset	Images	Matched Pairs	Stage	GPU-based Time		CPU-based Time		Speed Up	
				RANSAC	Total	RANSAC	Total	RANSAC	Total
NotreDame	715	97408	F-matrix	9.633	14.096	1882.277	2317.989	195.4×	164.4×
			H-matrix	11.177	21.906	632.915	668.815	56.6×	30.5×
			Overall	-	36.002	-	2986.804	-	83.0×
Piccadilly	7351	2221097	F-matrix	53.703	75.722	9348.471	10528.590	174.1×	139.0×
			H-matrix	28.927	53.073	1273.132	1334.644	44.0×	25.1×
			Overall	-	128.795	-	11863.234	-	92.1×
Rome16K	15178	3229080	F-matrix	195.707	317.577	36529.010	45322.348	186.7×	142.7×
			H-matrix	277.864	522.389	14787.368	15538.534	53.2×	29.7×
			Overall	-	839.966	-	60860.882	-	72.5×

PART 3

Multiple starting points selection and data partition for large scale SFM

Structure from Motion

Giving a set of images, estimate the camera poses and the sparse 3D structure.



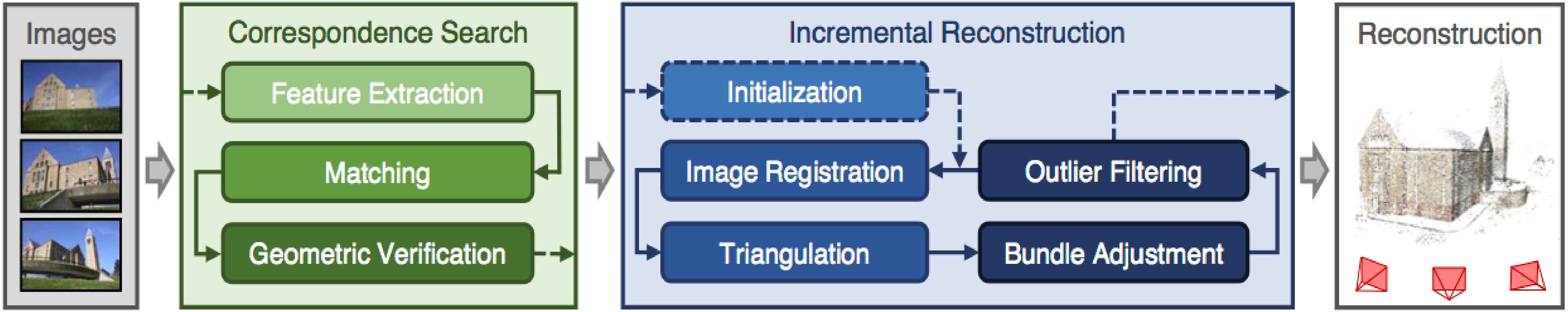
***Scene geometry (structure):** Given 2D point matches in two or more images, where are the corresponding points in 3D?*

***Correspondence (matching):** Given a point in just one image, how does it constrain the position of the corresponding point in another image?*

***Camera geometry (motion):** Given a set of corresponding points in two or more images, what are the camera matrices for these views?*

Structure from Motion

The general pipeline of the SfM algorithm



Structure from Motion

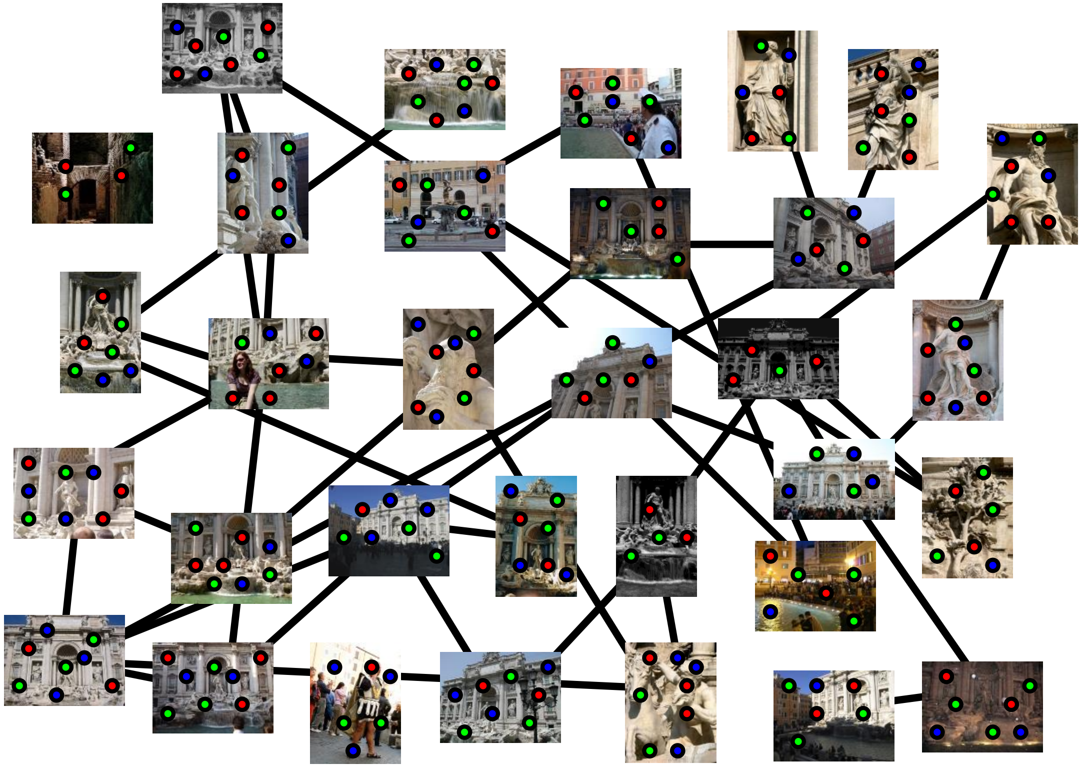
Matching graph construction

Structure from Motion

Matching graph construction

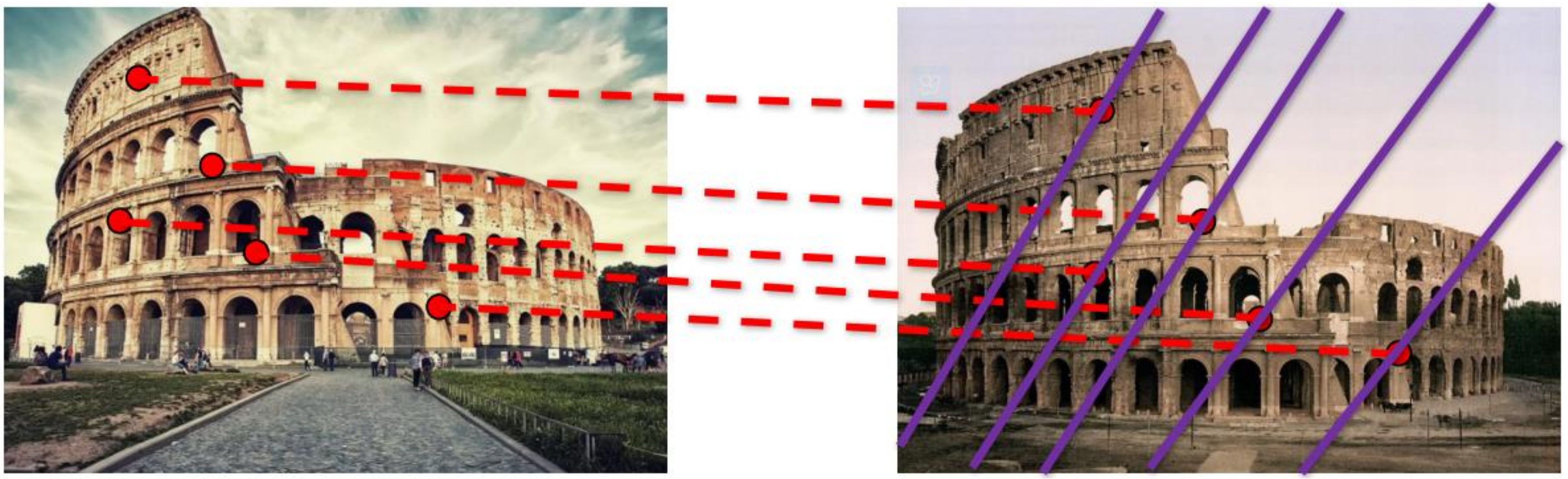
Structure from Motion

Matching graph construction



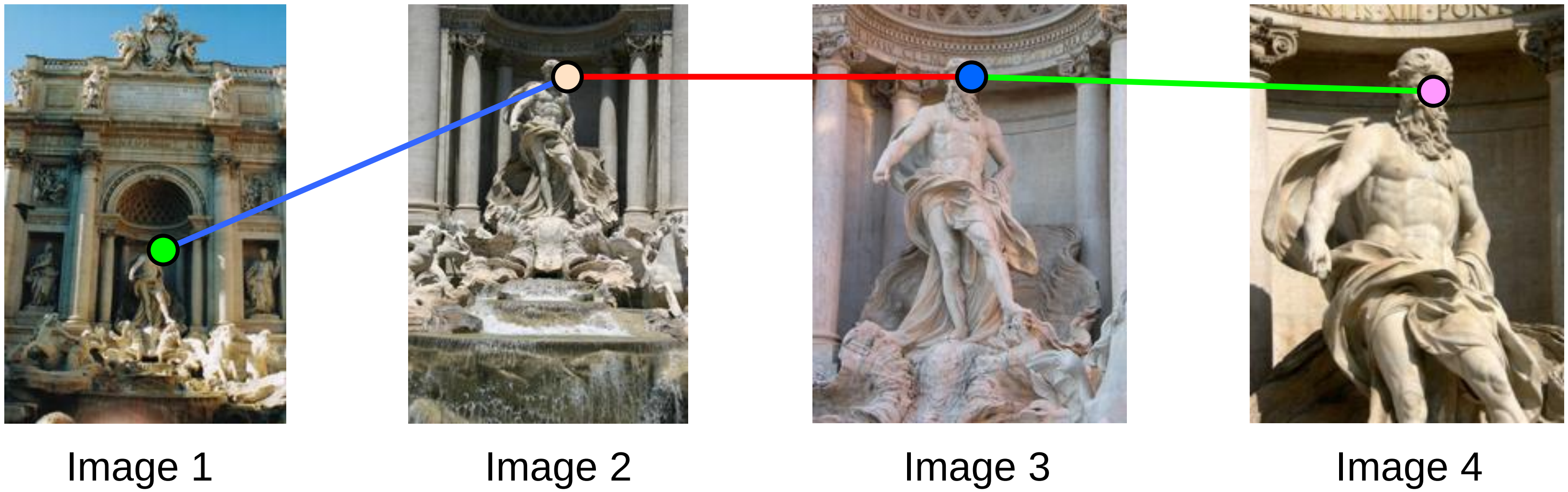
Structure from Motion

Epipolar Geometry estimated by RANSAC



Structure from Motion

Build tracks from matches



- □ Link up matches between pairs of images into tracks between multiple images
- □ Each track corresponds to a 3D point

Structure from Motion

Choose two views

- *They have the most number of feature correspondences*
- *They have wide baseline (The baseline can be measured by the inlier ratio of a planar homography)*

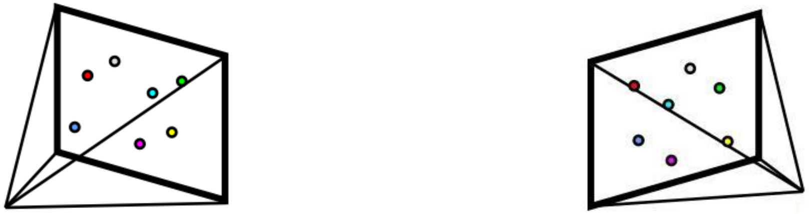
Structure from Motion

Estimate relative pose using two-view geometry

- *Camera intrinsics known*

  Essential matrix, ***E*** (5 points)
- *Camera intrinsics unknown*

  Fundamental matrix, ***F*** (7 points)



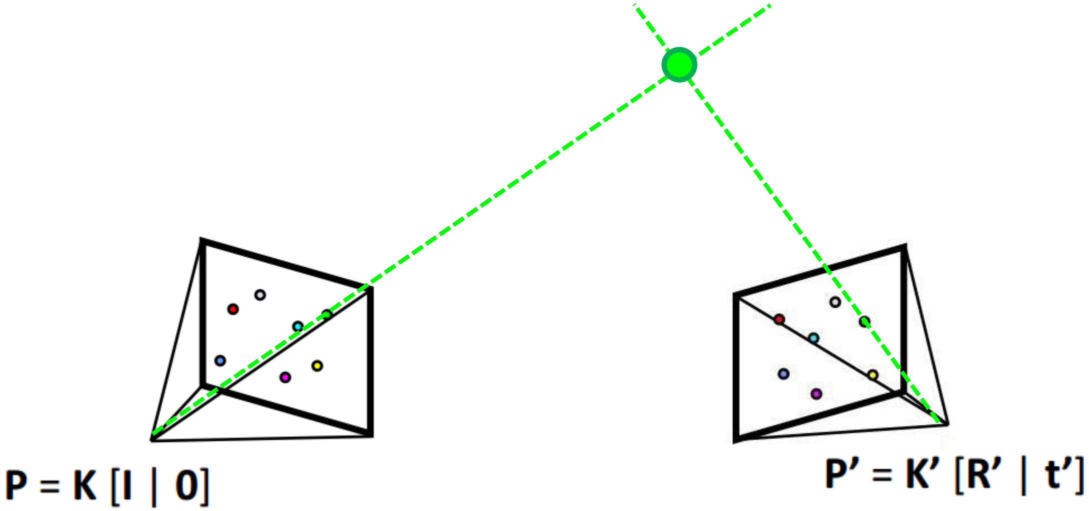
$P = K [I \mid 0]$

$P' = K' [R' \mid t']$

Structure from Motion

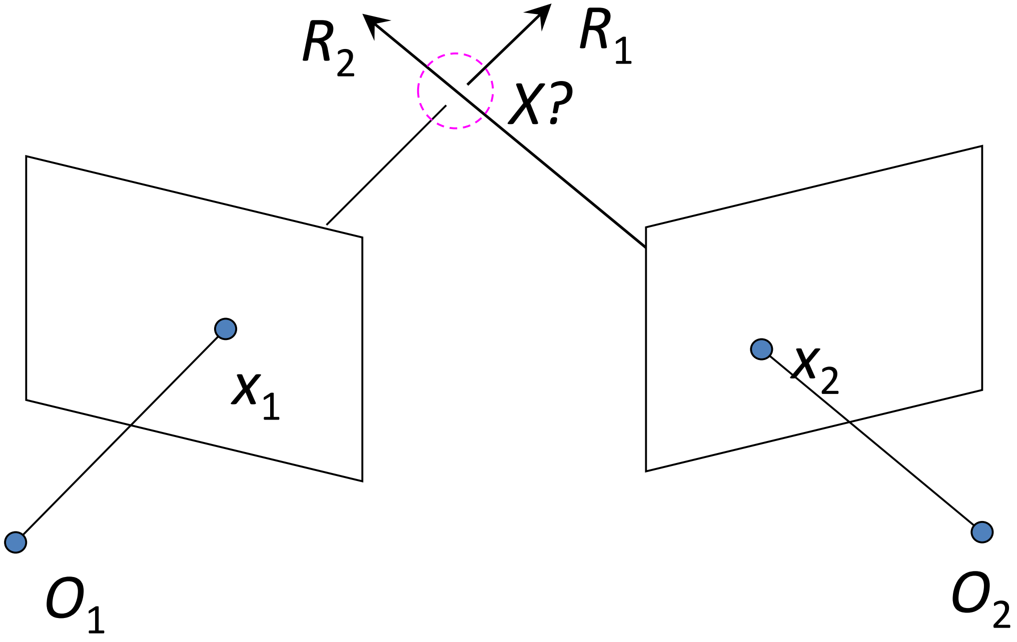
Triangulate inlier correspondences

- *Given projections of a 3D point in two or more images (with known camera matrices), find the coordinates of the point*



Structure from Motion

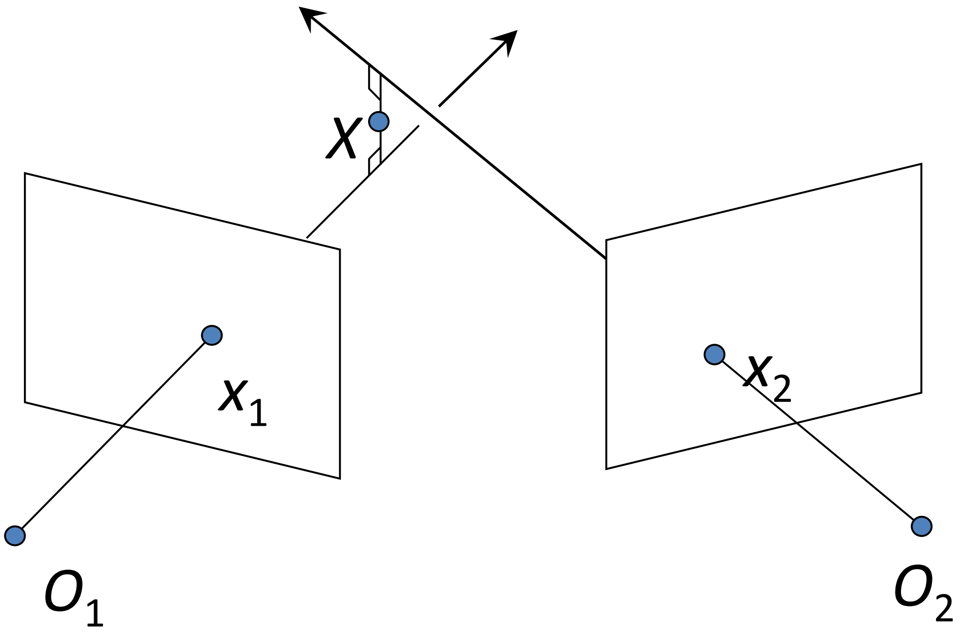
Triangulation



- *We want to intersect the two visual rays corresponding to x1 and x2, but because of noise and numerical errors, they don’t meet exactly*

Structure from Motion

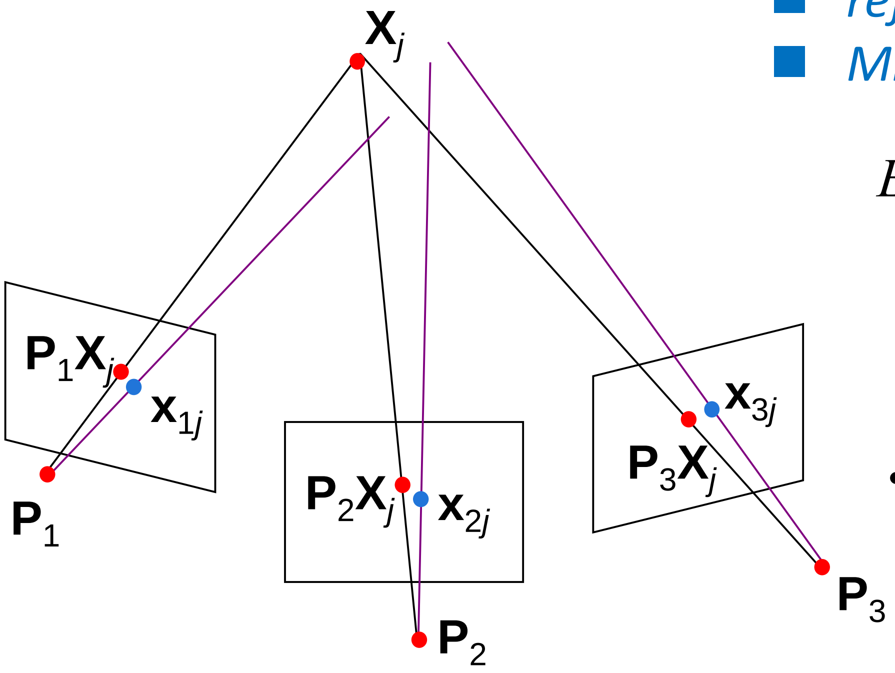
Triangulation



- *Find shortest segment connecting the two viewing rays and let X be the midpoint of that segment*

Structure from Motion

Bundle Adjustment



- *refine 3D points*
- *refine camera parameters*
- *Minimize reprojection error:*

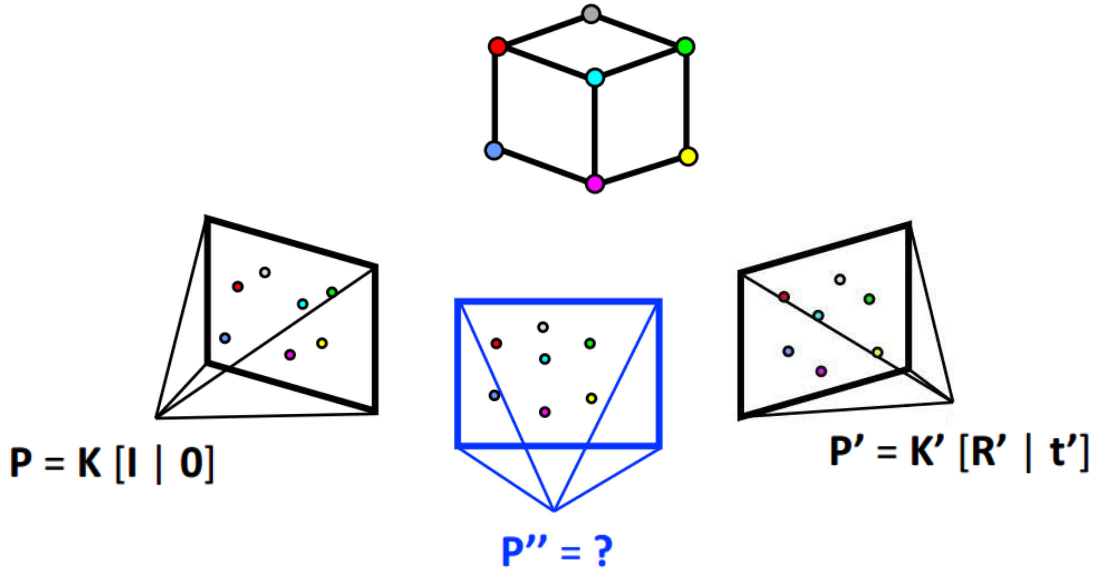
$$E(\mathbf{P},\mathbf{X}) = \sum_{i=1}^{m}\sum_{j=1}^{n} w_{ij}\, D\left(\mathbf{x}_{ij}, \mathbf{P}_i\mathbf{X}_j\right)^2$$

w_{ij} indicator variable for visibility of point $\mathbf{X}_j$ in camera $\mathbf{P}_i$

- Minimizing this function is called bundle adjustment
  - Optimized using non-linear least squares, e.g. Levenberg-Marquardt

Structure from Motion

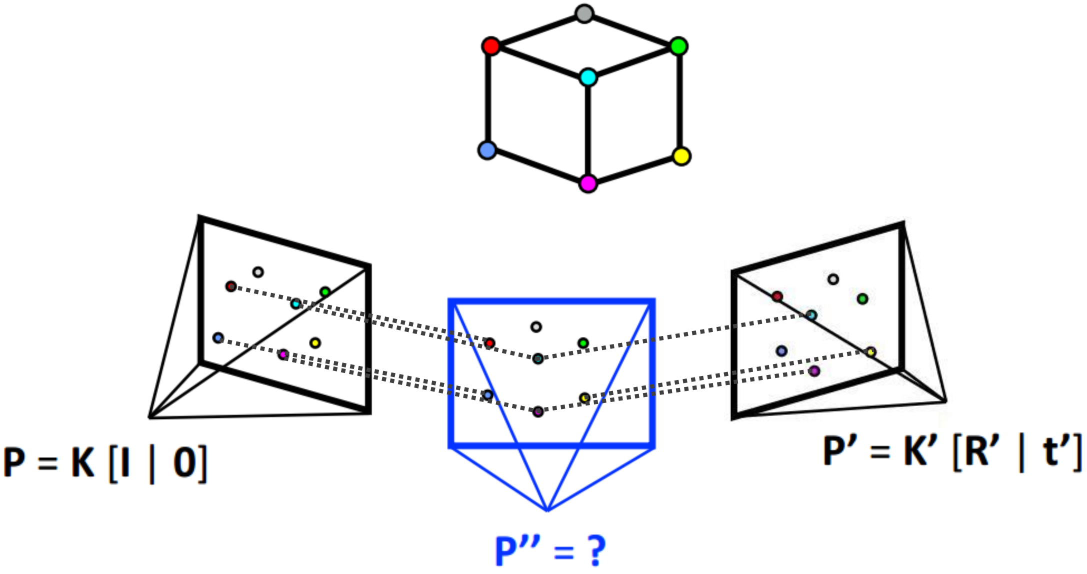
Add new cameras



Structure from Motion

Add new cameras

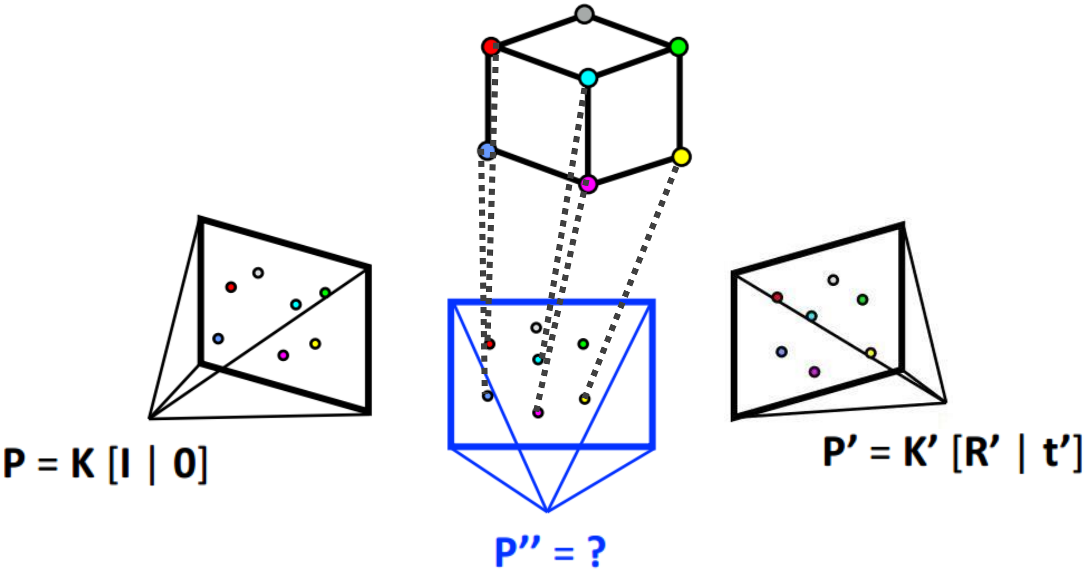
- *2D-2D correspondences*



Structure from Motion

Add new cameras

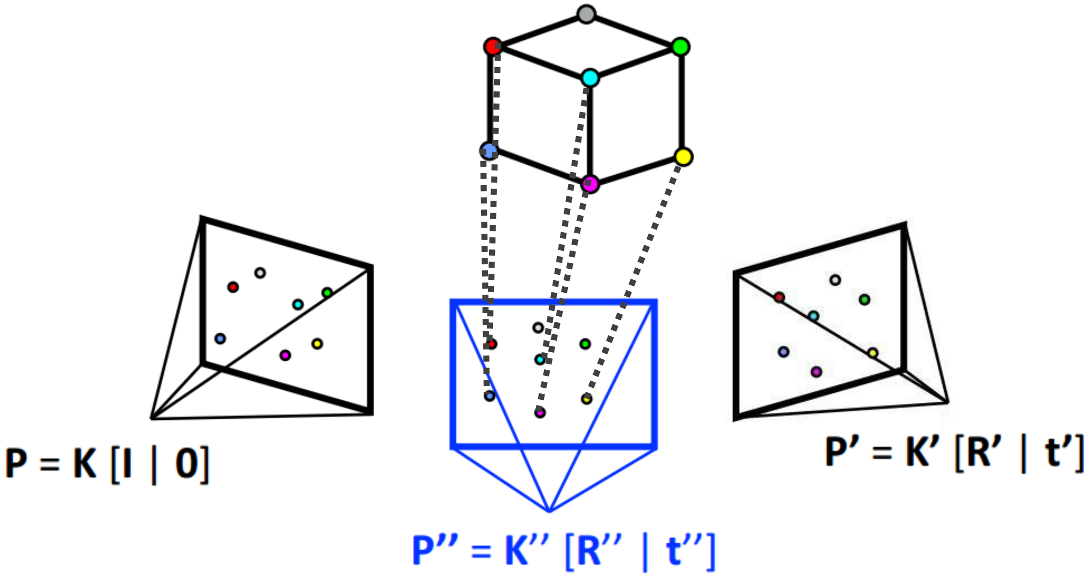
- *Feature tracks help a lot*
- *Maximize number of 2D-3D correspondences*



Structure from Motion

Add new cameras

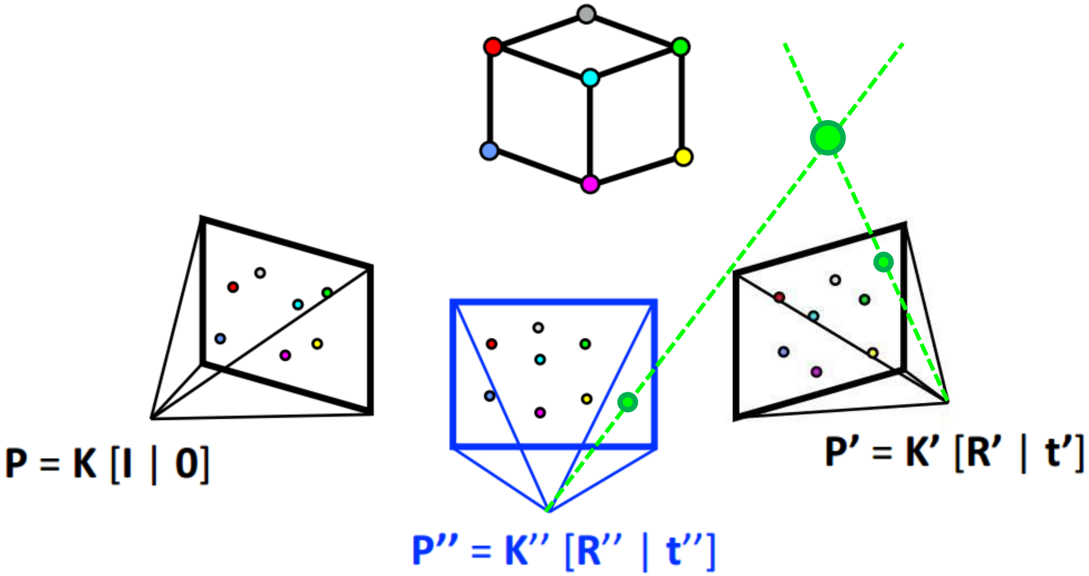
- *Solve Perspective-n-Point problem*



Structure from Motion

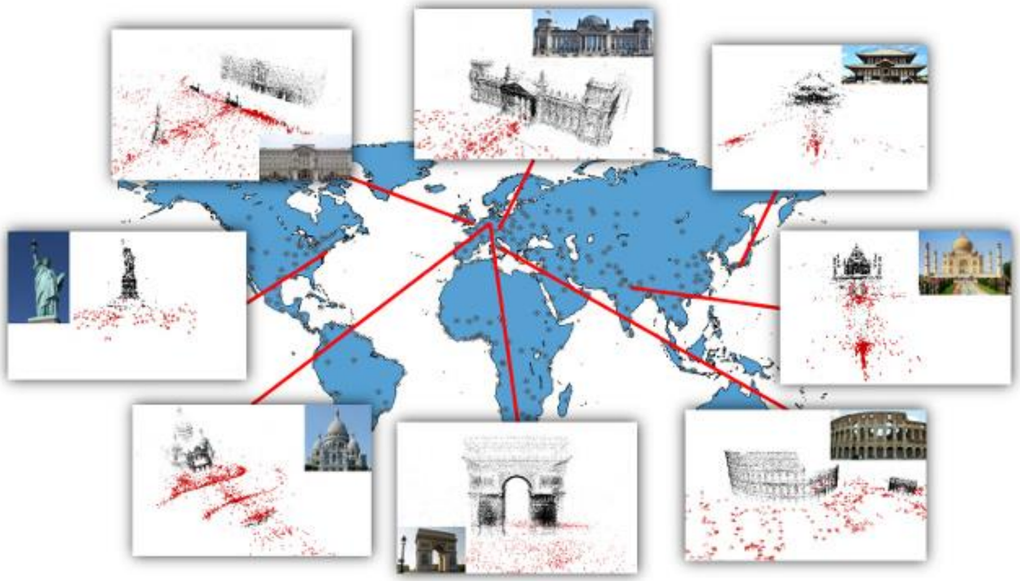
Add new cameras

- *Triangulate new points*
- *Bundle adjustment*



Difficulties

The difficulties in SfM for large scale unordered images.



***100 million** images on Yahoo*

1. Explosive image data:

- *Image matching is time consuming*
- *Sequentially adding them is time consuming*
- *How to partition the image set properly?*

Difficulties

The difficulties in SfM for large scale unordered images.

VS

unstructured

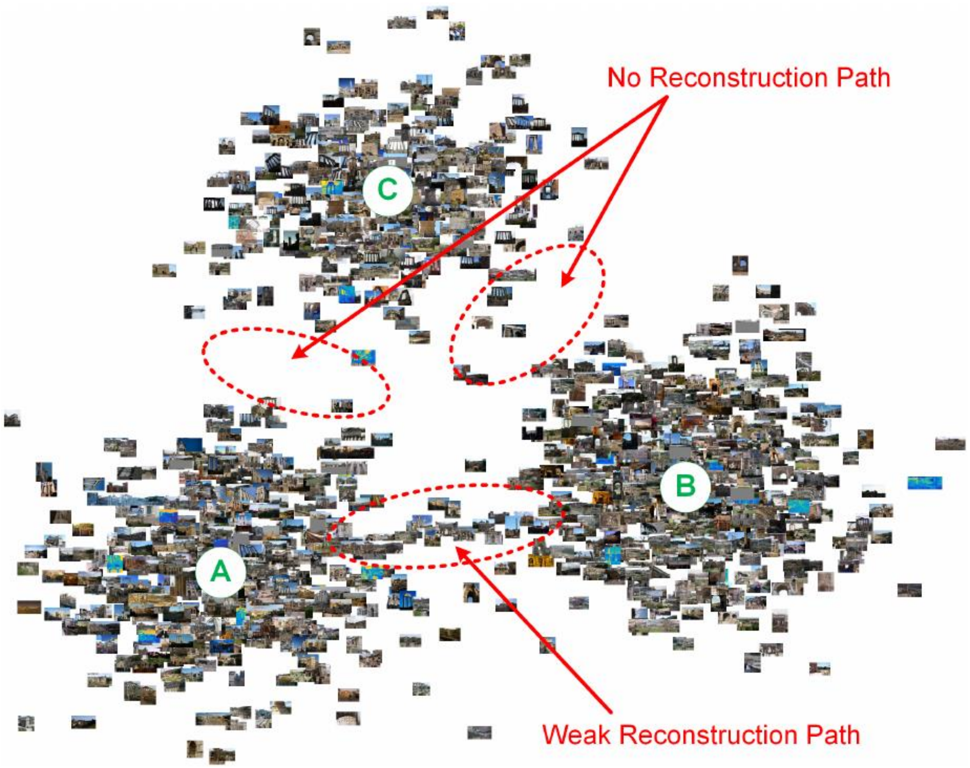
structured

2. Unordered:

- *Unknown neighborhood, unknown scene overlap*
- *Burdensome image matching procedure*

Difficulties

The difficulties in SfM for large scale unordered images.

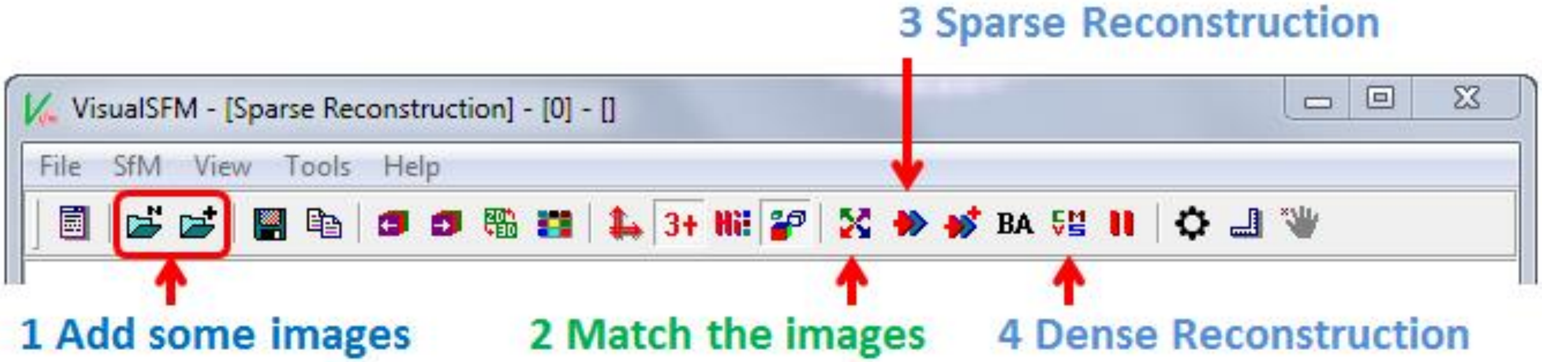


3. Non-uniform distributed images:

- *Weak or no overlap between images*
- *If start from C, neither A nor B could be reconstructed*
- *If start from A or B, large error could be accumulated*

Linear time SfM

Run a new SfM procedure in the remaining images.



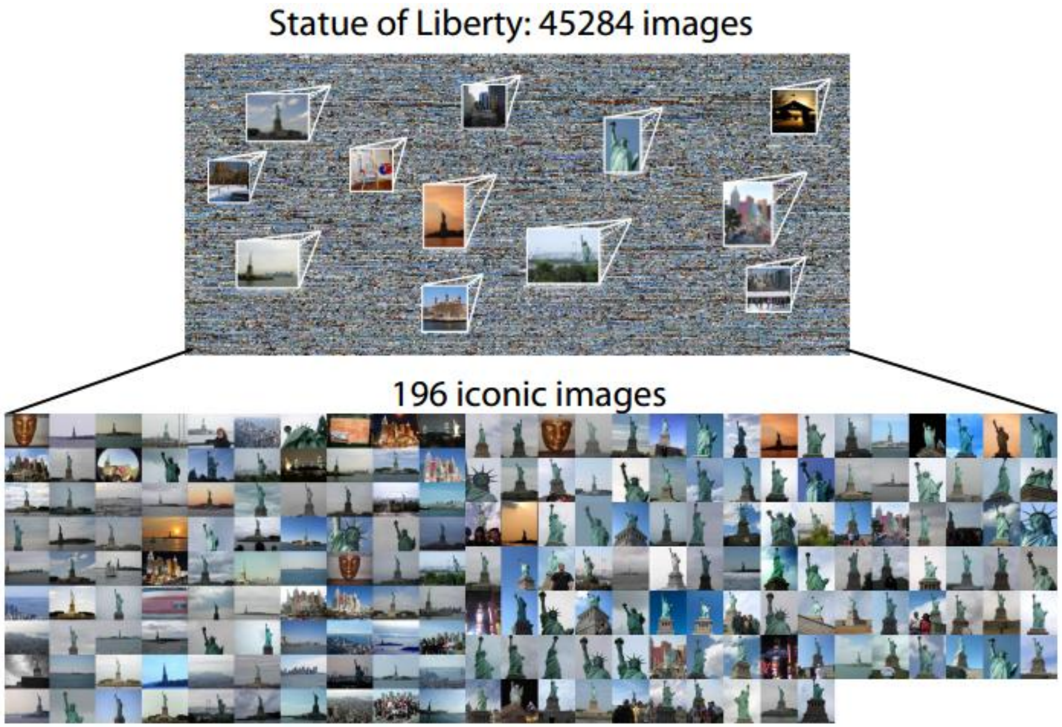
- ◆ A linear-time incremental SfM system including: GPU-based SIFT, GPU-based BA
- ◆ Restarting a new SfM procedure from the remaining images.
- ◆ Models are not produced in parallel.
- ◆ Good models might be reconstructed after many failures, which wastes a lot of time.

Wu C., VisualSFM, http://ccwu.me/vsfm/.
Wu C., et al., 3DV 2013, CVPR2011.
Schonberger J. et al., CVPR2016.

Related works

Iconic Scene Graph

Summarize the scene by extracting iconic images.



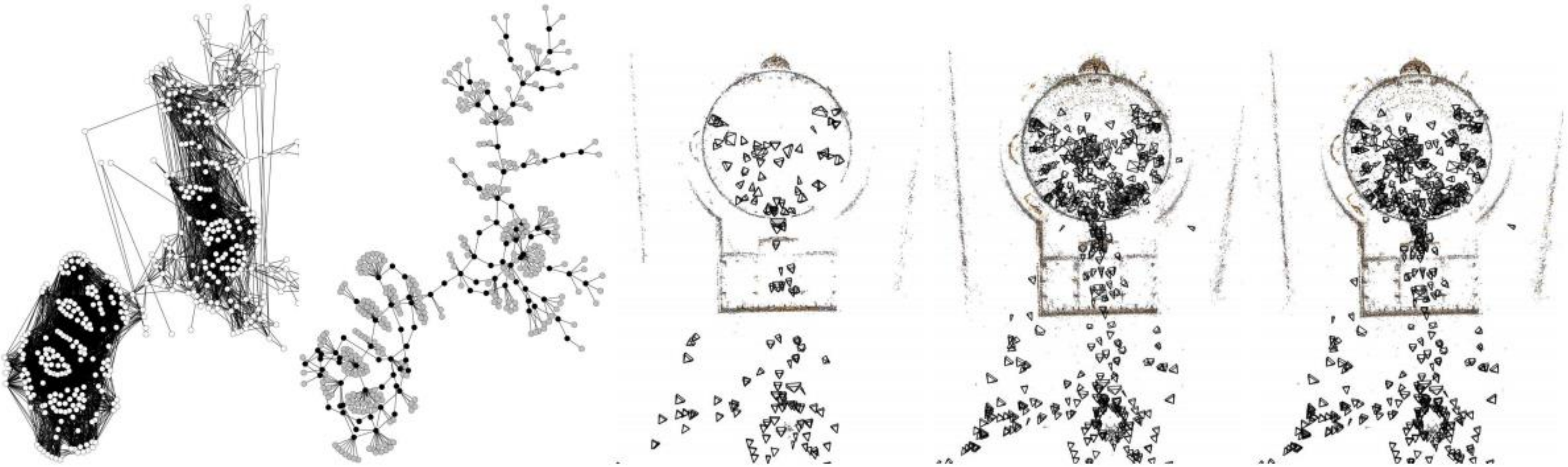
- k-means clustering with gist descriptors.
- Select an iconic image for each cluster.
- Run normalized cuts to break iconic scene graph into smaller components.
- Data discontinuity not solved & the number of clusters is hard to know in advance

X. Li, et al. Modeling and recognition of landmark image collections using iconic scene graphs. ECCV 2008.
J.-M. Frahm et al. Building rome on a cloudless day. ECCV 2010.
J. Heinly, et al. Reconstructing the world in six days. In CVPR, 2015, pages 3287–3295.
J. L. Schonberger et al. Structure-from-motion revisited. CVPR2016.

Related works

Skeletal Graph

Find a subset of skeletal graphs from the image matching graph.

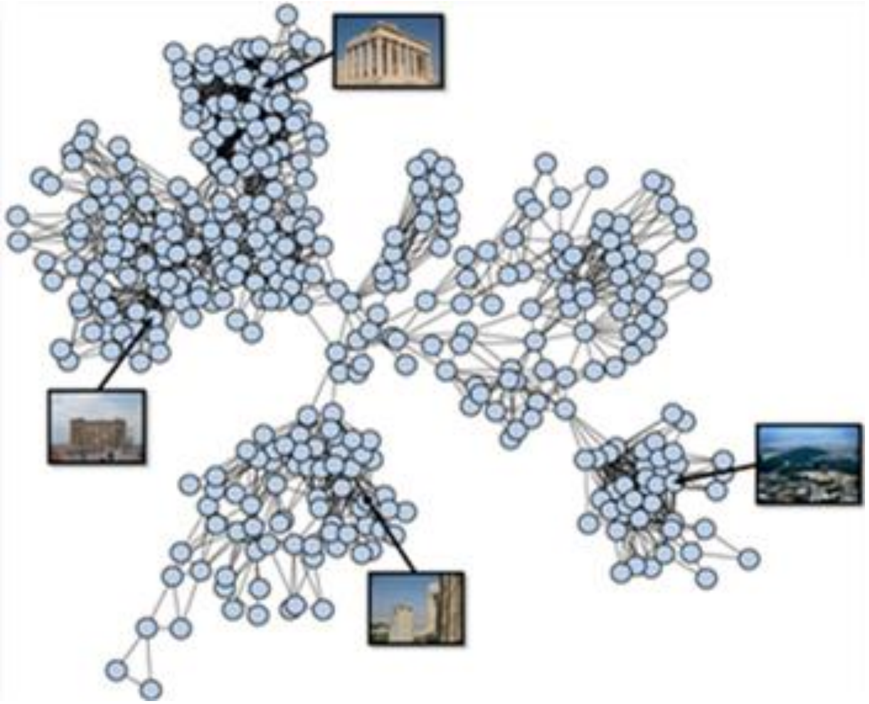


- Reconstructs the skeletal set, and adds the remaining images using pose.
- Drastically reduces the number of parameters that are considered, resulting in dramatic speedups.
- The skeletal image set approximates the coverage and robustness of the full set.
- Data discontinuity not solved

N. Snavely, et al. Skeletal graphs for efficient structure from motion. CVPR2008.
S. Agarwal, et al. Building rome in a day. ICCV2009.

Preliminary

The matching graph



Two kinds of matching graphs:

- □ The similarity matching graph ***S***
- □ The difference matching graph ***D***

$$s_{ij} = \frac{n_{ij}}{n_i \cup n_j}$$ weigth for ***S***

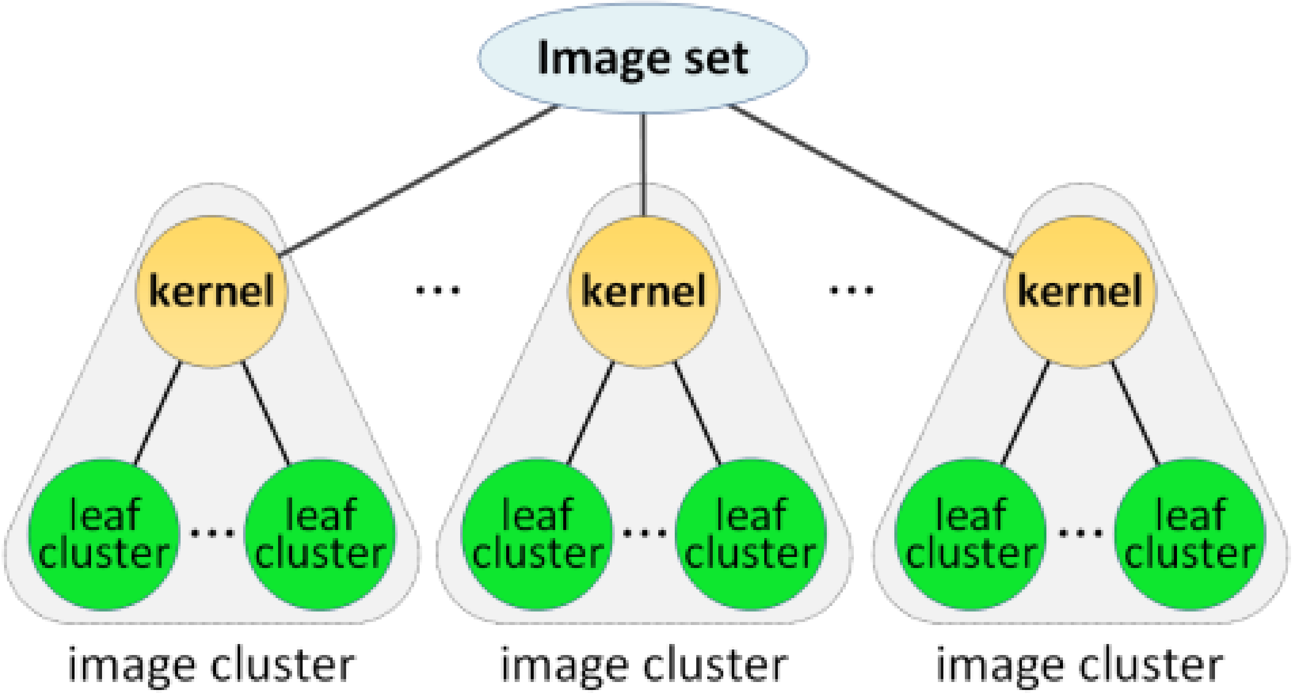
$$d_{ij} = 1 - s_{ij}$$ weigth for ***D***

An image matching graph is a weighted undirected graph. Each node represents an image, and an edge indicates scene overlap between two images.

Kun Sun, Wenbing Tao*, Multiple Starting Points Selection and Data Partitioning for Accurate, Efficient Structure from Motion, arXiv:1612.07153.

Preliminary

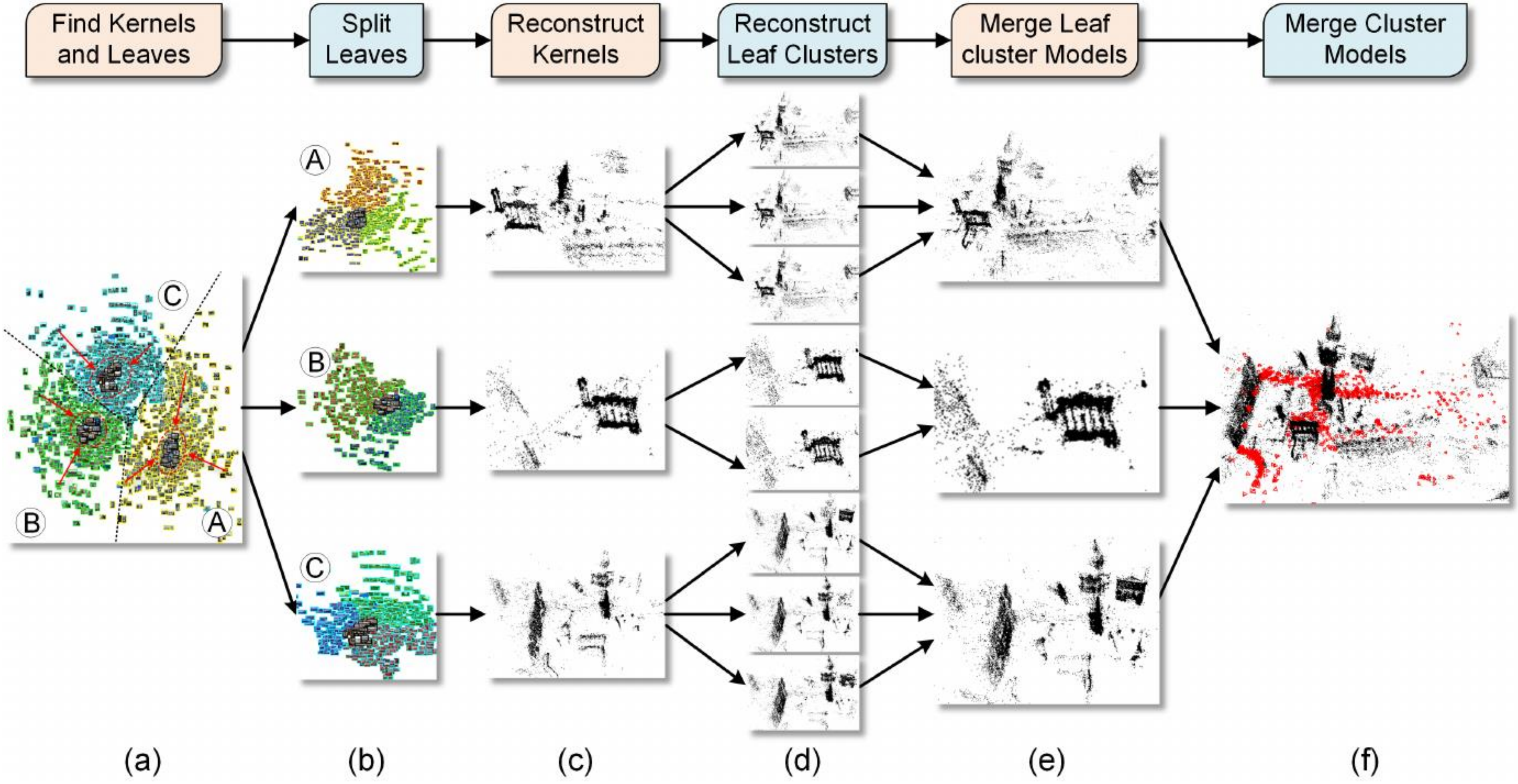
The trilaminar multiway reconstruction tree



The whole image set is partitioned into several image clusters. Each image cluster contains a kernel and several leaf clusters.

Overall Flowchart

The overall flowchart of the proposed method.



Algorithm

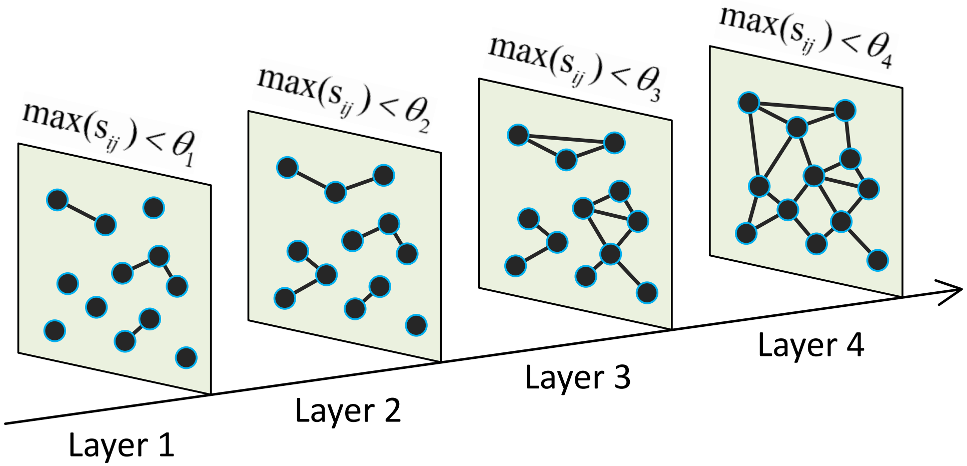
Key step 1: Finding Kernels

Adopt a greedy strategy to find kernels in a layered graph

- Compute a set of thresholds from

$$\theta_i = a + \frac{b-a}{1.5^{i-1}}, i \in 1, 2, \ldots, k$$

- Divide the similarity matching graph ***S*** into k layers
- Find connected components in each layer
- Remove already found kernels from subsequent layers


$\max(s_{ij}) < \theta_1$
$\max(s_{ij}) < \theta_2$
$\max(s_{ij}) < \theta_3$
$\max(s_{ij}) < \theta_4$
Layer 1
Layer 2
Layer 3
Layer 4

✓ Kernels are found at places where images are densely distributed.

✓ Kernels are used to reconstruct base models of the scene.

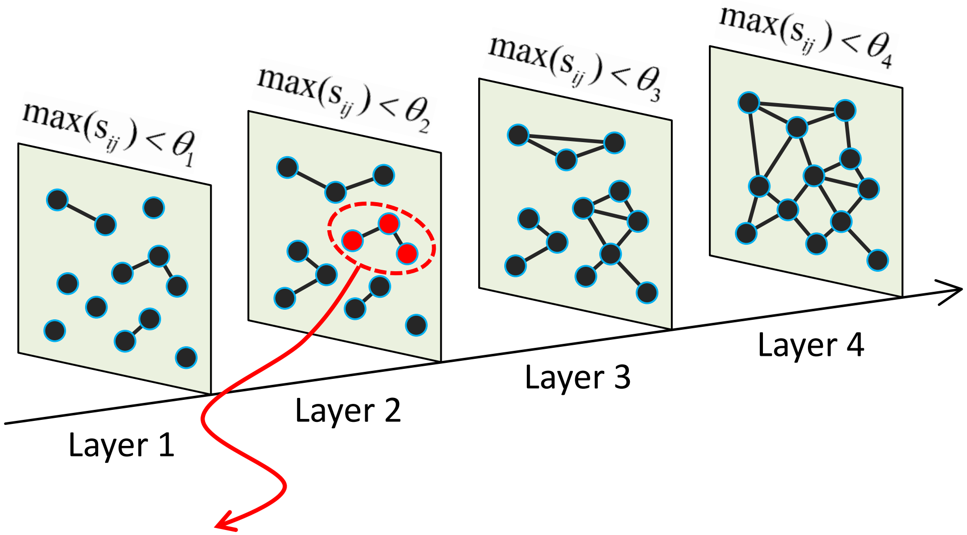
Key step 1: Finding Kernels

Adopt a greedy strategy to find kernels in a layered graph

- Compute a set of thresholds from

$$\theta_i = a + \frac{b-a}{1.5^{i-1}}, i \in 1, 2, \ldots, k$$

- Divide the similarity matching graph ***S*** into k layers
- Find connected components in each layer
- Remove already found kernels from subsequent layers


$\max(s_{ij}) < \theta_1$
$\max(s_{ij}) < \theta_2$
$\max(s_{ij}) < \theta_3$
$\max(s_{ij}) < \theta_4$
Layer 1
Layer 2
Layer 3
Layer 4

Too small component, continue to the next layer.

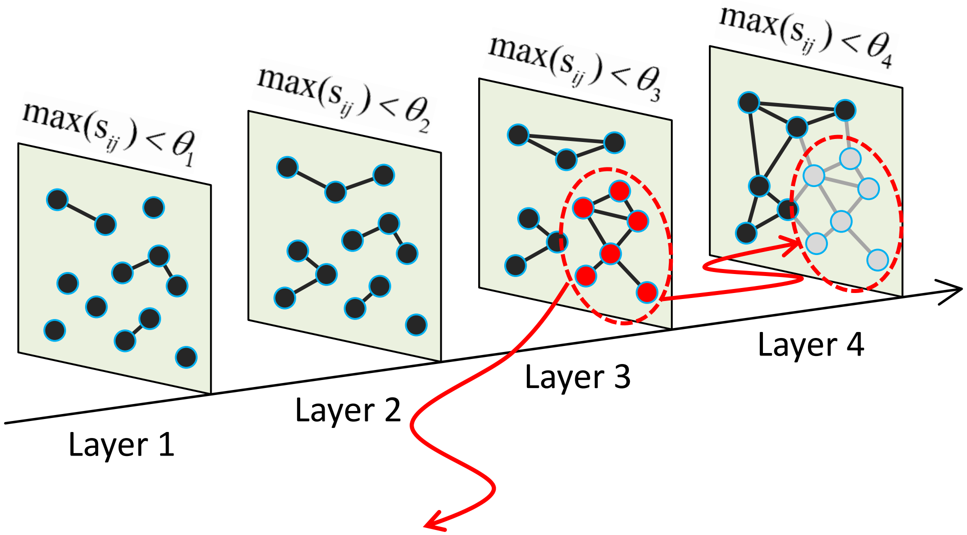
Key step 1: Finding Kernels

Adopt a greedy strategy to find kernels in a layered graph

- Compute a set of thresholds from

$$\theta_i = a + \frac{b-a}{1.5^{i-1}}, i \in 1, 2, \ldots, k$$

- Divide the similarity matching graph ***S*** into k layers
- Find connected components in each layer
- Remove already found kernels from subsequent layers


$\max(s_{ij}) < \theta_1$
$\max(s_{ij}) < \theta_2$
$\max(s_{ij}) < \theta_3$
$\max(s_{ij}) < \theta_4$
Layer 1
Layer 2
Layer 3
Layer 4

Good kernel, remove vertexes and edges from subsequent layers.

Key step 2: Select An Exemplar Image

Select an exemplar image in each valid kernel

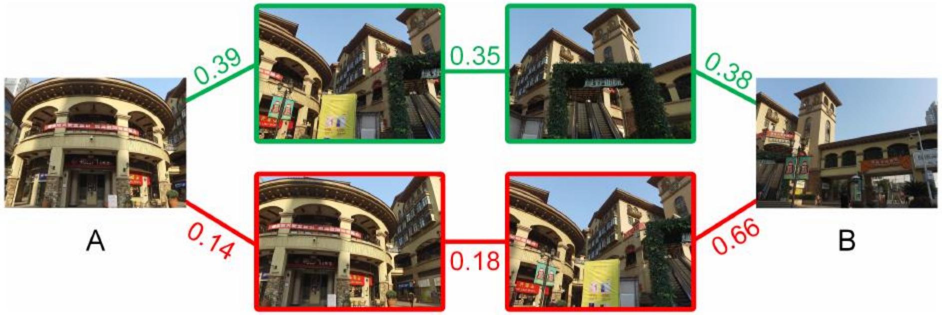
- The Affinity Propagation (AP) clustering algorithm is applied to images in each kernel.
- All the centers and their adjacent neighbors on the similarity graph are treated as the candidates for the exemplar image.
- Select the image with the highest score.

$$\delta(v) = \underbrace{h_{deg}(v)}_{\text{Degree of this vertex}} + \beta_1 \cdot \overbrace{h_{sim}(v)}^{\text{Average similarity with its neighbors of this vertex}} + \beta_2 \cdot \underbrace{h_{ndeg}(v)}_{\text{Average degree of the neighbors of this vertex}}$$

The exemplar image will be used as the starting image in the reconstruction

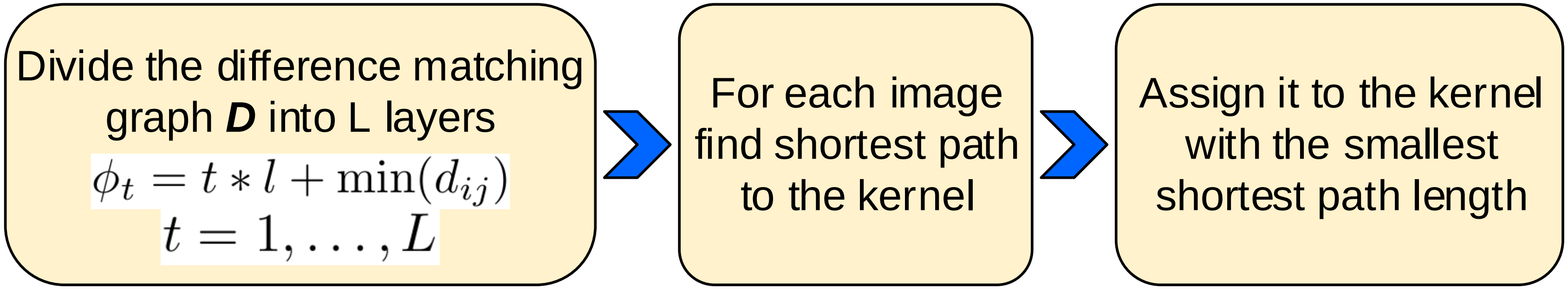
Key step 3: Finding Image Clusters

Clustering images according to their optimal reconstruction path to the kernels



- Proposed the concept of optimal reconstruction path
  - large and equal overlapping
  - the maximum difference between adjacent images should be minimized

- □ Images are clustered by treating the kernels as centers.
- □ A Multi-layer Shortest Path (MSP) algorithm is proposed to find the optimal reconstruction paths from each image to the kernels.



Divide the difference matching graph ***D*** into L layers

$$\phi_t = t * l + \min(d_{ij})$$

$$t = 1, \ldots, L$$

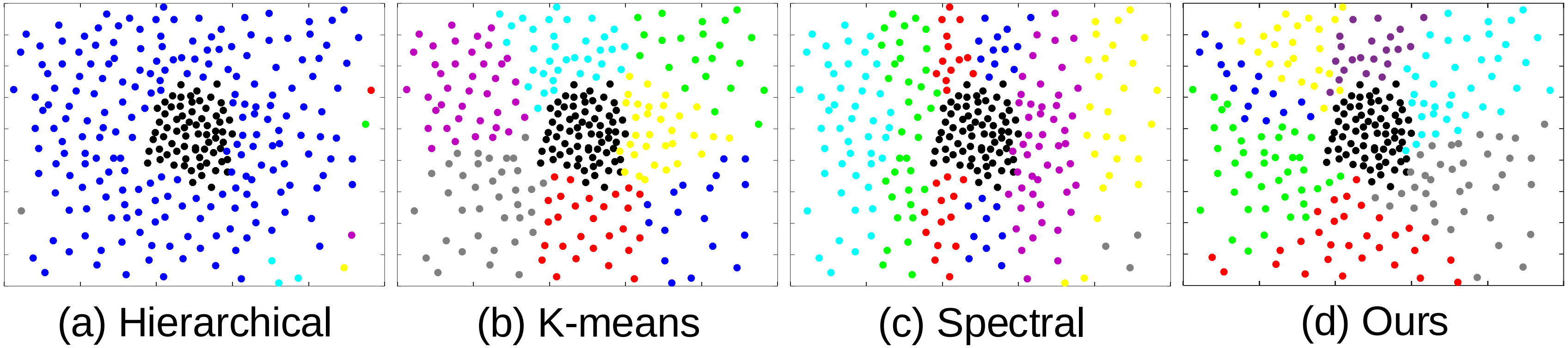
For each image find shortest path to the kernel

Assign it to the kernel with the smallest shortest path length

Key step 4: Finding Leaf Clusters

Find Leaf Clusters using Radial Agglomerate Clustering

Leaves are split so that they can be reconstructed in parallel.



(a) Hierarchical

(b) K-means

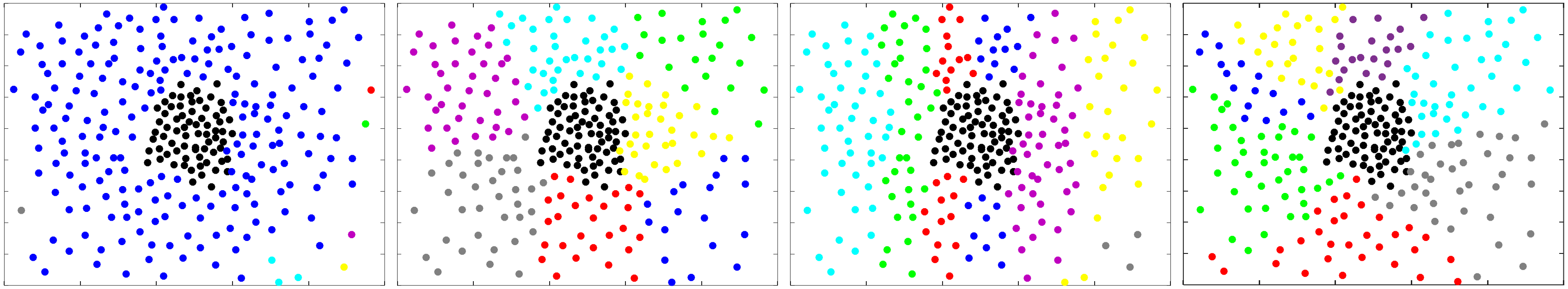
(c) Spectral

(d) Ours

Key step 4: Finding Leaf Clusters

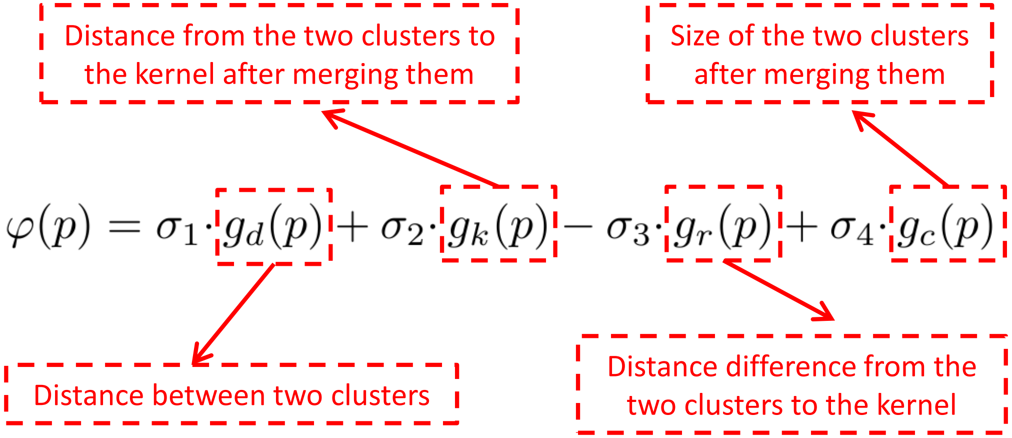
Find Leaf Clusters using Radial Agglomerate Clustering

Leaves are split so that they can be reconstructed in parallel.



(a) Hierarchical (b) K-means (c) Spectral (d) Ours

- Three conditions
  - Images within each leaf cluster should have considerable overlap
  - Each leaf cluster should have strong overlap with the kernel
  - The size for these leaf clusters should be balanced
- Each leaf is initialized as a cluster and each step two of them with the smallest cost is merged.



$$\varphi(p) = \sigma_1 \cdot \boxed{g_d(p)} + \sigma_2 \cdot \boxed{g_k(p)} - \sigma_3 \cdot \boxed{g_r(p)} + \sigma_4 \cdot \boxed{g_c(p)}$$

Distance from the two clusters to the kernel after merging them

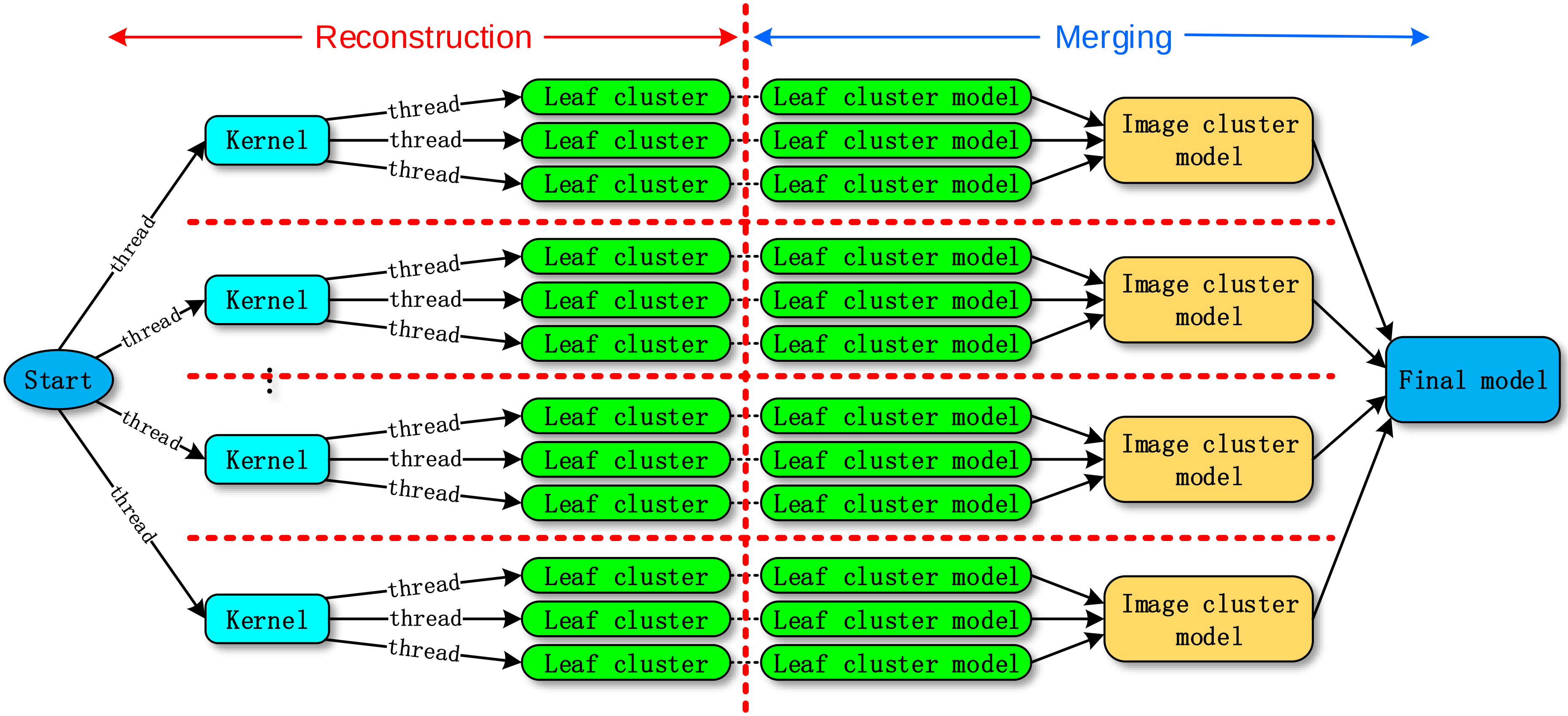
Size of the two clusters after merging them

Distance between two clusters

Distance difference from the two clusters to the kernel

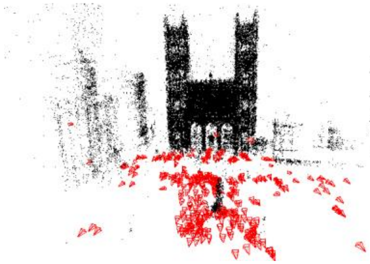
Key step 5: Parallel Reconstruction

Reconstruct kernels, leaf clusters and then merge them

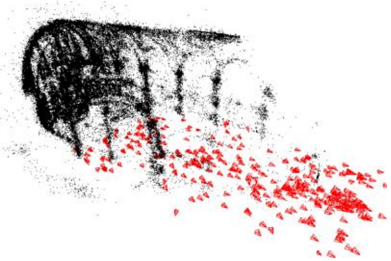


Results on Public Available Datasets

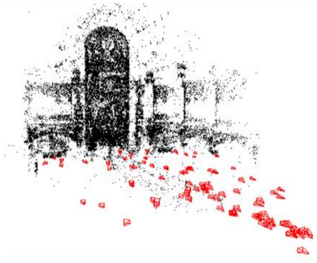
Results on three large scale Internet datasets ranging from 2K~6K



Montreal Notre Dame - 1



Montreal Notre Dame - 2

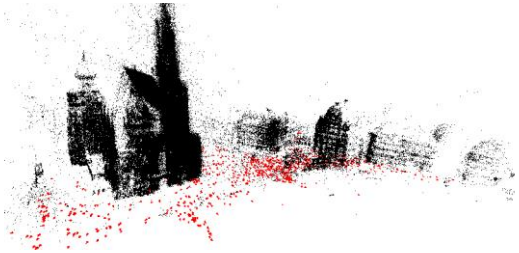


Montreal Notre Dame - 3

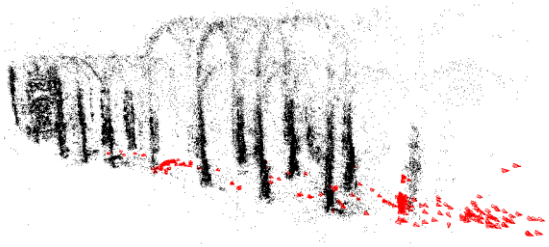
Dataset 1: Montreal Notre Dame

contains 2298 images

reconstructed 3 principle models



Vienna Cathedral - 1

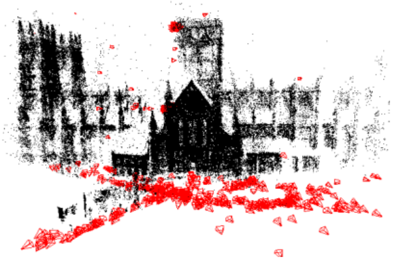


Vienna Cathedral - 2

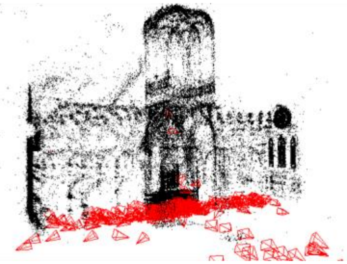
Dataset 2: Vienna Cathedral

contains 6288 images

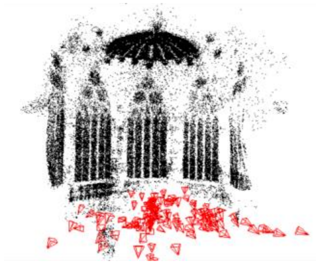
reconstructed 2 principle models



Yorkminster - 1



Yorkminster - 2



Yorkminster - 3

Dataset 3: Yorkminster

contains 3368 images

reconstructed 3 principle models

Results on Public Available Datasets

Results on three large scale Internet datasets ranging from 2K~6K

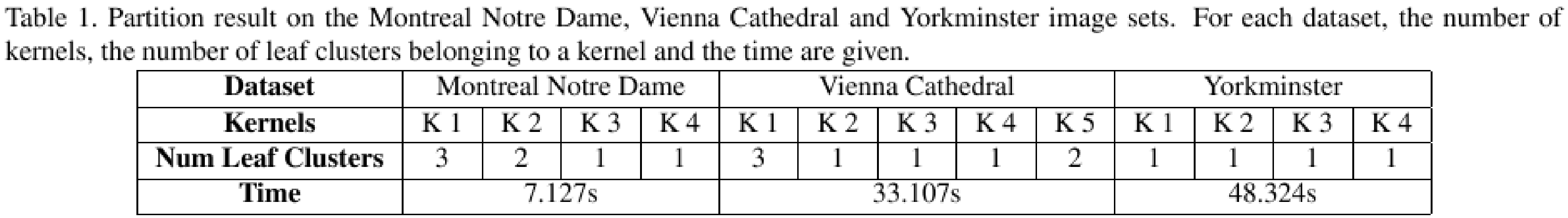


Table 1. Partition result on the Montreal Notre Dame, Vienna Cathedral and Yorkminster image sets. For each dataset, the number of kernels, the number of leaf clusters belonging to a kernel and the time are given.

Dataset	Montreal Notre Dame				Vienna Cathedral					Yorkminster			
Kernels	K 1	K 2	K 3	K 4	K 1	K 2	K 3	K 4	K 5	K 1	K 2	K 3	K 4
Num Leaf Clusters	3	2	1	1	3	1	1	1	2	1	1	1	1
Time	7.127s				33.107s					48.324s			

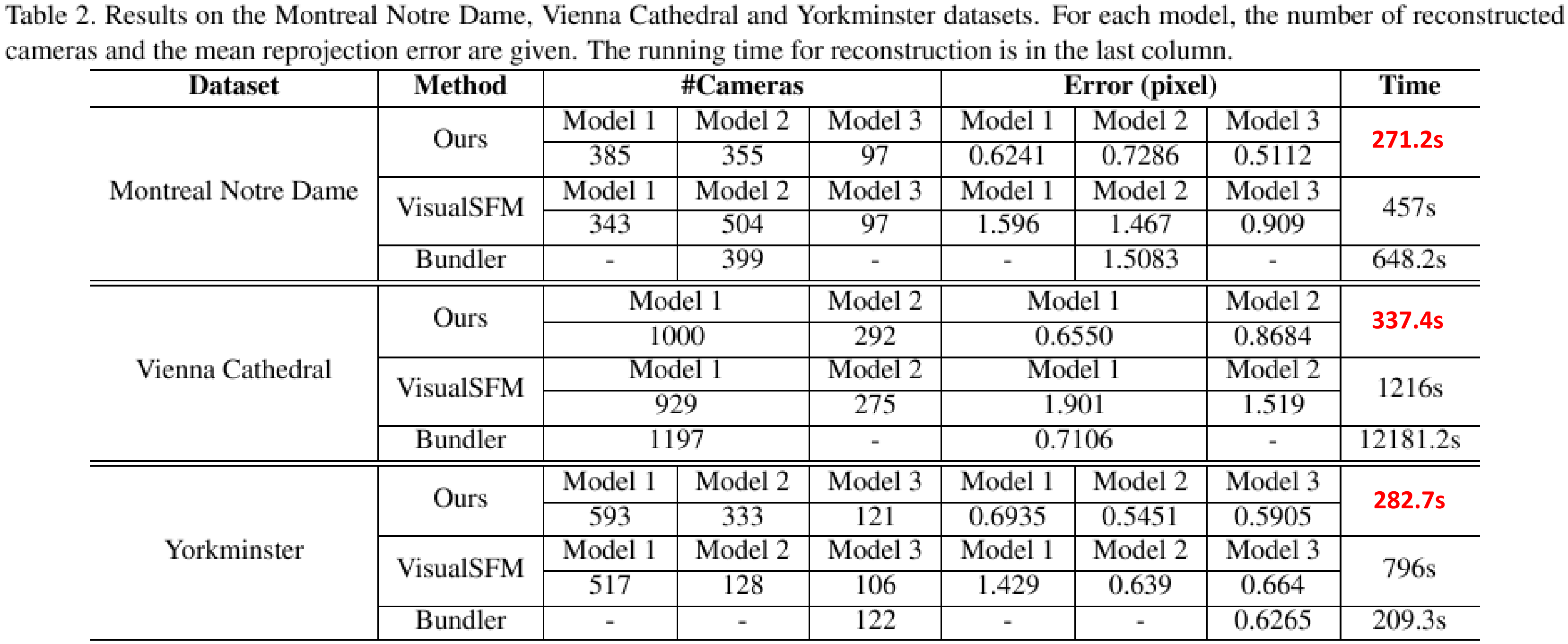


Table 2. Results on the Montreal Notre Dame, Vienna Cathedral and Yorkminster datasets. For each model, the number of reconstructed cameras and the mean reprojection error are given. The running time for reconstruction is in the last column.

Dataset	**Method**	**#Cameras**			**Error (pixel)**			**Time**
Montreal Notre Dame	Ours	Model 1	Model 2	Model 3	Model 1	Model 2	Model 3	**271.2s**
		385	355	97	0.6241	0.7286	0.5112	
	VisualSFM	Model 1	Model 2	Model 3	Model 1	Model 2	Model 3	457s
		343	504	97	1.596	1.467	0.909	
	Bundler	-	399	-	-	1.5083	-	648.2s
Vienna Cathedral	Ours	Model 1		Model 2	Model 1		Model 2	**337.4s**
		1000		292	0.6550		0.8684	
	VisualSFM	Model 1		Model 2	Model 1		Model 2	1216s
		929		275	1.901		1.519	
	Bundler	1197		-	0.7106		-	12181.2s
Yorkminster	Ours	Model 1	Model 2	Model 3	Model 1	Model 2	Model 3	**282.7s**
		593	333	121	0.6935	0.5451	0.5905	
	VisualSFM	Model 1	Model 2	Model 3	Model 1	Model 2	Model 3	796s
		517	128	106	1.429	0.639	0.664	
	Bundler	-	-	122	-	-	0.6265	209.3s

PART 4

Multi-View Stereo with Asymmetric Checkerboard Propagation

Introduction

- ***Multi-View Stereo:*** Given several calibrated images of the same object or scene, compute a dense representation of its 3D shape

  - ***Calibrated images:***

    Known camera parameters (robot arm, SfM)

    Arbitrary number of images

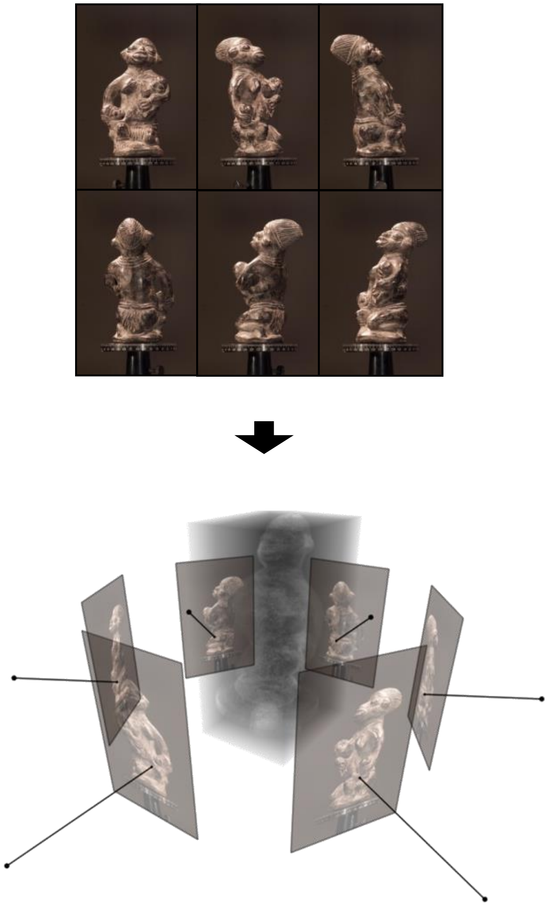
  - ***Dense representation:***

    Depth maps

    Point clouds

    Meshes

    Voxels



Related Works

□ Region Growing (PMVS[Furukawa2010])

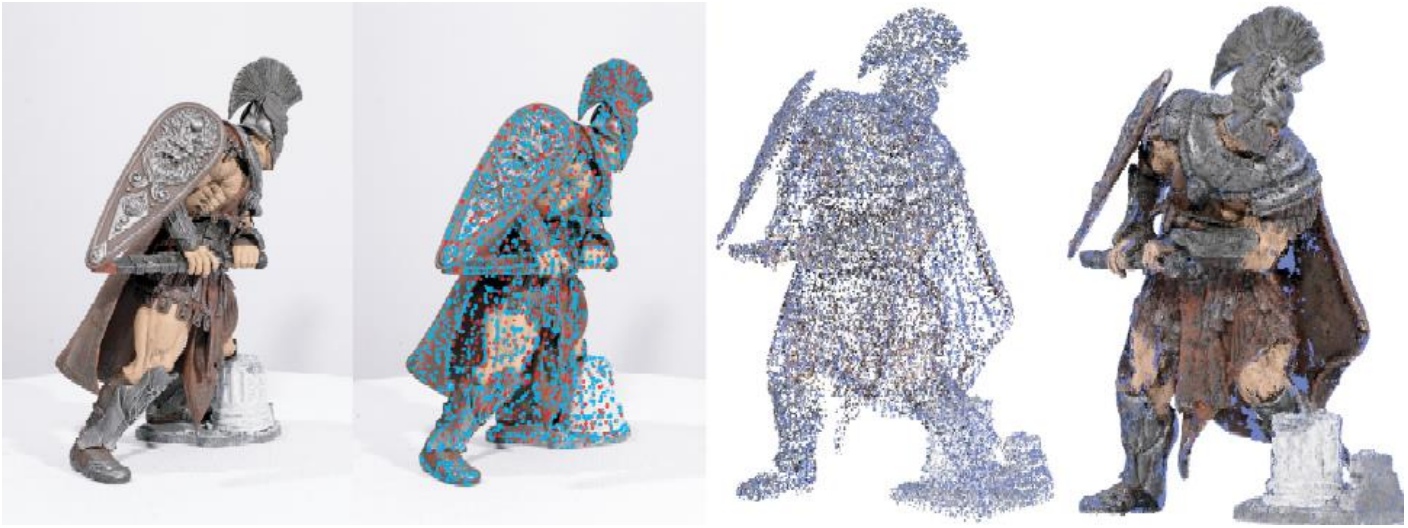
- Algorithm:

(1) Initial feature matching (2) Patch expansion (3) Patch filtering

- Drawback:

(1) Depend on initial feature matching

(2) Hard to execute parallel for irregular patch expansion

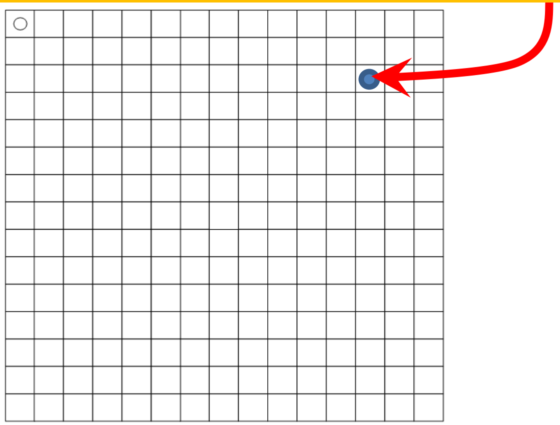


Input image #1 #2 #3

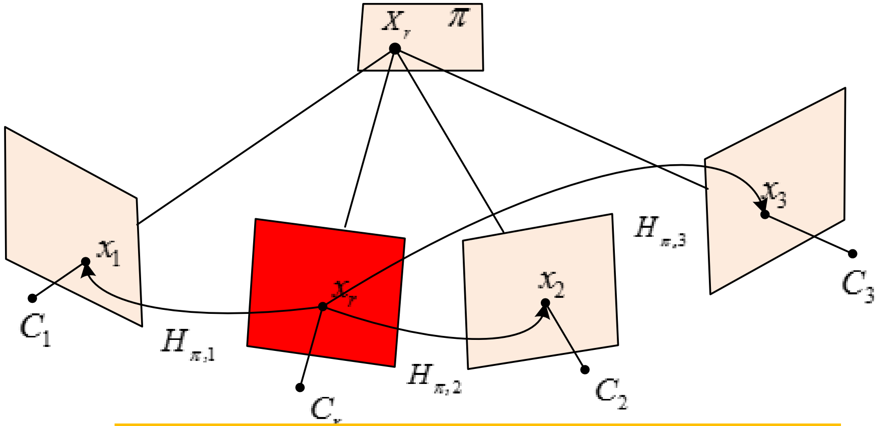
Related Works

□ PatchMatch Stereo（Gipuma[Galliani15], COLMAP[Schonberger16]）

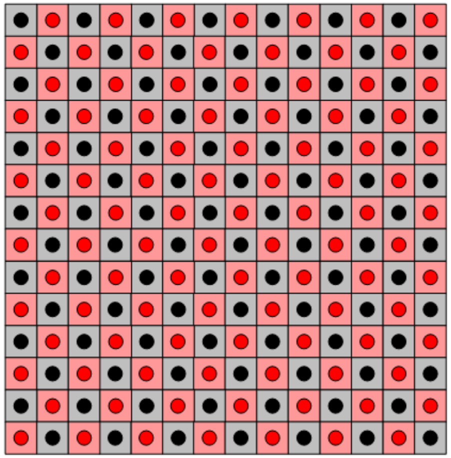
Random Hypothesis d, n^T
for each point



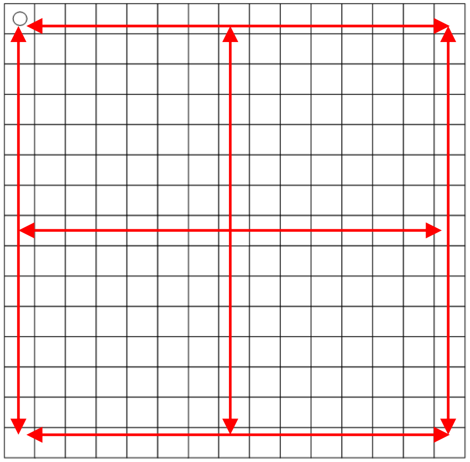
Multi-View homography
Choose the optimal hypothesis



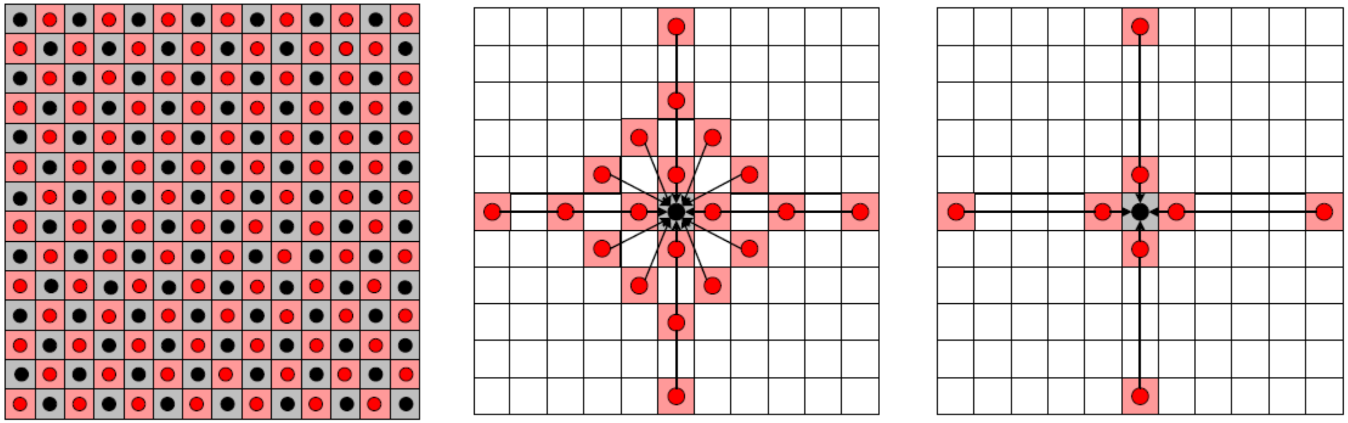
Gipuma: Checkerboard Pattern



COLMAP: Serial Propagation

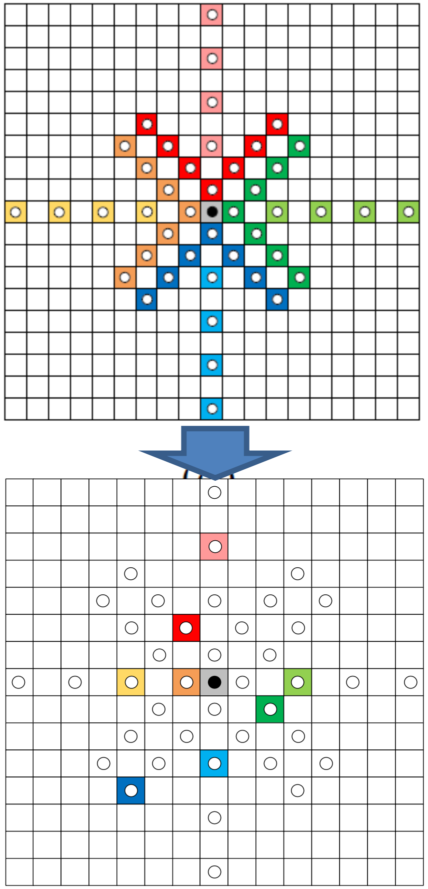


Asymmetric Checkerboard Propagation(AMHMVS)



(a) (b) (c)

Gipuma Symmetric Checkerboard Propagation



(d) Asymmetric

(a) The red-black checkerboard for updating the depth and normal of black pixels using the red pixels and vice versa.
(b) The standard checkerboard diffusion-like propagation.
(c) The fast checkerboard diffusion-like propagation.
(d) Our proposed asymmetric checkerboard.

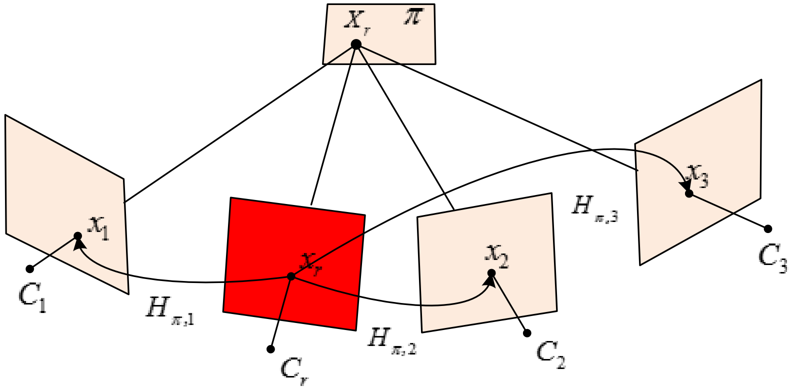
- Smooth region, hypothesis spread further
- Mutation region, hypothesis changes accordingly
- Hypothesis with high confidence spreads preferentially

Qingshan Xu, Wenbing Tao*, Multi-View Stereo with Asymmetric Checkerboard Propagation and Multi-Hypothesis Joint View Selection, arXiv:1805.07920

Multi-Hypothesis Joint View Selection

□ Parameterization for scene space

Hypothesis: normal n^T depth d



Multi-view homography correspondence

□ Cost Matrix

$$M = \begin{bmatrix} m_{11} & m_{12} & \mathrm{L} & m_{1N-1} \\ m_{21} & m_{22} & \mathrm{L} & m_{2N-1} \\ \mathrm{M} & \mathrm{M} & \mathrm{O} & \mathrm{M} \\ m_{81} & m_{82} & \mathrm{L} & m_{8N-1} \end{bmatrix}$$

More reliable hypothesis after our propagation scheme

□ Heuristic View Selection

$$\tau_{mc}(t) = \tau_{mc\_init} \cdot e^{-\frac{t^2}{\alpha}}, \psi(\chi^j) = \frac{1}{8}\sum_{i=1}^{8} C(m_{ij}) \leftarrow \boxed{C(m_{ij}) = e^{-\frac{m_{ij}^2}{2\beta^2}}}$$

$$m_{final}(i) = \frac{\sum \psi_{\mathrm{mod}}(\chi^z) \cdot m_{iZ}}{\sum \psi_{\mathrm{mod}}(\chi^z)}$$

$$\boxed{M = SVD}$$

Row： current optimal hypothesis selection

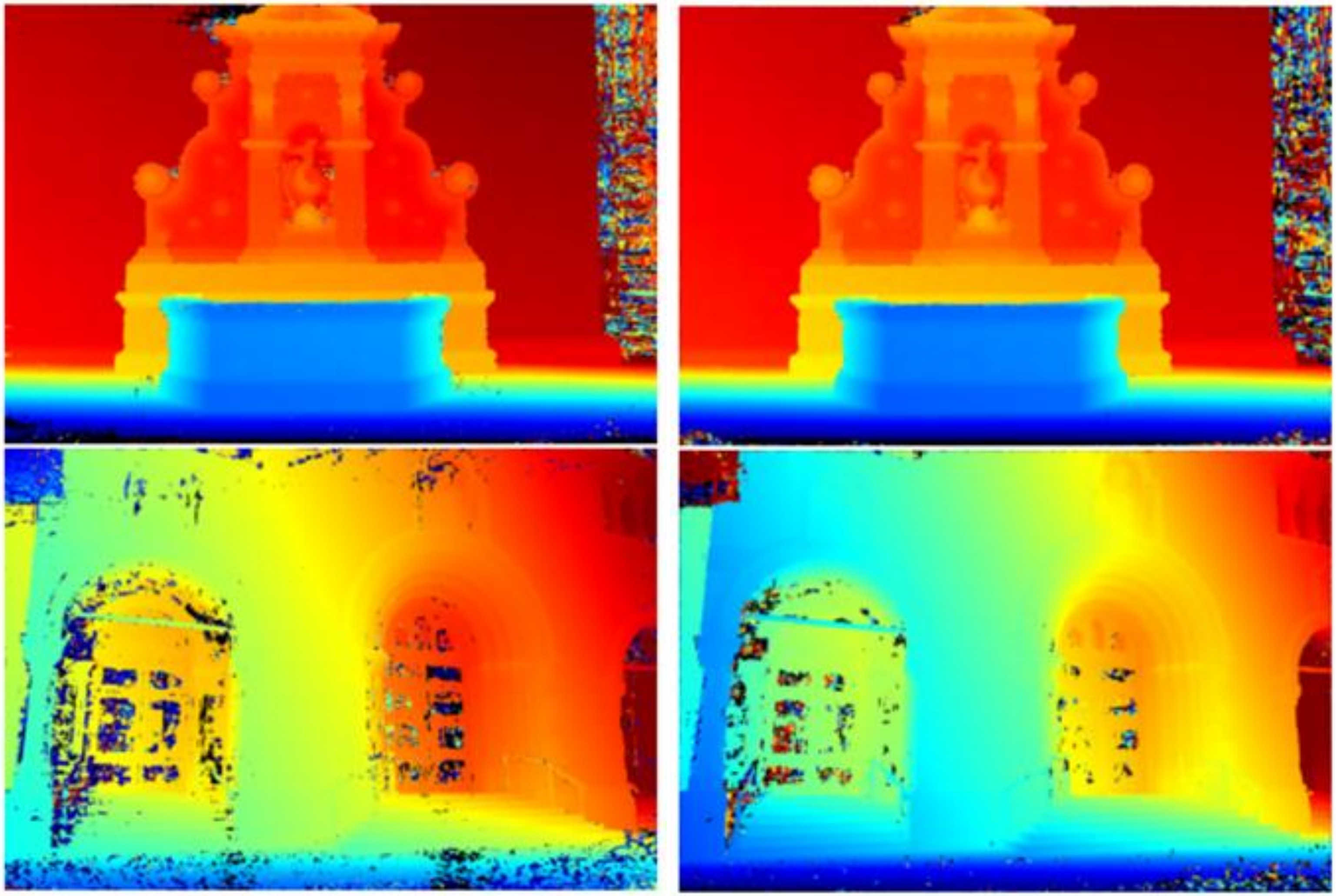
Column： aggregation view inference & weight integration

The largest singular value corresponds the most informed aggregation views

Qingshan Xu, Wenbing Tao*, Multi-View Stereo with Asymmetric Checkerboard Propagation and Multi-Hypothesis Joint View Selection, arXiv:1805.07920

Experiments

□ Strecha Dataset

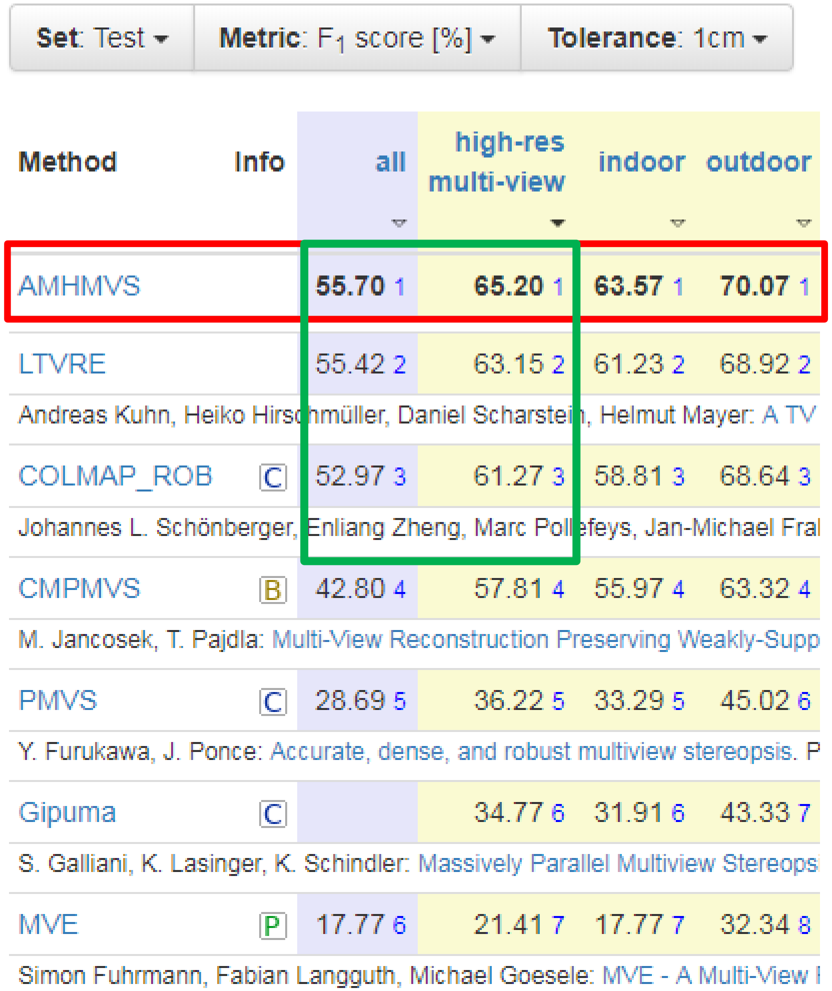


Gipuma

Ours

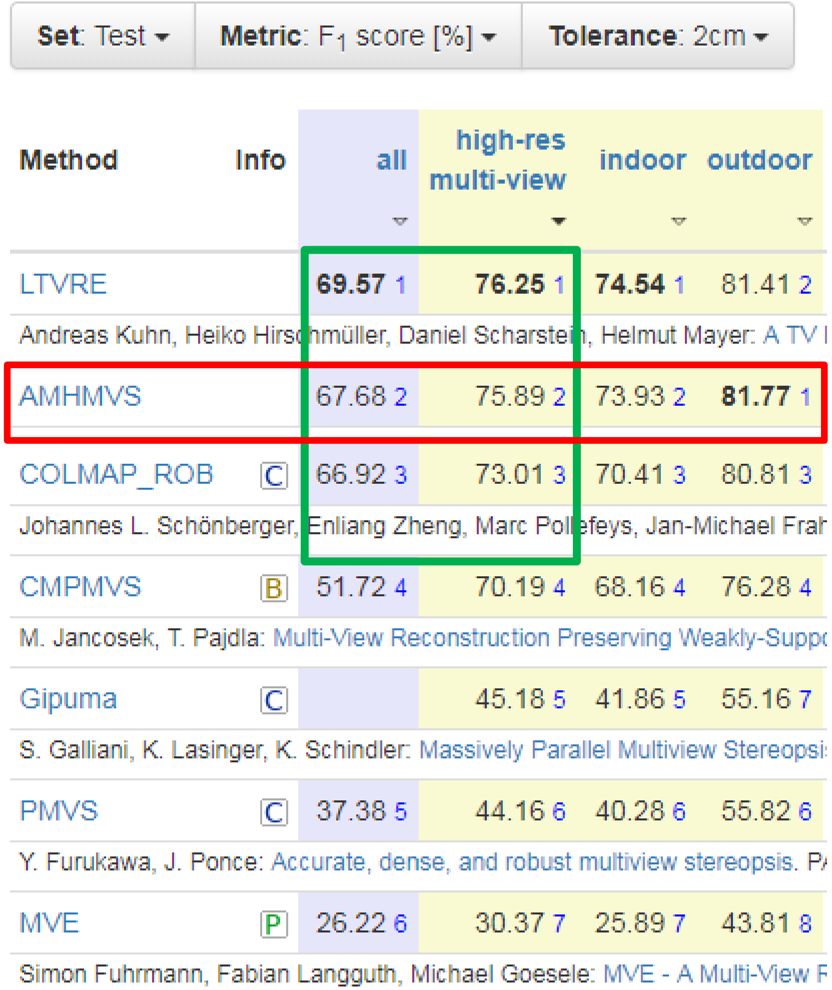
Experiments

ETH3D Benchmark (Schoeps et al., CVPR17, ETH Zurich)



Set: Test ▾ | **Metric**: F$_1$ score [%] ▾ | **Tolerance**: 1cm ▾

Method	Info	all	high-res multi-view	indoor	outdoor
AMHMVS		**55.70** 1	**65.20** 1	**63.57** 1	**70.07** 1
LTVRE		55.42 2	63.15 2	61.23 2	68.92 2
Andreas Kuhn, Heiko Hirschmüller, Daniel Scharstein, Helmut Mayer: A TV					
COLMAP_ROB	C	52.97 3	61.27 3	58.81 3	68.64 3
Johannes L. Schönberger, Enliang Zheng, Marc Pollefeys, Jan-Michael Fral					
CMPMVS	B	42.80 4	57.81 4	55.97 4	63.32 4
M. Jancosek, T. Pajdla: Multi-View Reconstruction Preserving Weakly-Supp					
PMVS	C	28.69 5	36.22 5	33.29 5	45.02 6
Y. Furukawa, J. Ponce: Accurate, dense, and robust multiview stereopsis. P					
Gipuma	C		34.77 6	31.91 6	43.33 7
S. Galliani, K. Lasinger, K. Schindler: Massively Parallel Multiview Stereops					
MVE	P	17.77 6	21.41 7	17.77 7	32.34 8
Simon Fuhrmann, Fabian Langguth, Michael Goesele: MVE - A Multi-View I					



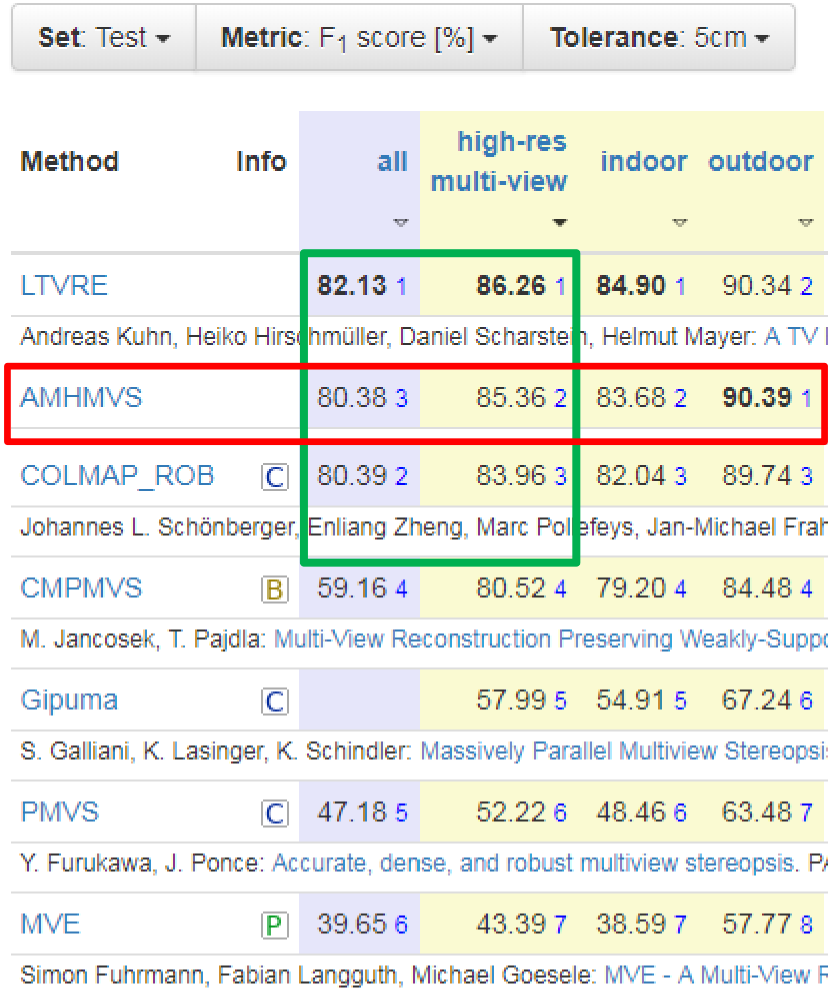
Set: Test ▾ | **Metric**: F$_1$ score [%] ▾ | **Tolerance**: 2cm ▾

Method	Info	all	high-res multi-view	indoor	outdoor
LTVRE		**69.57** 1	**76.25** 1	**74.54** 1	81.41 2
Andreas Kuhn, Heiko Hirschmüller, Daniel Scharstein, Helmut Mayer: A TV I					
AMHMVS		67.68 2	75.89 2	73.93 2	**81.77** 1
COLMAP_ROB	C	66.92 3	73.01 3	70.41 3	80.81 3
Johannes L. Schönberger, Enliang Zheng, Marc Pollefeys, Jan-Michael Frah					
CMPMVS	B	51.72 4	70.19 4	68.16 4	76.28 4
M. Jancosek, T. Pajdla: Multi-View Reconstruction Preserving Weakly-Suppo					
Gipuma	C		45.18 5	41.86 5	55.16 7
S. Galliani, K. Lasinger, K. Schindler: Massively Parallel Multiview Stereopsi					
PMVS	C	37.38 5	44.16 6	40.28 6	55.82 6
Y. Furukawa, J. Ponce: Accurate, dense, and robust multiview stereopsis. P/					
MVE	P	26.22 6	30.37 7	25.89 7	43.81 8
Simon Fuhrmann, Fabian Langguth, Michael Goesele: MVE - A Multi-View F					

T. Schoeps, J. Schoenberger, S. Galliani, T. Sattler, K. Schindler, A. Geiger, M. Pollefeys, A Multi-View Stereo Benchmark with High-Resolution Images and Multi-Camera Videos in Unstructured Scenes, CVPR 2017

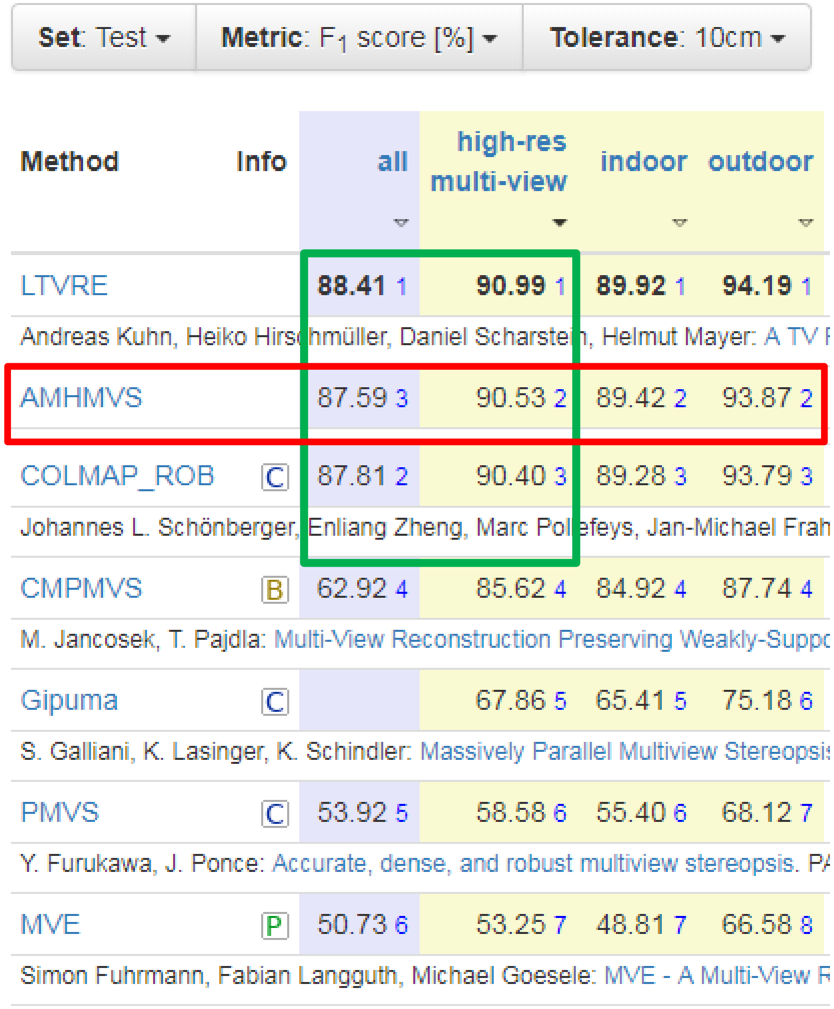
Experiments

□ ETH3D Benchmark (Schoeps et al., CVPR17, ETH Zurich)



Set: Test ▾ **Metric**: F_1 score [%] ▾ **Tolerance**: 5cm ▾

Method	Info	all	high-res multi-view	indoor	outdoor
LTVRE		**82.13** 1	**86.26** 1	**84.90** 1	90.34 2
Andreas Kuhn, Heiko Hirschmüller, Daniel Scharstein, Helmut Mayer: A TV I					
AMHMVS		80.38 3	85.36 2	83.68 2	**90.39** 1
COLMAP_ROB	C	80.39 2	83.96 3	82.04 3	89.74 3
Johannes L. Schönberger, Enliang Zheng, Marc Pollefeys, Jan-Michael Frah					
CMPMVS	B	59.16 4	80.52 4	79.20 4	84.48 4
M. Jancosek, T. Pajdla: Multi-View Reconstruction Preserving Weakly-Suppo					
Gipuma	C		57.99 5	54.91 5	67.24 6
S. Galliani, K. Lasinger, K. Schindler: Massively Parallel Multiview Stereopsi					
PMVS	C	47.18 5	52.22 6	48.46 6	63.48 7
Y. Furukawa, J. Ponce: Accurate, dense, and robust multiview stereopsis. PA					
MVE	P	39.65 6	43.39 7	38.59 7	57.77 8
Simon Fuhrmann, Fabian Langguth, Michael Goesele: MVE - A Multi-View F					



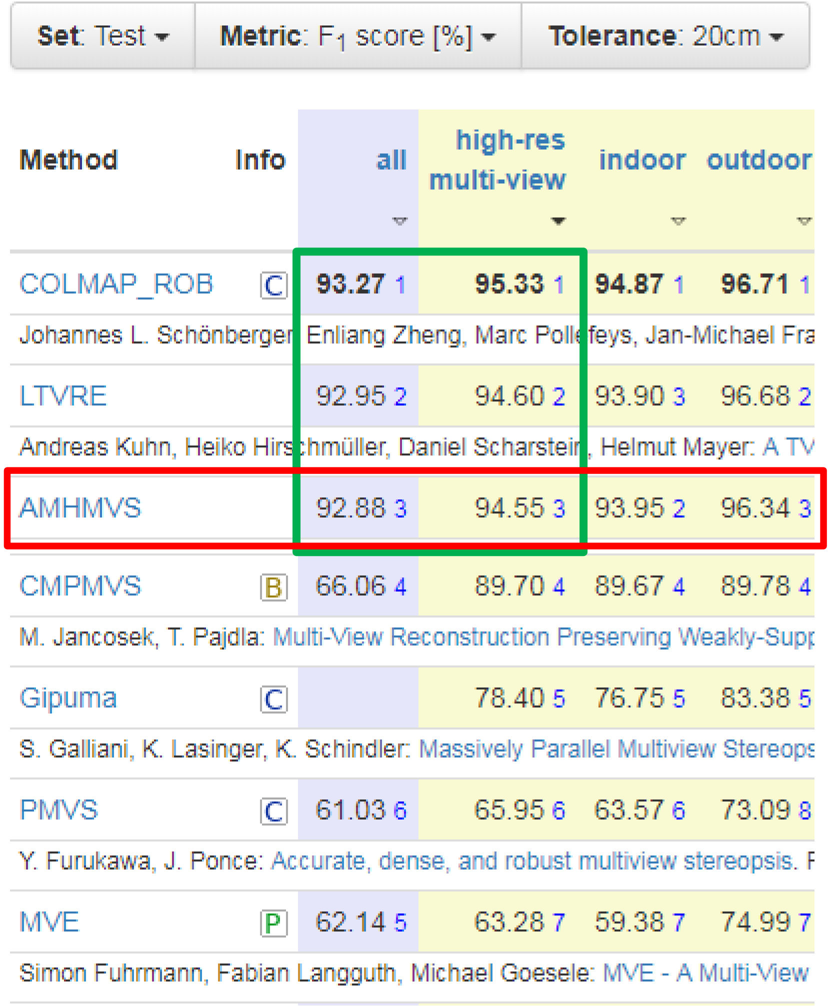
Set: Test ▾ **Metric**: F_1 score [%] ▾ **Tolerance**: 10cm ▾

Method	Info	all	high-res multi-view	indoor	outdoor
LTVRE		**88.41** 1	**90.99** 1	**89.92** 1	**94.19** 1
Andreas Kuhn, Heiko Hirschmüller, Daniel Scharstein, Helmut Mayer: A TV F					
AMHMVS		87.59 3	90.53 2	89.42 2	93.87 2
COLMAP_ROB	C	87.81 2	90.40 3	89.28 3	93.79 3
Johannes L. Schönberger, Enliang Zheng, Marc Pollefeys, Jan-Michael Frah					
CMPMVS	B	62.92 4	85.62 4	84.92 4	87.74 4
M. Jancosek, T. Pajdla: Multi-View Reconstruction Preserving Weakly-Suppo					
Gipuma	C		67.86 5	65.41 5	75.18 6
S. Galliani, K. Lasinger, K. Schindler: Massively Parallel Multiview Stereopsis					
PMVS	C	53.92 5	58.58 6	55.40 6	68.12 7
Y. Furukawa, J. Ponce: Accurate, dense, and robust multiview stereopsis. PA					
MVE	P	50.73 6	53.25 7	48.81 7	66.58 8
Simon Fuhrmann, Fabian Langguth, Michael Goesele: MVE - A Multi-View R					

T. Schoeps, J. Schoenberger, S. Galliani, T. Sattler, K. Schindler, A. Geiger, M. Pollefeys, A Multi-View Stereo Benchmark with High-Resolution Images and Multi-Camera Videos in Unstructured Scenes, CVPR 2017

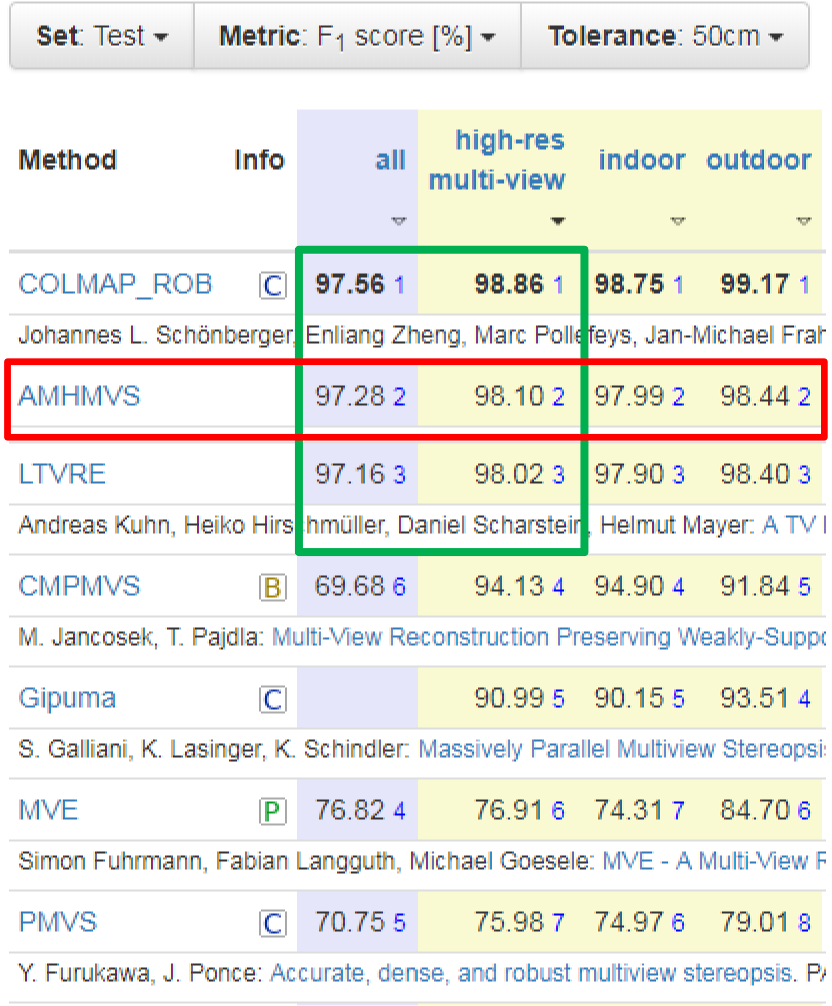
Experiments

□ ETH3D Benchmark (Schoeps et al., CVPR17, ETH Zurich)



Set: Test ▾ | Metric: F_1 score [%] ▾ | Tolerance: 20cm ▾

Method	Info	all	high-res multi-view	indoor	outdoor
COLMAP_ROB	C	**93.27** 1	**95.33** 1	**94.87** 1	**96.71** 1
Johannes L. Schönberger, Enliang Zheng, Marc Pollefeys, Jan-Michael Fra					
LTVRE		92.95 2	94.60 2	93.90 3	96.68 2
Andreas Kuhn, Heiko Hirschmüller, Daniel Scharstein, Helmut Mayer: A TV					
AMHMVS		92.88 3	94.55 3	93.95 2	96.34 3
CMPMVS	B	66.06 4	89.70 4	89.67 4	89.78 4
M. Jancosek, T. Pajdla: Multi-View Reconstruction Preserving Weakly-Supp					
Gipuma	C		78.40 5	76.75 5	83.38 5
S. Galliani, K. Lasinger, K. Schindler: Massively Parallel Multiview Stereops					
PMVS	C	61.03 6	65.95 6	63.57 6	73.09 8
Y. Furukawa, J. Ponce: Accurate, dense, and robust multiview stereopsis. F					
MVE	P	62.14 5	63.28 7	59.38 7	74.99 7
Simon Fuhrmann, Fabian Langguth, Michael Goesele: MVE - A Multi-View					



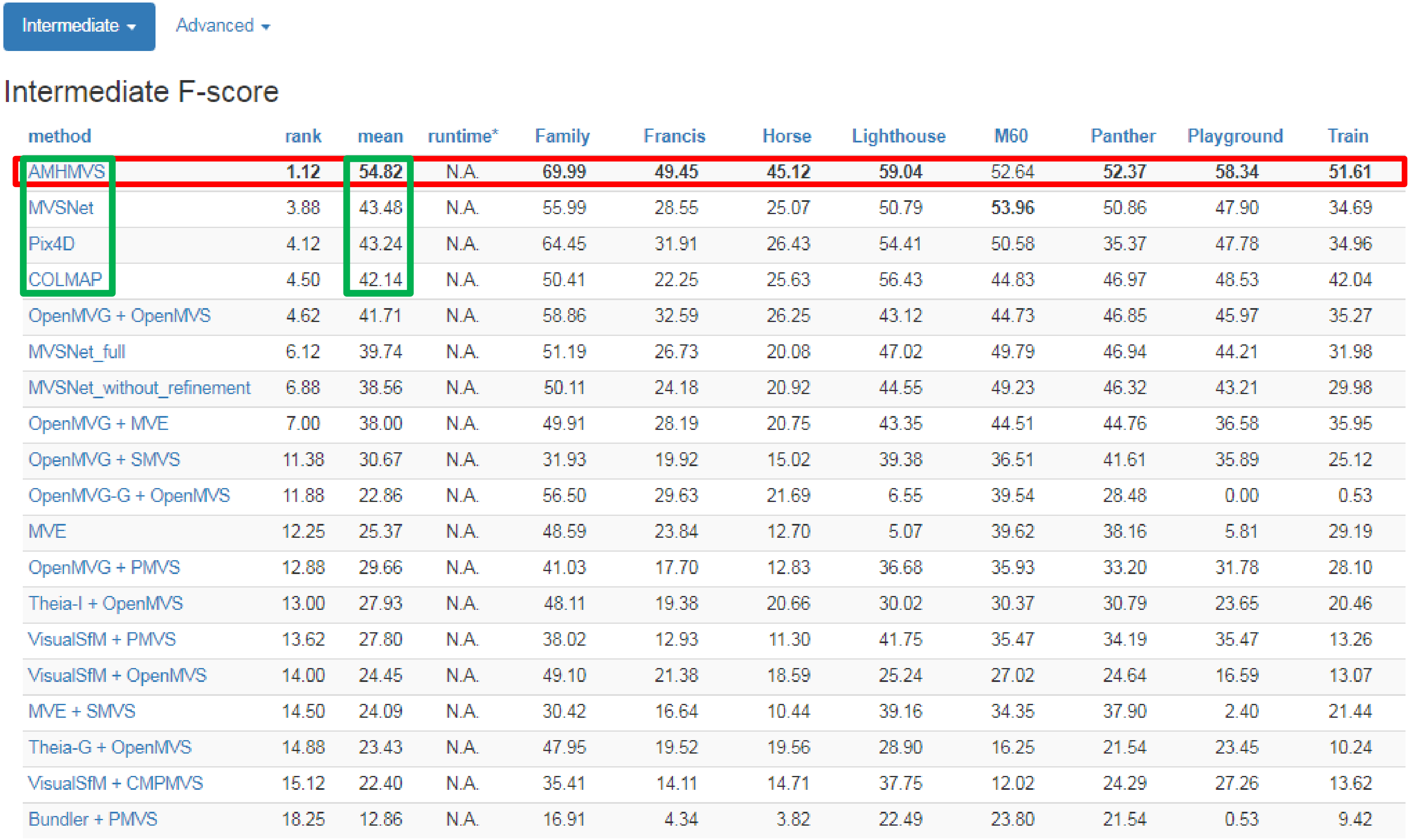
Set: Test ▾ | Metric: F_1 score [%] ▾ | Tolerance: 50cm ▾

Method	Info	all	high-res multi-view	indoor	outdoor
COLMAP_ROB	C	**97.56** 1	**98.86** 1	**98.75** 1	**99.17** 1
Johannes L. Schönberger, Enliang Zheng, Marc Pollefeys, Jan-Michael Frah					
AMHMVS		97.28 2	98.10 2	97.99 2	98.44 2
LTVRE		97.16 3	98.02 3	97.90 3	98.40 3
Andreas Kuhn, Heiko Hirschmüller, Daniel Scharstein, Helmut Mayer: A TV I					
CMPMVS	B	69.68 6	94.13 4	94.90 4	91.84 5
M. Jancosek, T. Pajdla: Multi-View Reconstruction Preserving Weakly-Suppo					
Gipuma	C		90.99 5	90.15 5	93.51 4
S. Galliani, K. Lasinger, K. Schindler: Massively Parallel Multiview Stereopsi					
MVE	P	76.82 4	76.91 6	74.31 7	84.70 6
Simon Fuhrmann, Fabian Langguth, Michael Goesele: MVE - A Multi-View F					
PMVS	C	70.75 5	75.98 7	74.97 6	79.01 8
Y. Furukawa, J. Ponce: Accurate, dense, and robust multiview stereopsis. PA					

T. Schoeps, J. Schoenberger, S. Galliani, T. Sattler, K. Schindler, A. Geiger, M. Pollefeys, A Multi-View Stereo Benchmark with High-Resolution Images and Multi-Camera Videos in Unstructured Scenes, CVPR 2017

Experiments

□ Tanks and Temples Dataset (Knapitsch, et al., SIGGRAPH2017, Intel)



Intermediate F-score

method	rank	mean	runtime*	Family	Francis	Horse	Lighthouse	M60	Panther	Playground	Train
AMHMVS	**1.12**	**54.82**	N.A.	**69.99**	**49.45**	**45.12**	**59.04**	52.64	**52.37**	**58.34**	**51.61**
MVSNet	3.88	43.48	N.A.	55.99	28.55	25.07	50.79	**53.96**	50.86	47.90	34.69
Pix4D	4.12	43.24	N.A.	64.45	31.91	26.43	54.41	50.58	35.37	47.78	34.96
COLMAP	4.50	42.14	N.A.	50.41	22.25	25.63	56.43	44.83	46.97	48.53	42.04
OpenMVG + OpenMVS	4.62	41.71	N.A.	58.86	32.59	26.25	43.12	44.73	46.85	45.97	35.27
MVSNet_full	6.12	39.74	N.A.	51.19	26.73	20.08	47.02	49.79	46.94	44.21	31.98
MVSNet_without_refinement	6.88	38.56	N.A.	50.11	24.18	20.92	44.55	49.23	46.32	43.21	29.98
OpenMVG + MVE	7.00	38.00	N.A.	49.91	28.19	20.75	43.35	44.51	44.76	36.58	35.95
OpenMVG + SMVS	11.38	30.67	N.A.	31.93	19.92	15.02	39.38	36.51	41.61	35.89	25.12
OpenMVG-G + OpenMVS	11.88	22.86	N.A.	56.50	29.63	21.69	6.55	39.54	28.48	0.00	0.53
MVE	12.25	25.37	N.A.	48.59	23.84	12.70	5.07	39.62	38.16	5.81	29.19
OpenMVG + PMVS	12.88	29.66	N.A.	41.03	17.70	12.83	36.68	35.93	33.20	31.78	28.10
Theia-I + OpenMVS	13.00	27.93	N.A.	48.11	19.38	20.66	30.02	30.37	30.79	23.65	20.46
VisualSfM + PMVS	13.62	27.80	N.A.	38.02	12.93	11.30	41.75	35.47	34.19	35.47	13.26
VisualSfM + OpenMVS	14.00	24.45	N.A.	49.10	21.38	18.59	25.24	27.02	24.64	16.59	13.07
MVE + SMVS	14.50	24.09	N.A.	30.42	16.64	10.44	39.16	34.35	37.90	2.40	21.44
Theia-G + OpenMVS	14.88	23.43	N.A.	47.95	19.52	19.56	28.90	16.25	21.54	23.45	10.24
VisualSfM + CMPMVS	15.12	22.40	N.A.	35.41	14.11	14.71	37.75	12.02	24.29	27.26	13.62
Bundler + PMVS	18.25	12.86	N.A.	16.91	4.34	3.82	22.49	23.80	21.54	0.53	9.42

Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun , Tanks and Temples: Benchmarking Large-Scale Scene Reconstruction, SIGGRAPH 2017

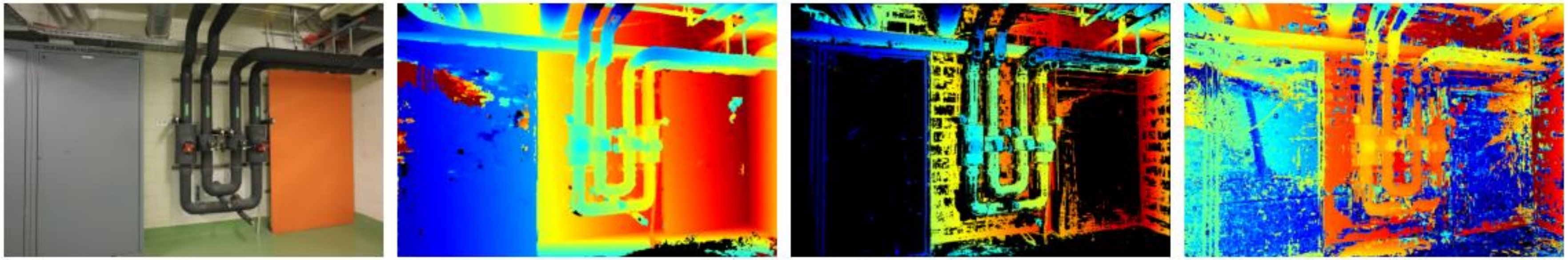
Futhermore

□ Our new method (PGC)

Evaluation on ETH3D training dataset:

Tolerance	Method	high-res multi-view	indoor	outdoor
1cm	AMHMVS	58.24	59.56	56.70
	PGC	64.12	64.69	63.45
2cm	AMHMVS	70.71	70.00	71.54
	PGC	75.82	74.30	77.58

Depth maps:



(a) Original image [15]

(b) PGC

(c) COLMAP [14]

(d) AMHMVS [21]

Thank you!