3D Line Reconstruction

ENGN2560 Final Presentation
Instructor: Prof. Benjamin Kimia
Student: Justin Miller

Primary Paper

Liu, Shaohui, Yifan Yu, Rémi Pautrat, Marc Pollefeys, and Viktor Larsson. "3d line mapping revisited." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21445-21455. 2023.

Secondary Paper

Bai, Xulong, Hainan Cui, and Shuhan Shen. "Consistent 3D Line Mapping." In *European Conference on Computer Vision*, pp. 57-74. Cham: Springer Nature Switzerland, 2024.

Presentation Outline

- Initial Presentation Recap & Refinement ***(5 minutes)***
    - Introduction/Problem/Related Work
    - Proposed Approaches and Reported Results

- Codebase, Implementation, and Replicated Paper Results ***(10 minutes)***
    - Brief Code Review & Implementation
    - Replicating Results of Original Paper

- Strengths, Issues, and Proposed Use Cases ***(10 minutes)***
    - Breaking down paper results - individual scenes and Ground Truth comparisons
    - Extending to Organic Structures - Problems with *too nonlinear* data
    - Extending to ABC Dataset - Problems with *too smooth* data

Initial Presentation Recap & Refinement

Section 1: ***<u>(5 minutes)</u>***

- Introduction/Problem/Related Work
- Proposed Approaches and Reported Results

Introduction

- Estimating 3D geometry of scenes from image data is a crucial problem in the field of computer vision
- This approach has many relevant applications in robotics, manufacturing and autonomous vehicles.

- Strong body of research, particularly in point-based methods
- But, point based reconstruction methods exhibit weaknesses in scenes with insufficient feature diversity, such as urban areas

Problem Statement

- This paper aims to address these weaknesses by using lines
- Research in line-based reconstruction lags behind the progress made in point-based approaches

- The primary paper introduces LIMAP to address this
- The secondary paper introduces CLMAP to improve upon LIMAP

Related Approaches

3D Line Reconstruction:

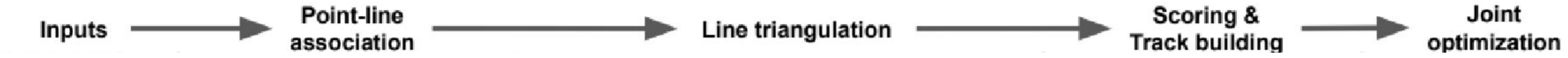
- L3D++: Efficient 3D Scene Abstraction Using Line Segments
  - Both papers cite & build upon this method from 2016
- ELSR: Efficient line segment reconstruction
  - LIMAP cites this paper from 2022 and compares results

Many other approaches in 3D Reconstruction using points, edges, etc… .

Hofer, M., Maurer, M., and Bischof, H.: Efficient 3D Scene Abstraction Using Line Segments, In Computer Vision and Image Understanding (CVIU), 2016.
Dong Wei, Yi Wan, Yongjun Zhang, Xinyi Liu, Bin Zhang, and Xiqi Wang. Elsr: Efficient line segment reconstruction with planes and points guidance. In CVPR, 2022

Proposed Approach - LIMAP

- Improvements over L3D++ involve exploiting prior knowledge about structure
- Specifically, included 3 different triangulation methods to account for degeneracy which occurs when lines are parallel with the epipolar lines; as well as introducing novel scoring components to encode line segment properties
- The pipeline below assumes line detections, point-based SfM have been run



LIMAP Pipeline

Liu, Shaohui, Yifan Yu, Rémi Pautrat, Marc Pollefeys, and Viktor Larsson. "3d line mapping revisited." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21445-21455. 2023.

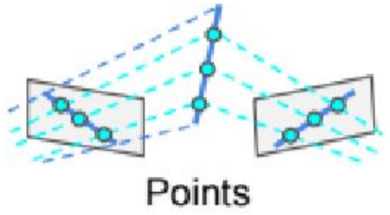
Triangulation Approach - LIMAP

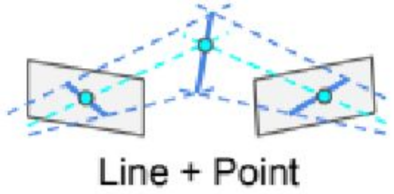
- To address degeneracy, the paper provides 3 methods:

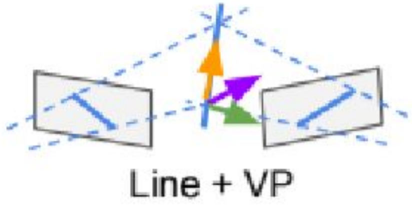
- **M1: Multiple 3D Points**
  - Supports fully degenerate endpoints; Requires 3D points on line
  - Uses these supplementary points to find the best fit line

- **M2: 2D Line + 1 3D Point**
  - Supports weakly degenerate endpoints; Requires 1 3D point
  - Uses extra point to anchor the line and fit

- **M3: 2D Line + Vanishing Point**
  - Supports weakly degenerate endpoints; Requires a vanishing point (VP) which can be sourced from a reference image or match
  - Uses the direction of the VP to to anchor the line and fit







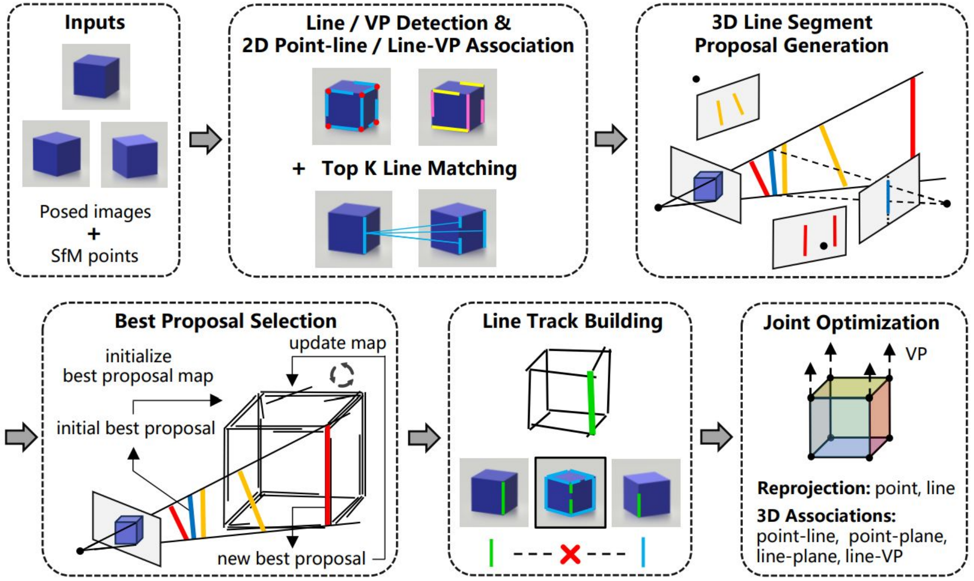
Liu, Shaohui, Yifan Yu, Rémi Pautrat, Marc Pollefeys, and Viktor Larsson. "3d line mapping revisited." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21445-21455. 2023.

Proposed Additions - CLMAP

- Released after LIMAP

- Main improvement was a stronger focus on maintaining consistency in line structure

- LIMAP did not ensure that the best-scored proposals aligned, CLMAP tries to fix this
    - Adds iterative 3D line fitting
    - Improves scoring to consider line coplanarity



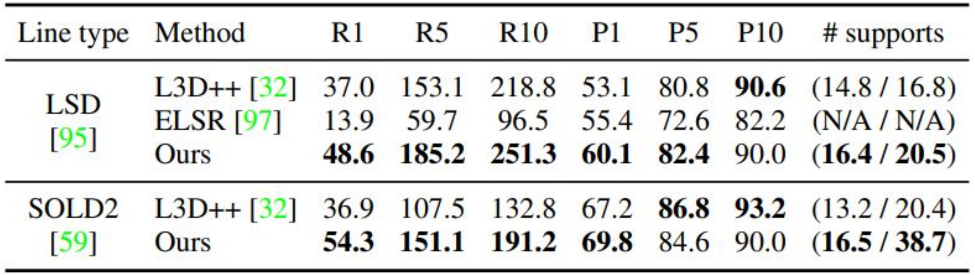
Bai, Xulong, Hainan Cui, and Shuhan Shen. "Consistent 3D Line Mapping." In *European Conference on Computer Vision*, pp. 57-74. Cham: Springer Nature Switzerland, 2024.

Reported Results

- Using the respective LIMAP & CLMAP libraries, both papers conducted large scale accuracy tests on the same two labeled datasets:
  - Dataset 1: Hypersim
    - 461 synthetic but realistic complex, indoor scenes
  - Dataset 2: Tanks and Temples
    - 8 Scenes, modeled and scanned

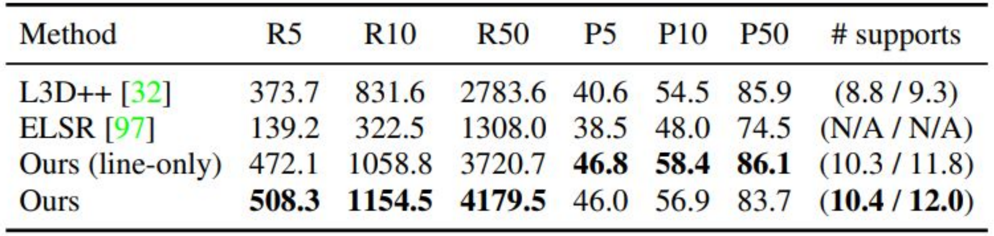
- Accuracy Metrics were also consistent
  - R_T: Sum of line lengths within T mm of ground truth scans
  - P_T: Percentage of tracks within T mm of ground truth scans
  - # Supports: Average number of image supports for all images & tracks

Reported Results - LIMAP



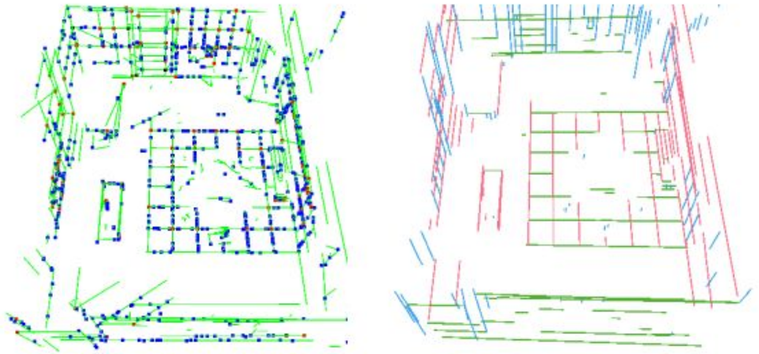
Line type	Method	R1	R5	R10	P1	P5	P10	# supports
LSD [95]	L3D++ [32]	37.0	153.1	218.8	53.1	80.8	**90.6**	(14.8 / 16.8)
	ELSR [97]	13.9	59.7	96.5	55.4	72.6	82.2	(N/A / N/A)
	Ours	**48.6**	**185.2**	**251.3**	**60.1**	**82.4**	90.0	(**16.4** / **20.5**)
SOLD2 [59]	L3D++ [32]	36.9	107.5	132.8	67.2	**86.8**	**93.2**	(13.2 / 20.4)
	Ours	**54.3**	**151.1**	**191.2**	**69.8**	84.6	90.0	(**16.5** / **38.7**)

Hypersim



Method	R5	R10	R50	P5	P10	P50	# supports
L3D++ [32]	373.7	831.6	2783.6	40.6	54.5	85.9	(8.8 / 9.3)
ELSR [97]	139.2	322.5	1308.0	38.5	48.0	74.5	(N/A / N/A)
Ours (line-only)	472.1	1058.8	3720.7	**46.8**	**58.4**	**86.1**	(10.3 / 11.8)
Ours	**508.3**	**1154.5**	**4179.5**	46.0	56.9	83.7	(**10.4** / **12.0**)

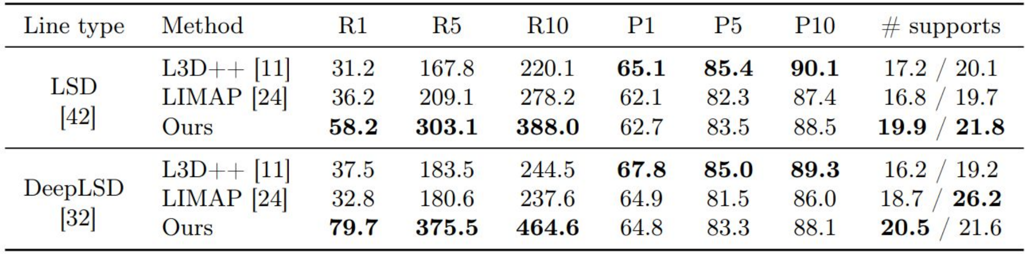
Tanks and Temples



Sample Result: Barn Interior

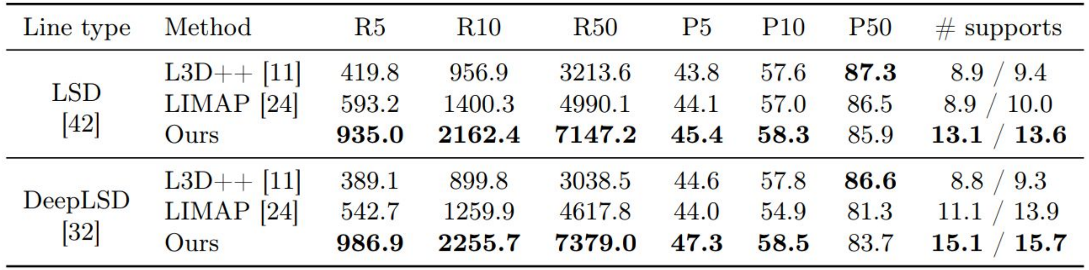
Liu, Shaohui, Yifan Yu, Rémi Pautrat, Marc Pollefeys, and Viktor Larsson. "3d line mapping revisited." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21445-21455. 2023.

Reported Results - CLMAP



Line type	Method	R1	R5	R10	P1	P5	P10	# supports
LSD [42]	L3D++ [11]	31.2	167.8	220.1	**65.1**	**85.4**	**90.1**	17.2 / 20.1
	LIMAP [24]	36.2	209.1	278.2	62.1	82.3	87.4	16.8 / 19.7
	Ours	**58.2**	**303.1**	**388.0**	62.7	83.5	88.5	**19.9** / **21.8**
DeepLSD [32]	L3D++ [11]	37.5	183.5	244.5	**67.8**	**85.0**	**89.3**	16.2 / 19.2
	LIMAP [24]	32.8	180.6	237.6	64.9	81.5	86.0	18.7 / **26.2**
	Ours	**79.7**	**375.5**	**464.6**	64.8	83.3	88.1	**20.5** / 21.6

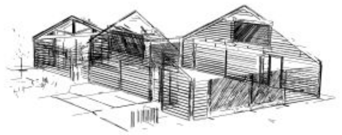
Hypersim

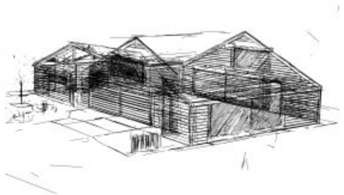


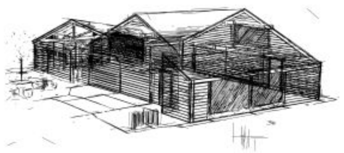
Line type	Method	R5	R10	R50	P5	P10	P50	# supports
LSD [42]	L3D++ [11]	419.8	956.9	3213.6	43.8	57.6	**87.3**	8.9 / 9.4
	LIMAP [24]	593.2	1400.3	4990.1	44.1	57.0	86.5	8.9 / 10.0
	Ours	**935.0**	**2162.4**	**7147.2**	**45.4**	**58.3**	85.9	**13.1** / **13.6**
DeepLSD [32]	L3D++ [11]	389.1	899.8	3038.5	44.6	57.8	**86.6**	8.8 / 9.3
	LIMAP [24]	542.7	1259.9	4617.8	44.0	54.9	81.3	11.1 / 13.9
	Ours	**986.9**	**2255.7**	**7379.0**	**47.3**	**58.5**	83.7	**15.1** / **15.7**

Tanks and Temples

Barn


L3D++


LIMAP


CLMAP

Bai, Xulong, Hainan Cui, and Shuhan Shen. "Consistent 3D Line Mapping." In *European Conference on Computer Vision*, pp. 57-74. Cham: Springer Nature Switzerland, 2024.

Codebase, Implementation, and Replicated Paper Results

Section 2: ***<u>(10 minutes)</u>***

- Code Overview & Implementation
  - Including Code Issues
- Replicating Results of Original Paper

Implementation - Overview

- Both code repositories are publicly available (LIMAP, CLMAP)

- Both papers are implemented in C++ with Python bindings
  - This is intended to optimize for speed while still having usable interfacing for running different types of simulations on different datasets
- Both code bases are only supported on Ubuntu

- LIMAP has dependencies on COLMAP, LSD, and other libraries
- CLMAP has dependencies on LIMAP and its libraries

Implementation - Building Packages

- For the OS & C++ components, the following packages were installed / setup
    - Windows Subsystem for Linux (WSL)
        - Various system packages (list provided in repo)
    - CMAKE > 3.17
    - CUDA (for GPU computation, current hardware is RTX4070)

- On the Python side of things, conda environments were used to set up dependencies (and keep LIMAP separate from CLMAP)
    - Python 3.9
    - Editable install of the LIMAP repository

Implementation - LIMAP vs. CLMAP

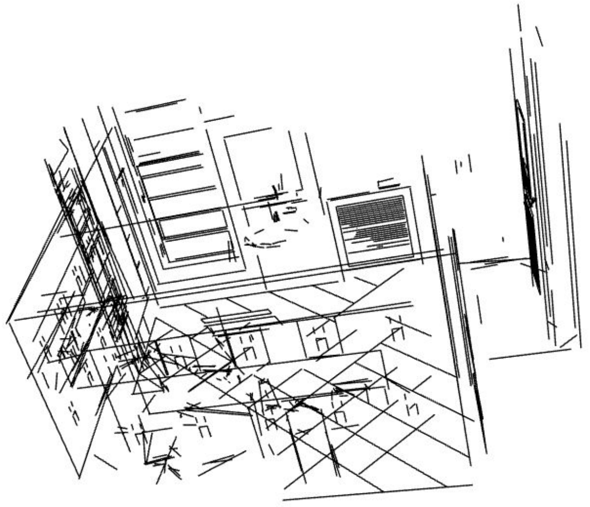
- CLMAP is a direct fork of the LIMAP repository
  - Additional components introduced in CLMAP are included as new .cpp and .py files
  - This is convenient for running experiments, as runner scripts are the same for both packages

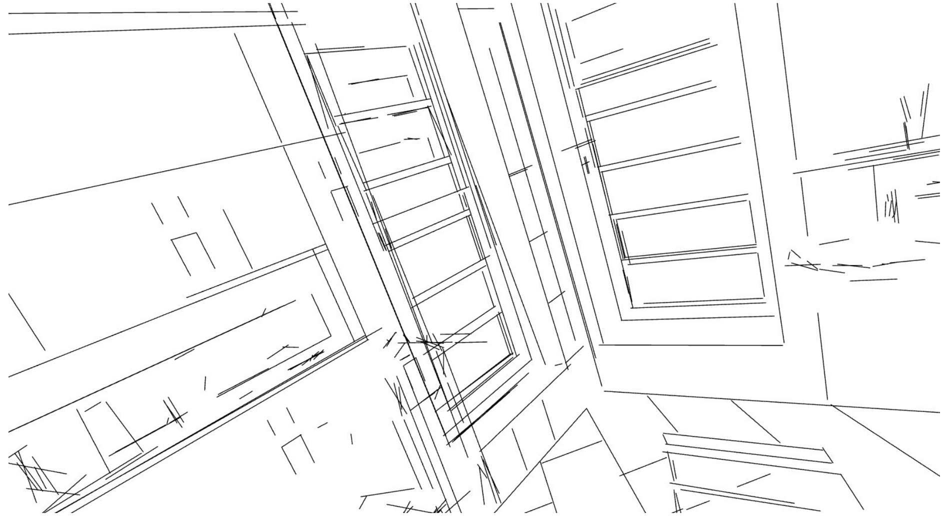
- Besides those new and modified files for CLMAP's methodology, the file structure, configuration setup, and runner procedures are all the same
  - While this is convenient for experimental purposes, it also creates lots of versioning conflicts if you are trying to run both on a single machine, given that CLMAP was released a year later

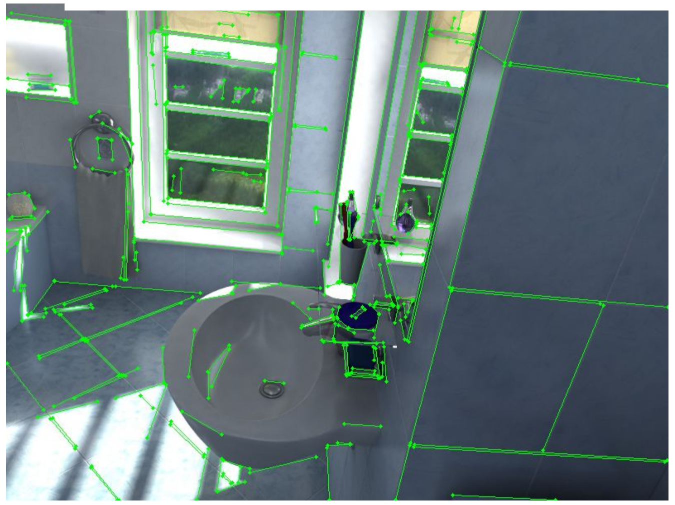
LIMAP - Built in Example

- LIMAP for Hypersim Scene 1 (bathroom)
  - Experimental Runner provided in repo

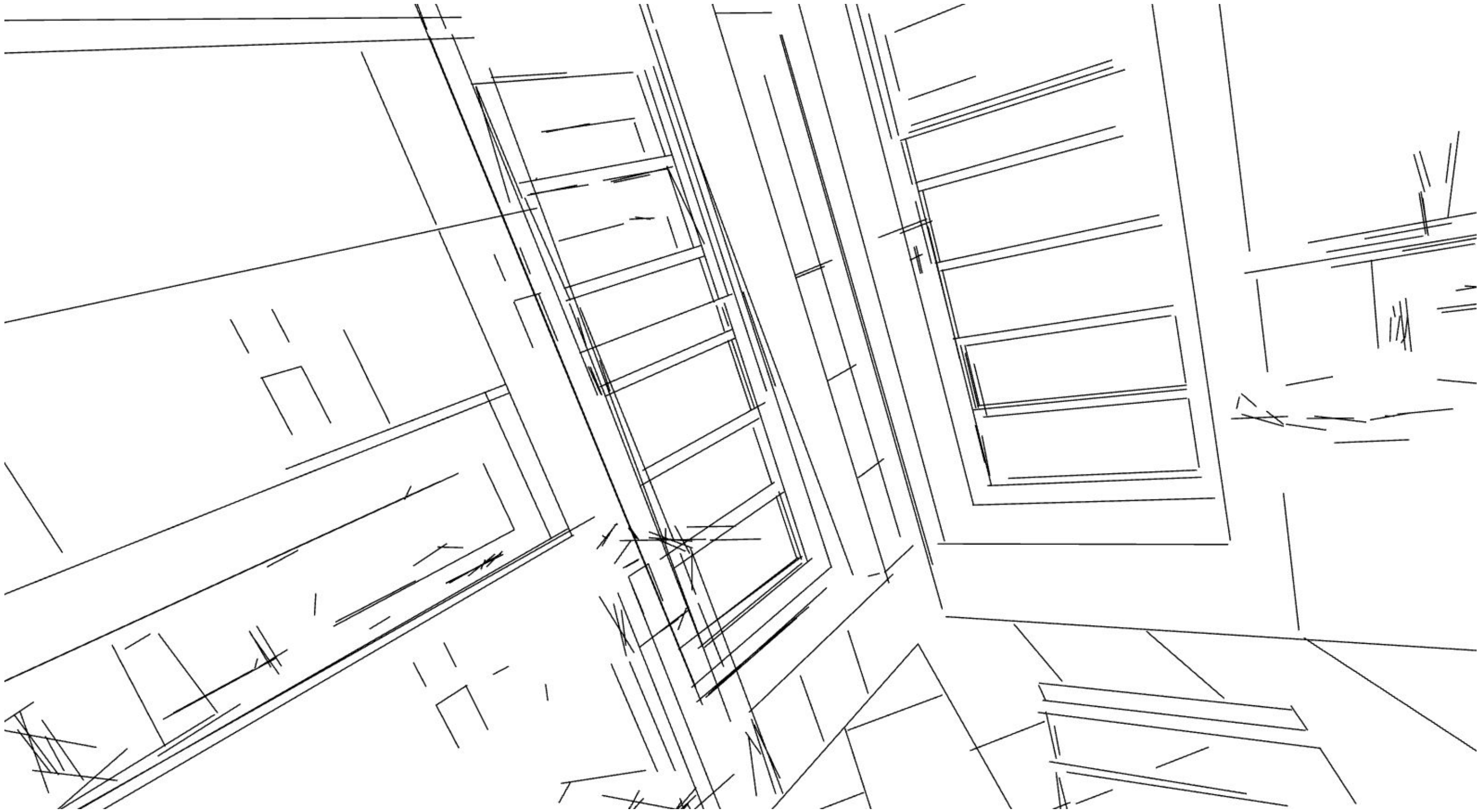
- Successful 3D Line Reconstruction







LIMAP - Built in Example - Zoomed



LIMAP - Replicating Paper

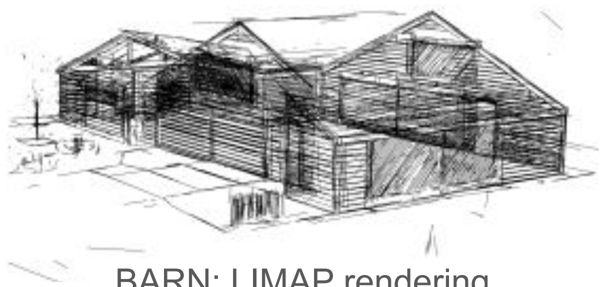
- Unfortunately, only the one sample had supporting scripts
  - *hypersim.py, hypersim/line_triangulation.sh, and hypersim_eval.py, … 7 total scripts*
  - This setup is very unfriendly to new datasets

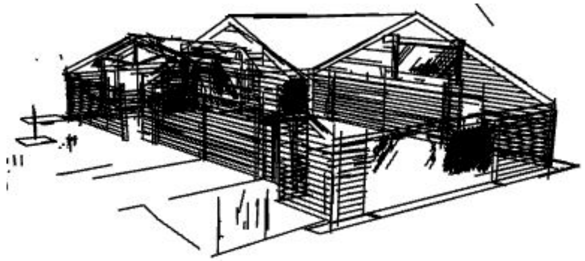
- New runner scripts were developed to allow for the input of raw datasets, particularly aimed to replicate the Tanks & Temples dataset and run ABC CAD

- This new procedure resulted in the following steps for each scene:
  - Download the image dataset as well as the ground truth for that scene (including a .ply file, alignment information and poses) and put it into a folder matching the configuration name
  - Using the COLMAP Structure From Motion package, run to get a collection of 3D points
  - With that data, run LIMAP's built in triangulation functionality to get the 3D line model
  - Run the modified evaluation script to get the quantitative result table for the input scene, or run the modified visualize_3d_lines python script for a 3D rendering of the result & GT data

LIMAP - Replicating Paper - Qualitative Results

- By creating more customisable scripts tuned for an audit of the line mapping process, running results on raw datasets was simplified
- These results were validated on a qualitative level by comparing with images published in the papers
- While configurations may vary, the line reconstructions match well


BARN: LIMAP rendering published in Paper


BARN: LIMAP rendering recreated with new runner scripts

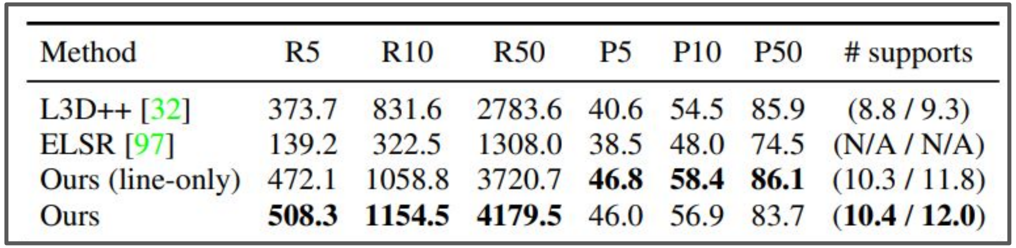
Liu, Shaohui, Yifan Yu, Rémi Pautrat, Marc Pollefeys, and Viktor Larsson. "3d line mapping revisited." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21445-21455. 2023.

LIMAP - Replicating Paper - Quantitative Results

- The scenes are then evaluated against Ground Truth and the summary metrics are presented for
  - R_T: sum of line lengths within T mm
  - P_T: percentage of line tracks within T mm
  - # line and # image supports present

- To replicate the results, all 7 training scenes for T&T were processed
  - Note that “Ignatius” is left out of these metrics… more on that later

LIMAP - Replicating Paper - Quantitative Results



Method	R5	R10	R50	P5	P10	P50	# supports
L3D++ [32]	373.7	831.6	2783.6	40.6	54.5	85.9	(8.8 / 9.3)
ELSR [97]	139.2	322.5	1308.0	38.5	48.0	74.5	(N/A / N/A)
Ours (line-only)	472.1	1058.8	3720.7	**46.8**	**58.4**	**86.1**	(10.3 / 11.8)
Ours	**508.3**	**1154.5**	**4179.5**	46.0	56.9	83.7	(**10.4 / 12.0**)
'Ours' Replicated	**525.7**	**2374.0**	3711.9	**47.8**	**73.0**	82.1	**(15.3 / 23.8)**

LIMAP T&T: Reported Metrics; Replicated Metrics

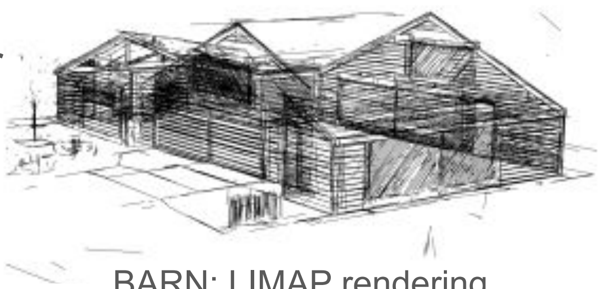
- In general, LIMAP results are a decent match
- We wouldn't expect this to line up exactly given that we don't have configs

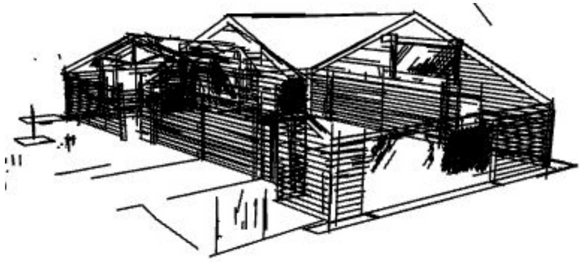
Liu, Shaohui, Yifan Yu, Rémi Pautrat, Marc Pollefeys, and Viktor Larsson. "3d line mapping revisited." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21445-21455. 2023.

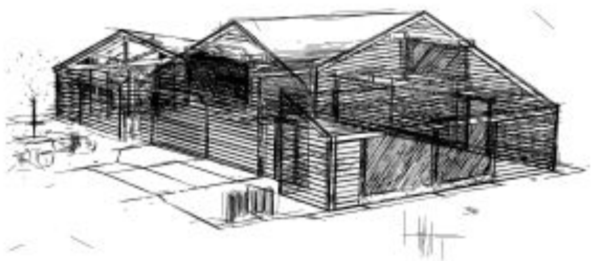
CLMAP - Replicating Paper - Qualitative Results

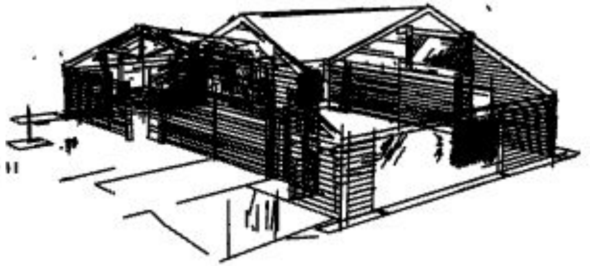
- CLMAP produces similar result as the paper CLMAP rendering

- Additionally, LIMAP vs. CLMAP has similar patterns as the paper
  - Denser line bundles
  - Sharper, more accurate alignment


BARN: LIMAP rendering published in Paper


BARN: LIMAP rendering recreated with new runner scripts

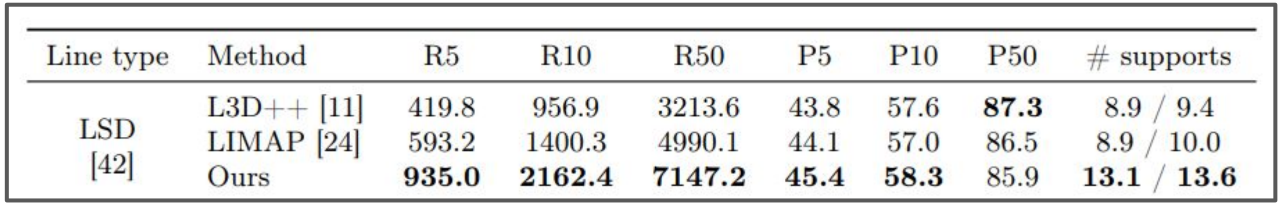

BARN: CLMAP rendering published in Paper


BARN: CLMAP rendering recreated with new runner scripts

Liu, Shaohui, Yifan Yu, Rémi Pautrat, Marc Pollefeys, and Viktor Larsson. "3d line mapping revisited." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21445-21455. 2023.

Bai, Xulong, Hainan Cui, and Shuhan Shen. "Consistent 3D Line Mapping." In *European Conference on Computer Vision*, pp. 57-74. Cham: Springer Nature Switzerland, 2024.

CLMAP - Replicating Paper - Quantitative Results



Line type	Method	R5	R10	R50	P5	P10	P50	# supports
LSD [42]	L3D++ [11]	419.8	956.9	3213.6	43.8	57.6	**87.3**	8.9 / 9.4
	LIMAP [24]	593.2	1400.3	4990.1	44.1	57.0	86.5	8.9 / 10.0
	Ours	**935.0**	**2162.4**	**7147.2**	**45.4**	**58.3**	85.9	**13.1** / **13.6**

LIMAP Replicated	**525.7**	**2374.0**	**3711.9**	**47.8**	**73.0**	**82.1**	(15.3 / 23.8)
'Ours' Replicated	473.5	2092.1	3152.0	47.3	72.8	80.5	**(19.8 / 30.9)**

LIMAP/CLMAP T&T: Reported Metrics; Replicated Metrics

- A bit of an unexpected result here, except for number of supports LIMAP outperforms CLMAP; although they are close
- Likely a configuration issue or slight difference in how CLMAP was run

Liu, Shaohui, Yifan Yu, Rémi Pautrat, Marc Pollefeys, and Viktor Larsson. "3d line mapping revisited." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21445-21455. 2023.

Bai, Xulong, Hainan Cui, and Shuhan Shen. "Consistent 3D Line Mapping." In *European Conference on Computer Vision*, pp. 57-74. Cham: Springer Nature Switzerland, 2024.

Codebase: Issues in Implementation

- Setup Difficulties - Environment and Dependencies
    - Deprecated code. Had to fiddle around to find a combination of older versions of dependencies that worked.
    - Had not worked with pybind11 before, learning curve to compiling and linking programs

- Data size & publicity
    - Only one of the Hypersim samples (100 images) is publicly available for free
    - Even if they were, the hypersim dataset is 1.9TB. Hard to get representative sample
    - Tanks and Temples training set (~50 GB) is more manageable but still hefty
        - Testing dataset on GPU (RTX 4070) still had runtime constraints

- Data Formatting & Reproducibility
    - Repository is hard coded for research, difficult to run new or differently formatted data

Strengths, Issues, and Proposed Use Cases

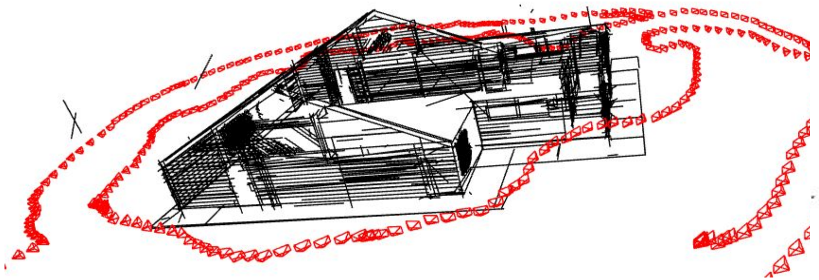
Section 3: ***(10 minutes)***

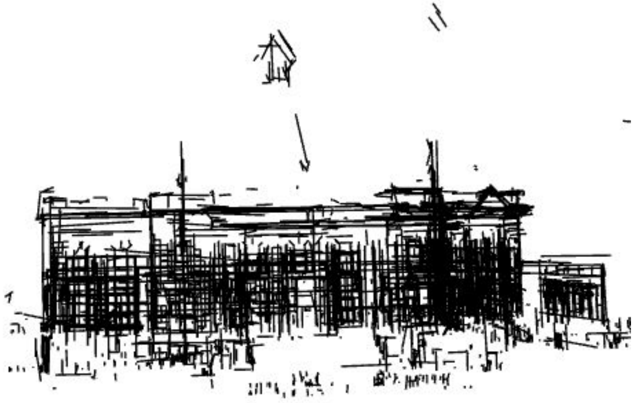
- Breaking down paper results - individual scenes and Ground Truth comparisons

- Extending to Organic Structures - Problems with *too nonlinear* data
- Extending to ABC Dataset - Problems with *too smooth* data

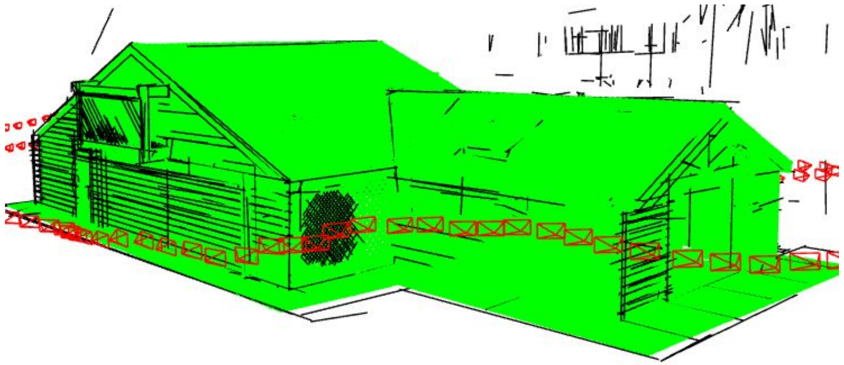
More Detailed Visualizations: Ground Truth Overlay

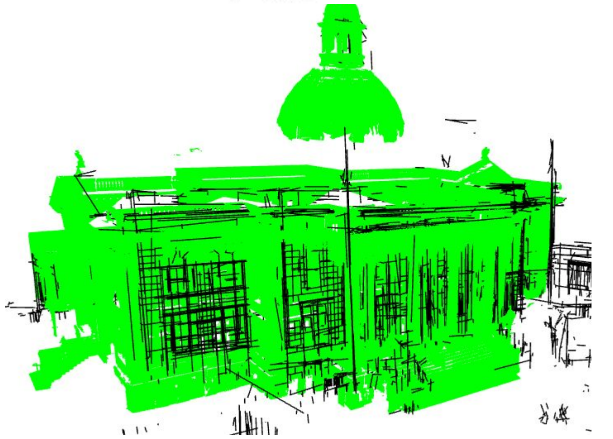
Provided Visualisation:



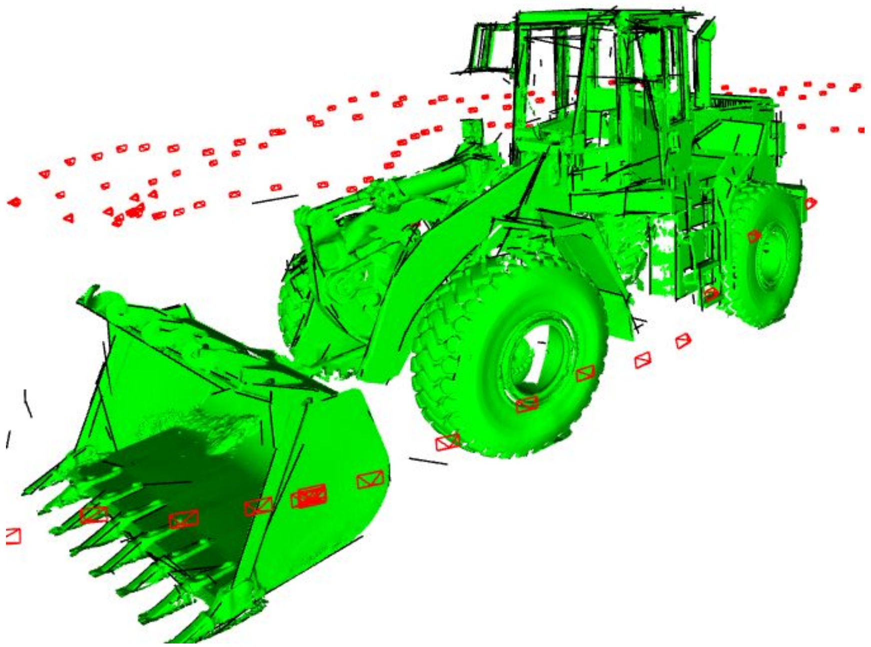


GT Aligned & Overlaid:



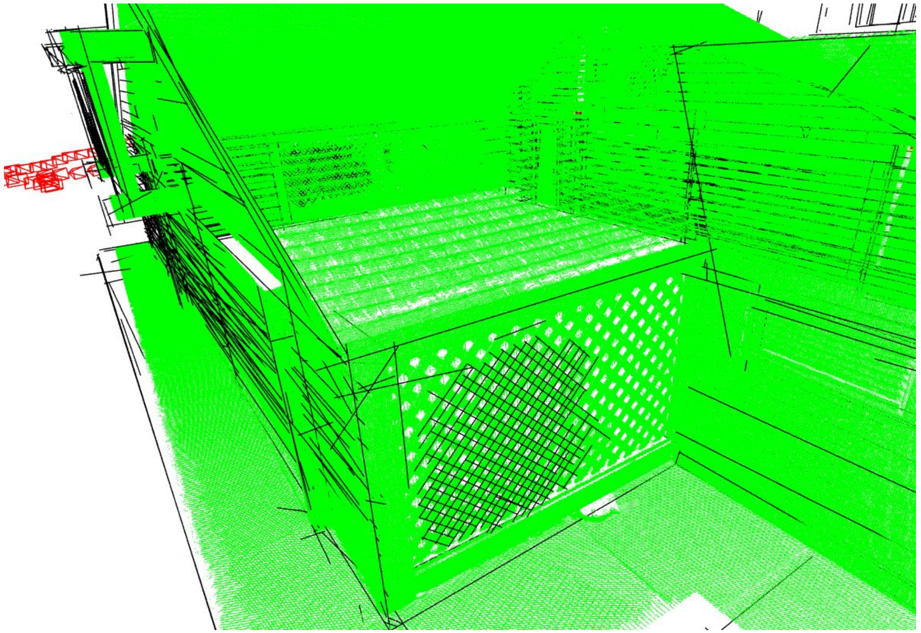


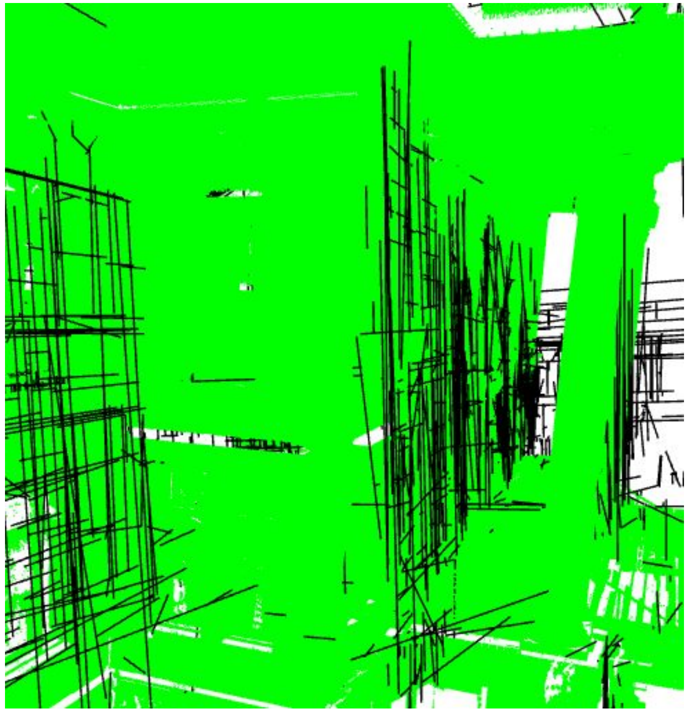
More Detailed Visualizations Reveal Strengths



Tanks and Temples: Caterpillar

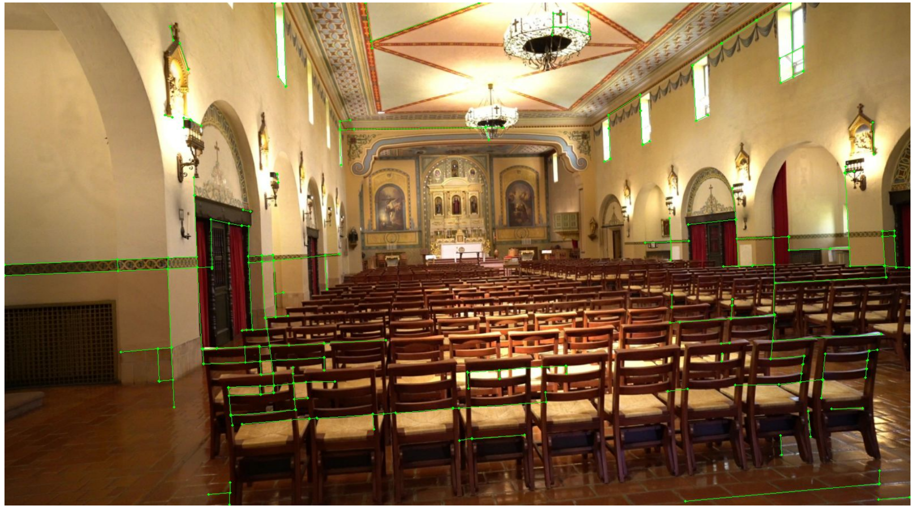
More Detailed Visualizations Reveal Strengths

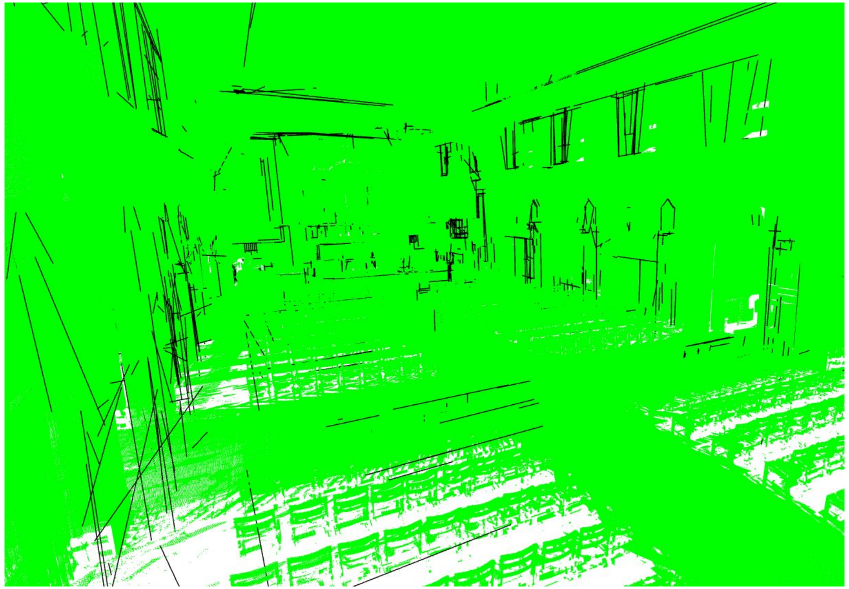

T&T Barn: Notice Detailed Outcroppings


T&T Courthouse: Notice Pillar Depths

More Detailed Visualizations Reveal Strengths

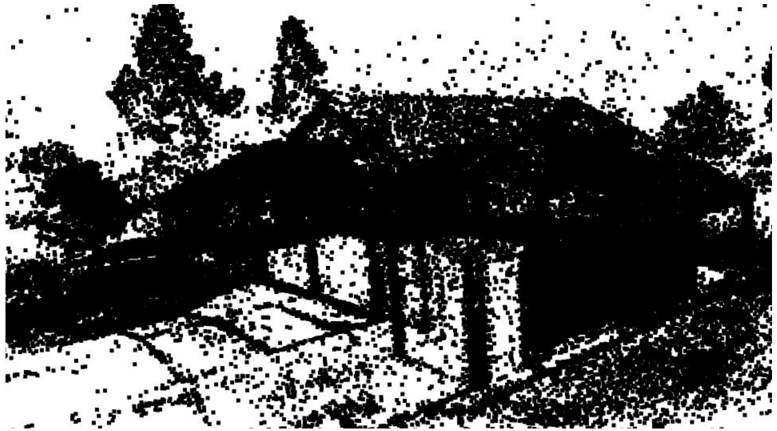
- Decent results on indoor environments too
- Correctly captures scale and some detail in the room's structure

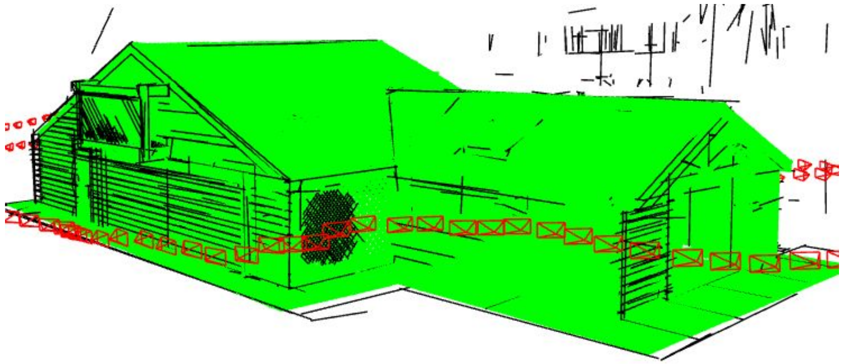

T&T Church - Image w/ Line Detections


T&T Church - 3D Reconstruction

Comparisons w/ Point Based Model (COLMAP)

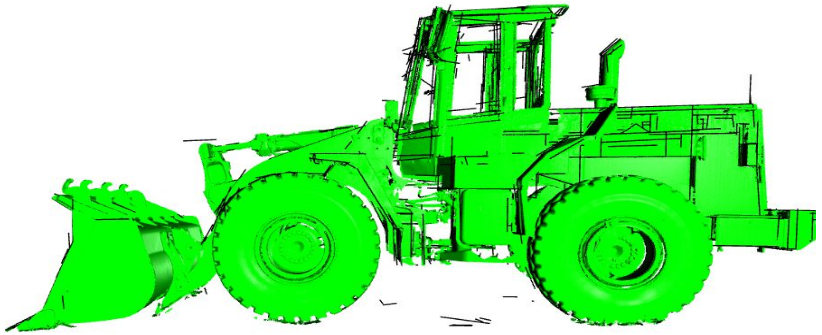
- Looking at pure point-based SfM, LIMAP far exceeds semantic understanding
- Also provides more structural detail, though it falls short on background info

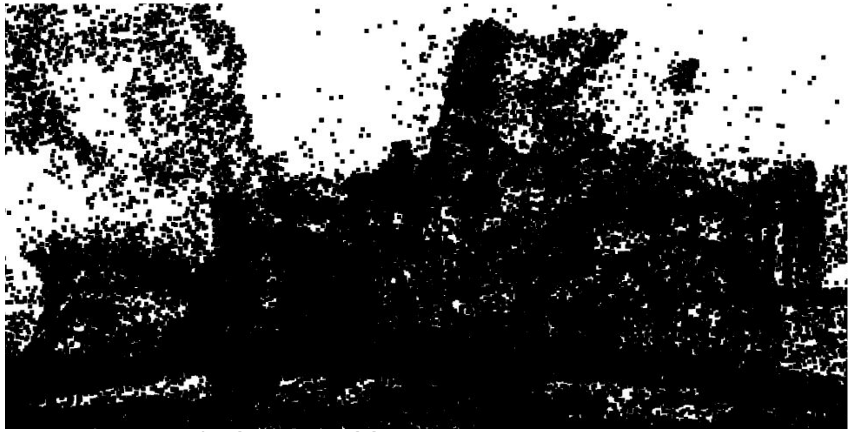

T&T Barn - COLMAP Point-Based SfM

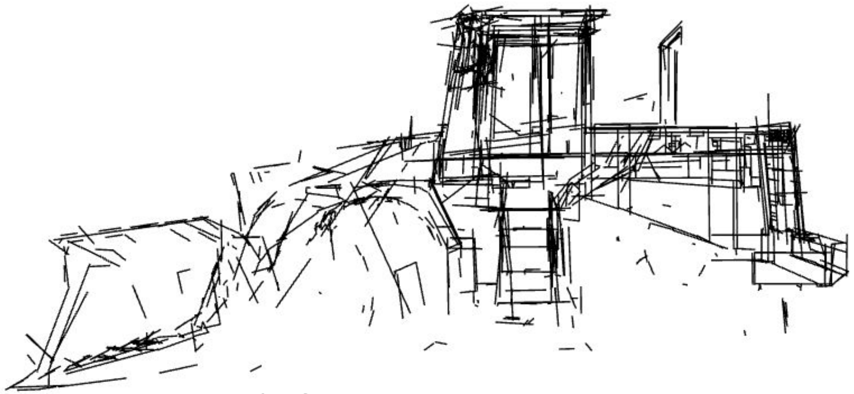

T&T Barn - LIMAP Result

Comparisons w/ Point Based Model (COLMAP)

- This semantic advantage is seen on more complex structures too, not just rectangular & uniform scenes


T&T Caterpillar- GT Reference


T&T Caterpillar- COLMAP Reconstruction

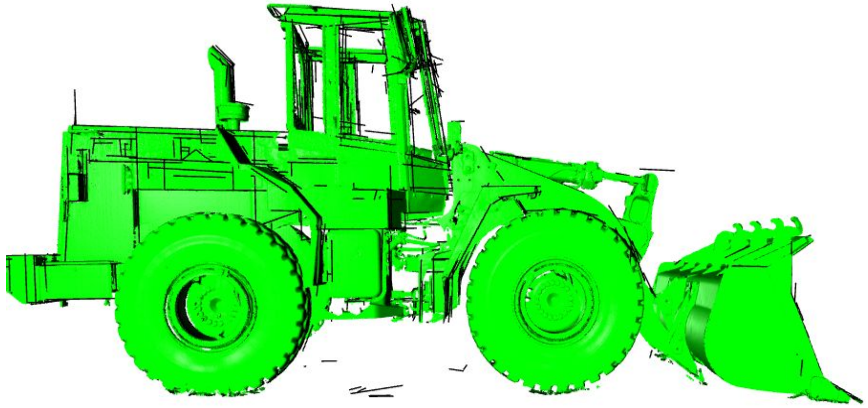

T&T Caterpillar- LIMAP Reconstruction

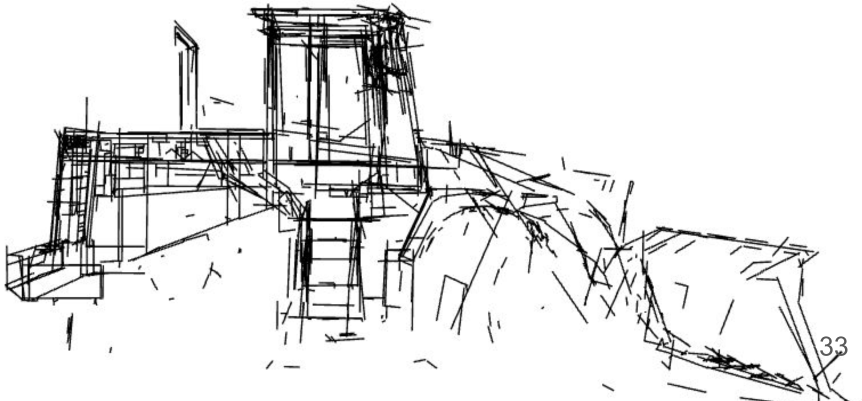
Certain Scenes Reveal Weaknesses

- *Line* detection has an obvious drawback

- This immediately lowers usefulness for most organic structures, also machines w/ wheels, gears, or other curvature

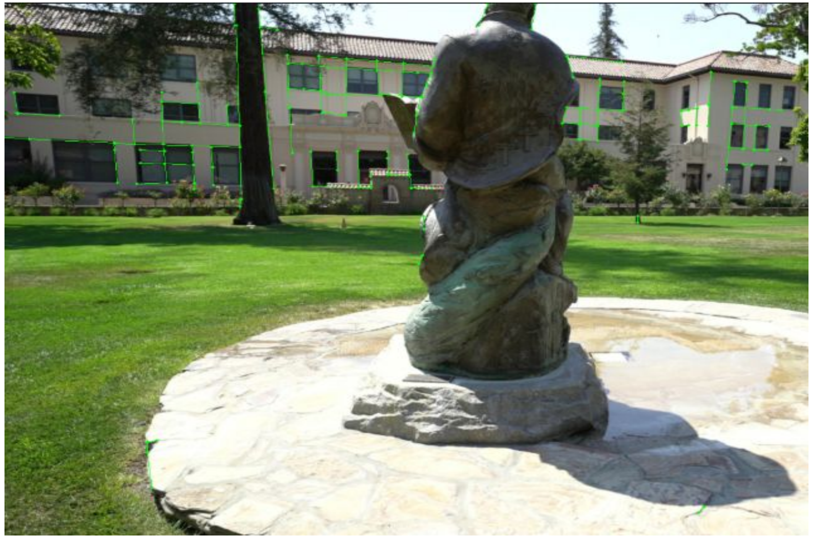
- We need to allow for smaller line detections to try and reconstruct these

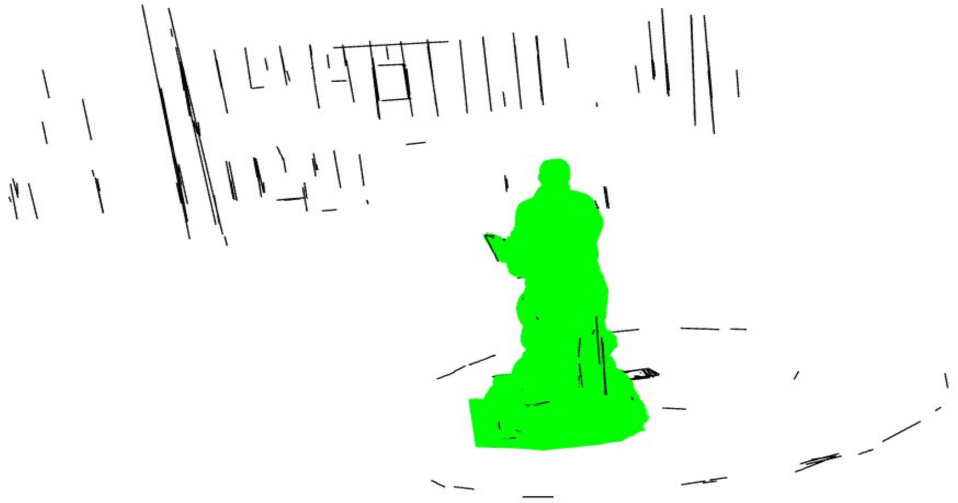




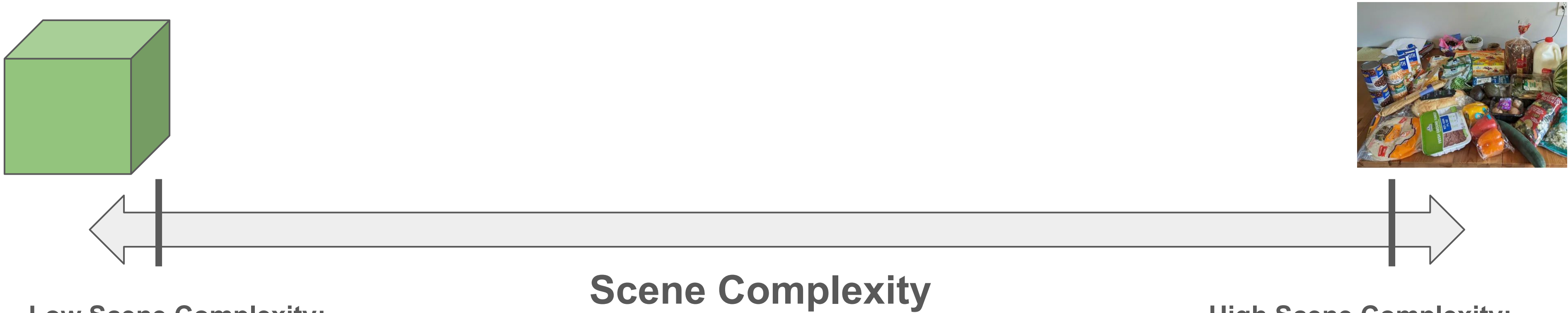
Certain Scenes Reveal Weaknesses

- However, in real life this isn't the most practical approach
- Additionally, background patterns can dominate the object of focus
    - A scene like the one below would be better suited for an edge-based approach


Ignatius Image


Ignatius 3D Reconstruction

When to use LIMAP/CLMAP?



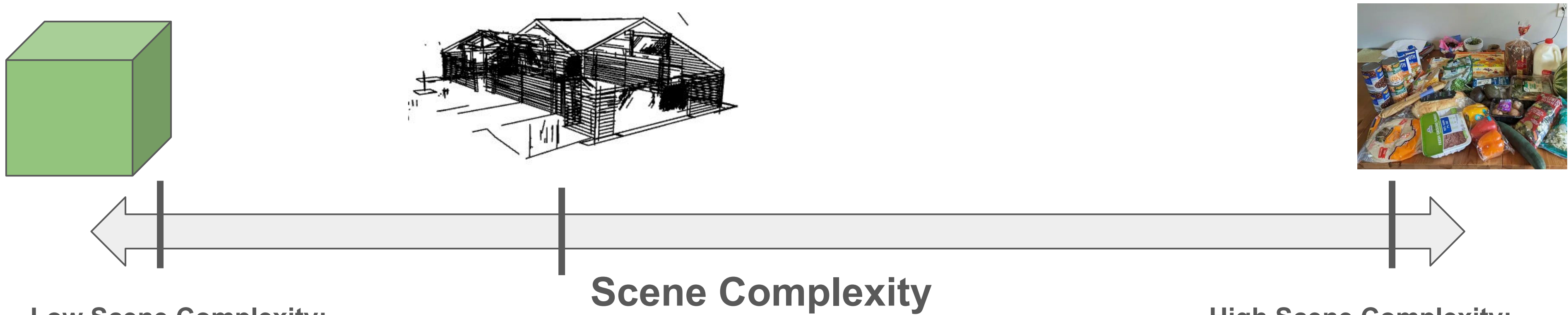
Low Scene Complexity:

- Simple Structure
- Geometric Lines
- Smooth Texture

High Scene Complexity:

- Organic Structure
- Curves and Gaps
- Chaotic Texture

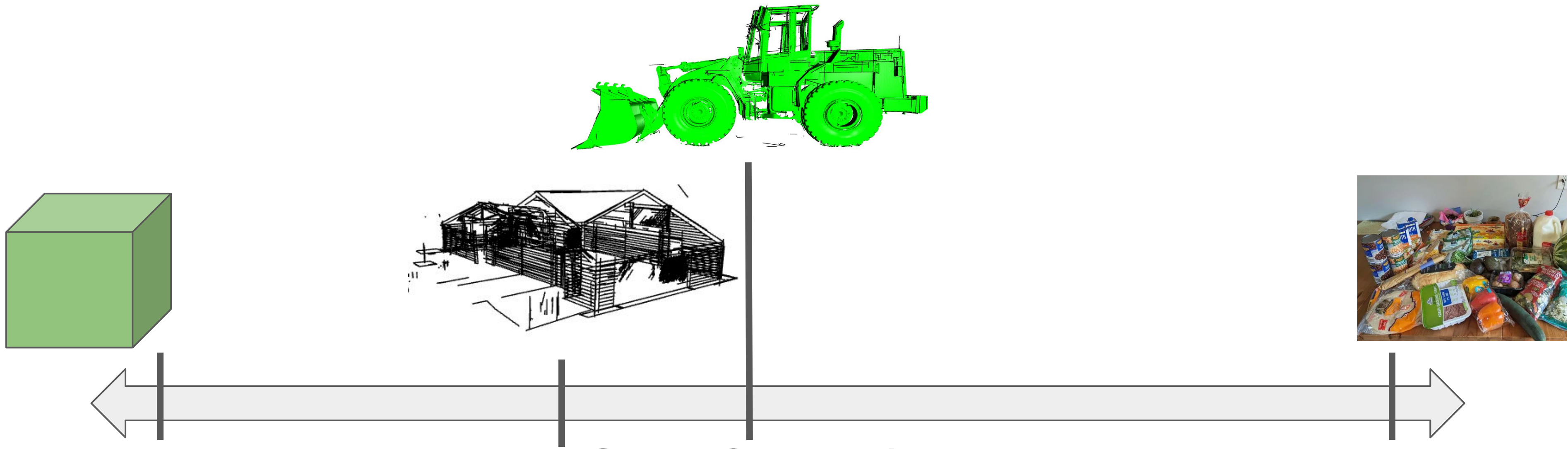
When to use LIMAP/CLMAP?



Low Scene Complexity:
- Simple Structure
- Geometric Lines
- Smooth Texture

High Scene Complexity:
- Organic Structure
- Curves and Gaps
- Chaotic Texture

When to use LIMAP/CLMAP?



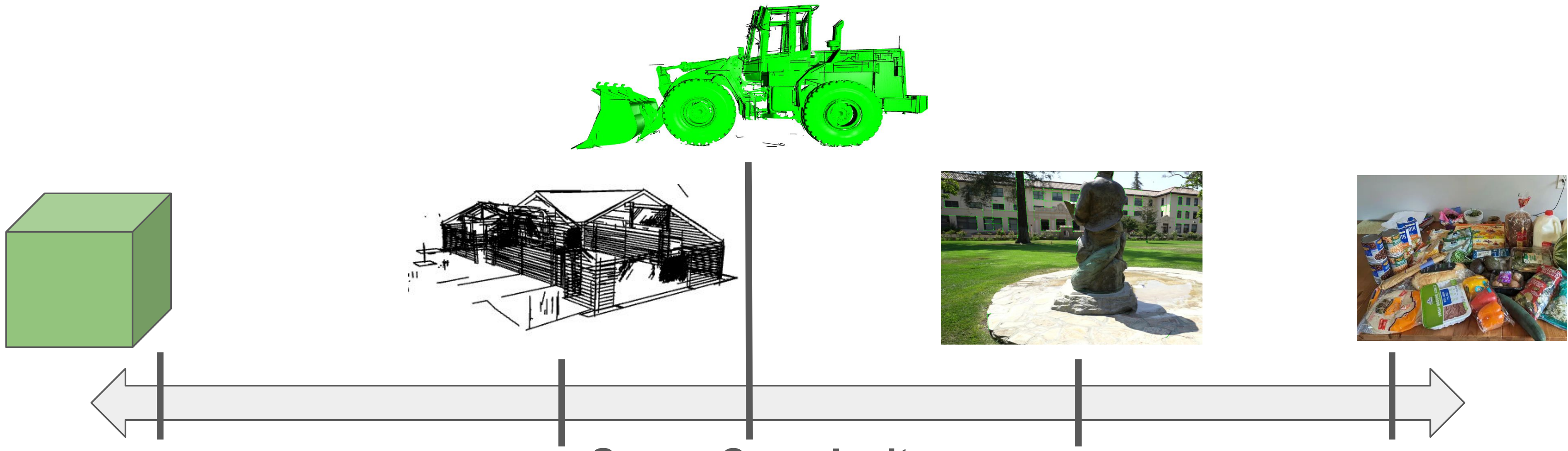
Low Scene Complexity:

- Simple Structure
- Geometric Lines
- Smooth Texture

High Scene Complexity:

- Organic Structure
- Curves and Gaps
- Chaotic Texture

When to use LIMAP/CLMAP?



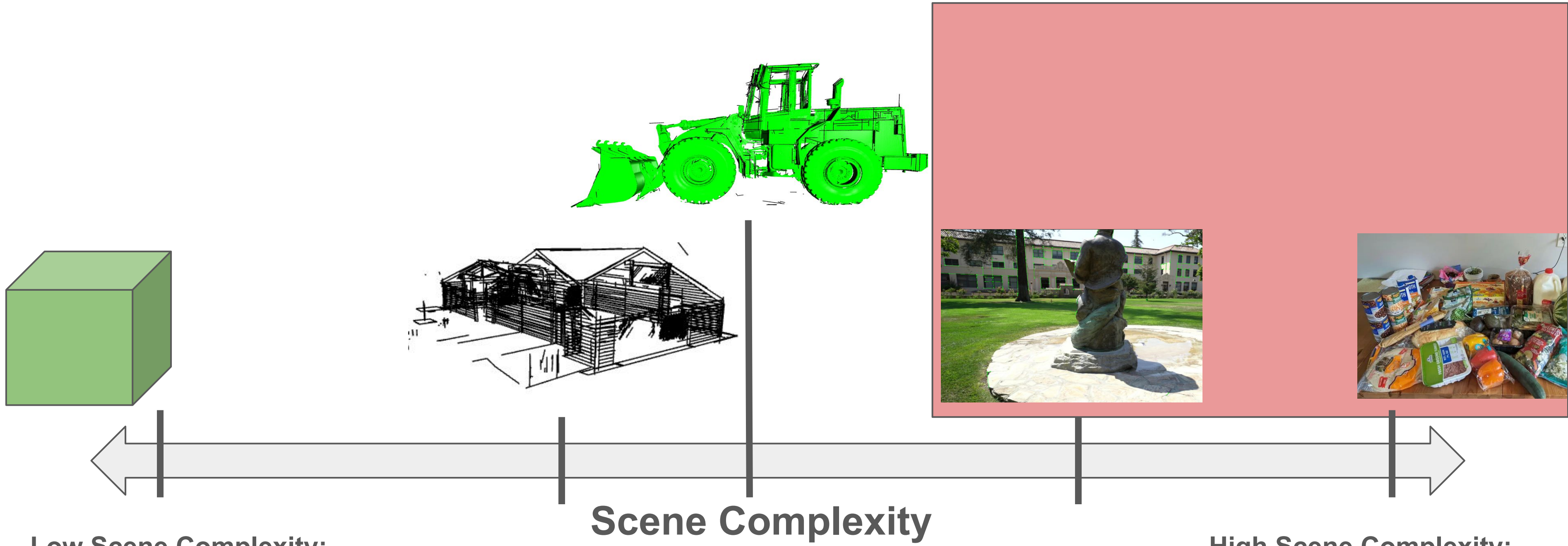
Low Scene Complexity:

- Simple Structure
- Geometric Lines
- Smooth Texture

High Scene Complexity:

- Organic Structure
- Curves and Gaps
- Chaotic Texture

When to use LIMAP/CLMAP?



Low Scene Complexity:
- Simple Structure
- Geometric Lines
- Smooth Texture

High Scene Complexity:
- Organic Structure
- Curves and Gaps
- Chaotic Texture

Low Complexity Scenes? Introducing ABC-NEF

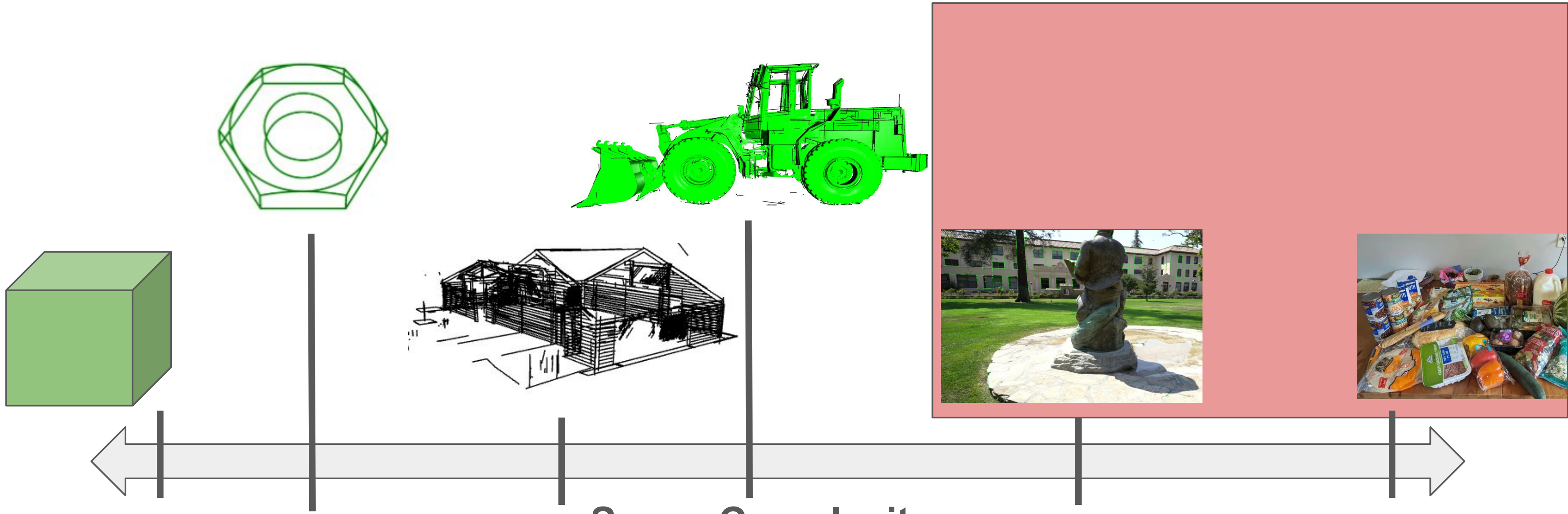
- Part of the original goal of the project was to compare results of Line Reconstruction with Edge Reconstruction on simple objects

- However, upon plugging in these CAD scenes into LIMAP, it became clear that these ultra-smooth shapes did not have the texture to support the SfM required for robust mapping

- The existing LIMAP repository could not generate any 3D reconstruction *at all*

- Hypothetically, using just Method 3 (Vanishing Points) would work, but that functionality was not built into the repository and I didn't have time to create a runner script for that

When to use LIMAP/CLMAP?



Scene Complexity

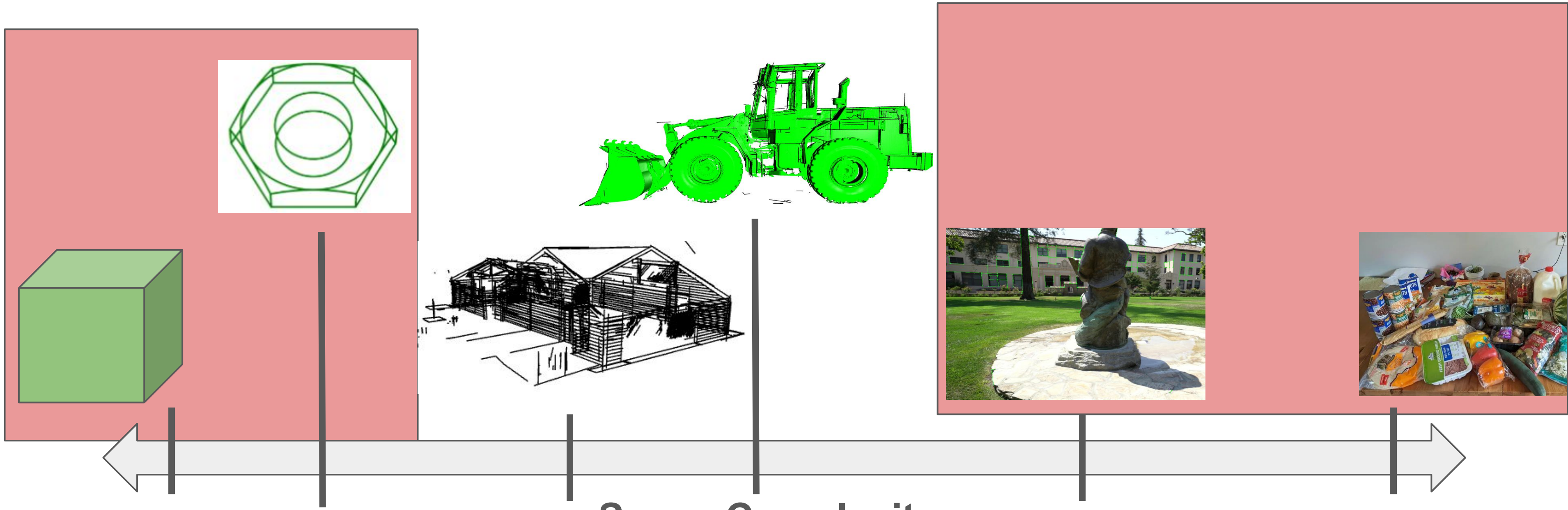
Low Scene Complexity:

- Simple Structure
- Geometric Lines
- Smooth Texture

High Scene Complexity:

- Organic Structure
- Curves and Gaps
- Chaotic Texture

When to use LIMAP/CLMAP?



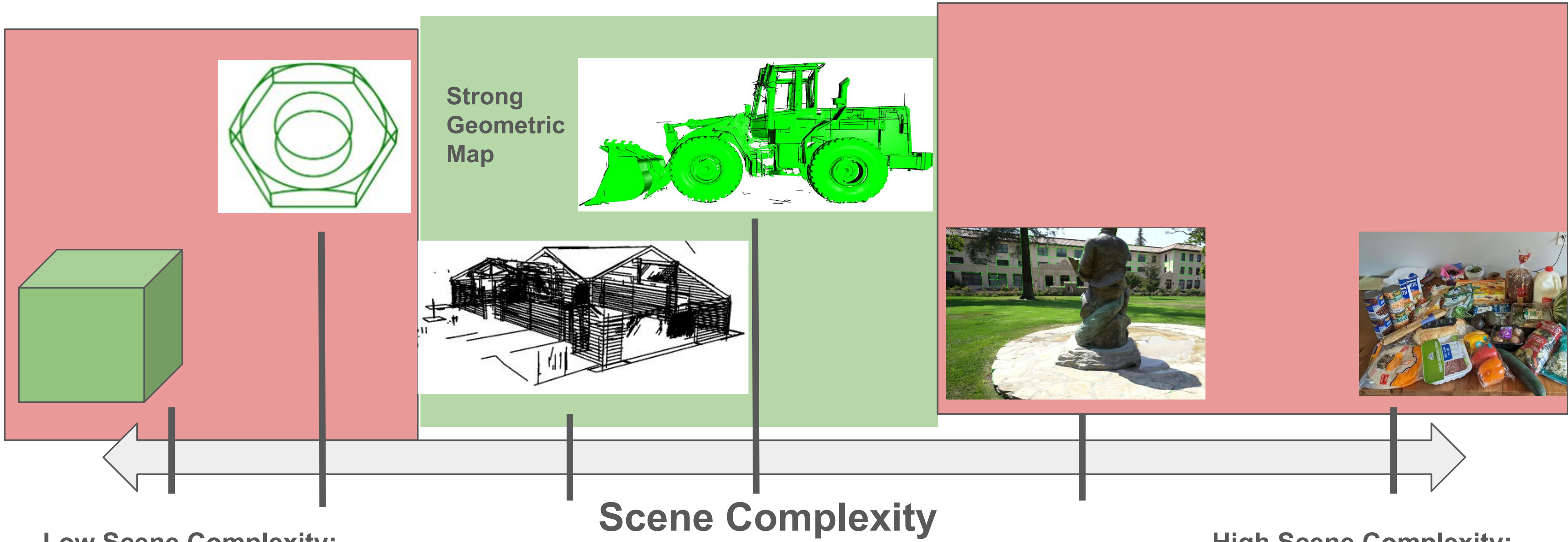
Low Scene Complexity:

- Simple Structure
- Geometric Lines
- Smooth Texture

High Scene Complexity:

- Organic Structure
- Curves and Gaps
- Chaotic Texture

When to use LIMAP/CLMAP?



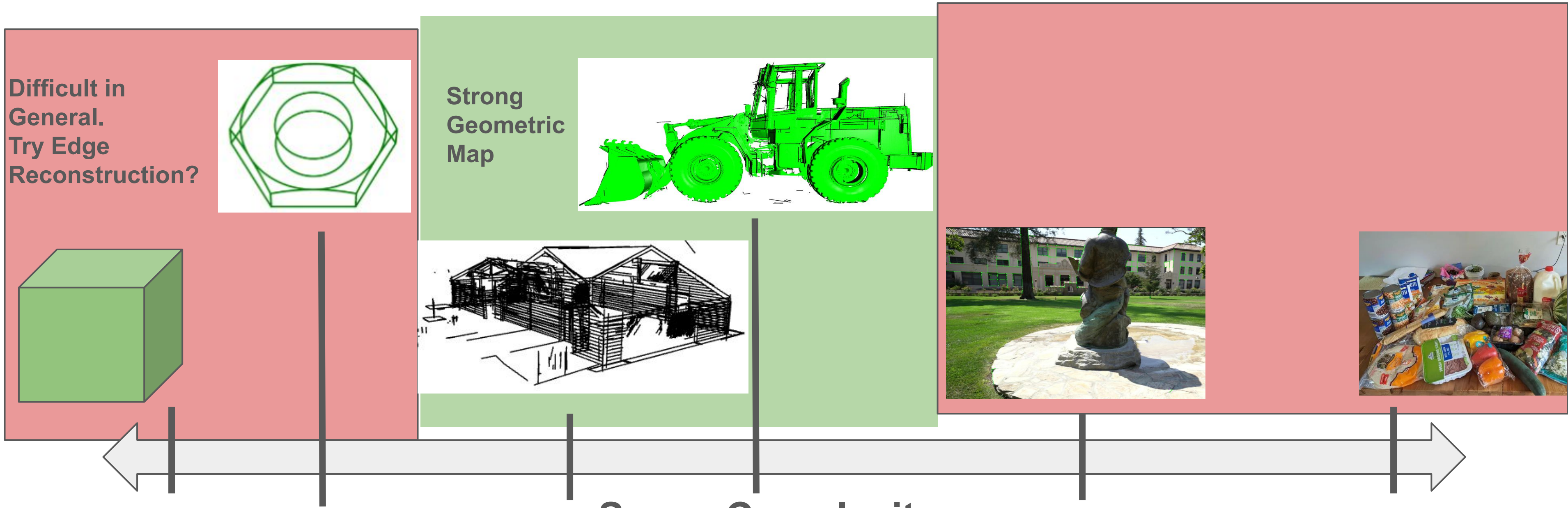
Low Scene Complexity:

- Simple Structure
- Geometric Lines
- Smooth Texture

High Scene Complexity:

- Organic Structure
- Curves and Gaps
- Chaotic Texture

When to use LIMAP/CLMAP?



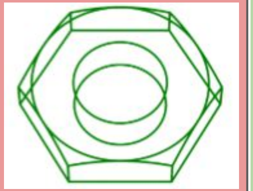
Scene Complexity

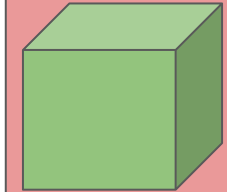
Low Scene Complexity:
- Simple Structure
- Geometric Lines
- Smooth Texture

High Scene Complexity:
- Organic Structure
- Curves and Gaps
- Chaotic Texture

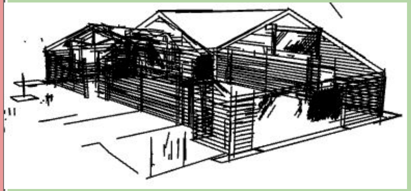
When to use LIMAP/CLMAP?

Difficult in General.
Try Edge Reconstruction?





Strong Geometric Map



Can’t Render Usable Lines.
Try Point or Edge Reconstruction?

Scene Complexity

Low Scene Complexity:

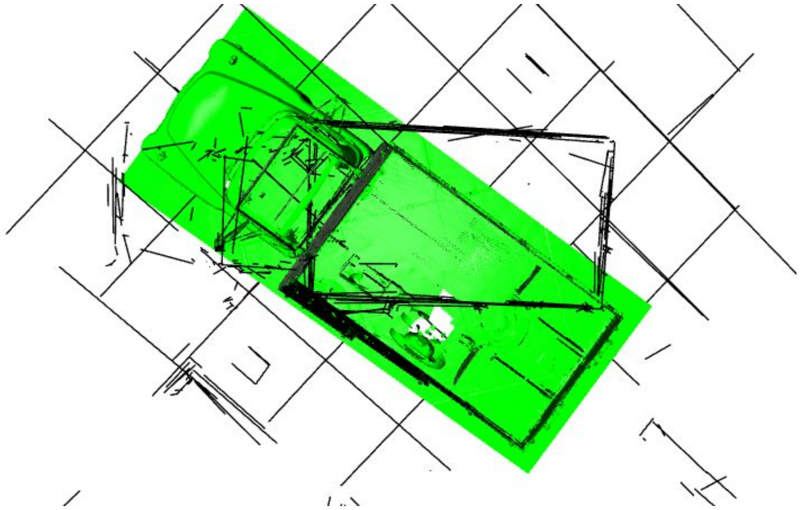
- Simple Structure
- Geometric Lines
- Smooth Texture

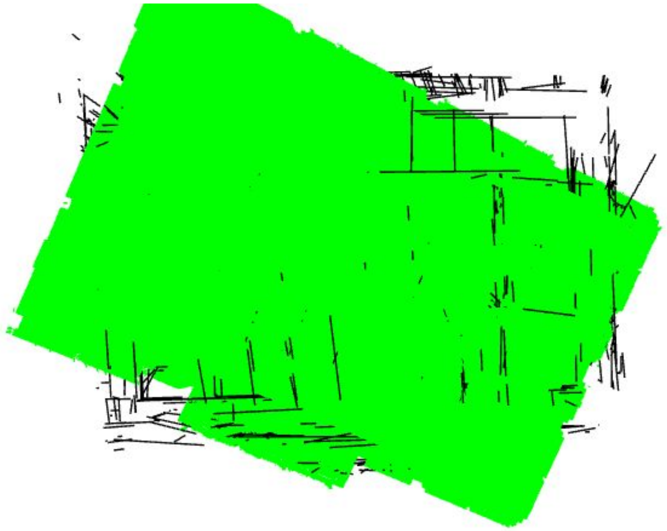
High Scene Complexity:

- Organic Structure
- Curves and Gaps
- Chaotic Texture

Sidenote 1: Detailed Visualizations Show Potential Bug

- Two scenes appear to have an alignment bug using the public data, the same data that the paper ran its results on. Were these bugs noticed and corrected for? If not the results may be incorrect (I have omitted these from results)


Truck


Meeting room

Sidenote 2: Limap Ablation Study

- LIMAP has lots of options for input parameters:
  - Line Detector (LSD, DeepLSD)
  - SfM number points to use (method)
  - SfM sourcing (COLMAP, other)
  - General Parameters (number of neighboring images / possible tracks)

- Options Tried:
  - Number of views (4,10, 40) to use
  - Number of neighbors to use (20, 40, 100)
  - Number of images in dataset, 100, 400 – effectively laps around scene
  - LSD vs. SOLD2

Documented Changes & Improvements

1. More Universal Experimental Procedure
    a. Main chunk of additional code. Updates a few different dataset-dependent scripts
    b. Clearer procedure developed for introducing data, now can input pure image/movie data

2. Improved Visualization Code
    a. Overlaid point clouds, identified potential T&T dataset inconsistencies for LIMAP evaluation

Original vs. Final Timeline Summary

- Main Differences:
  - Running LIMAP first than CLMAP
  - Diving more in depth on Tanks & Temples Dataset
  - Diving more into general strengths/weakness, less in depth on ABC-NEF paper

3/12-3/19: Choose Project; Paper Review; Initial Presentation

3/19-3/26: Finalize Paper Understanding; LIMAP & CLMAP Code Review
3/26-4/02: LIMAP & CLMAP Implementation & Debugging
4/02-4/09: LIMAP & CLMAP Running Test Datasets;
4/09-4/16: Incorporate & Debug ABC-NEF Dataset; Mid-Project Presentation

4/16-4/23: Run LIMAP & CLMAP on ABC-NEF; Analyze Results
4/23-4/30: Recreate Result Plots from 3D Edge Sketch on LIMAP / CLMAP
4/30-5/06: Compare Results from LIMAP / CLMAP with 3D Edge Sketch

5/06-5/14: Final Cleanup / Buffer Week; Preparation for Final Presentation
5/15-5/16: Final Presentation

3/12-3/19: Choose Project; Paper Review; Initial Presentation

3/19-3/26: Finalize Paper Understanding; LIMAP & CLMAP Code Review
3/26-4/02: LIMAP & CLMAP Implementation & Debugging
4/02-4/09: LIMAP Running Test Datasets;
4/09-4/16: Incorporate & Debug T&T, ABC-NEF; Mid-Project Presentation

4/16-4/23: Create Runner Code for LIMAP on T&T and ABC-NEF
4/23-4/30: Run LIMAP on T&T and ABC-NEF; Replicate Quantitative Results
4/30-5/06: Create improved visualizations; Quantify successes/failures

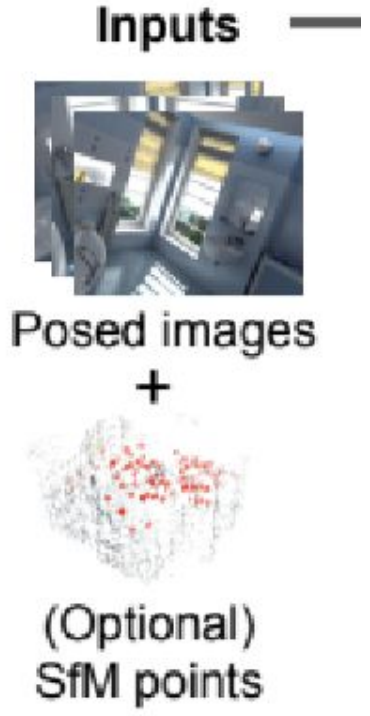
5/06-5/14: Replicate LIMAP results on CLMAP; Compare and Analyze
5/15-5/16: Cleanup & Preparation for Final Presentation

Appendix:

- The following slides were skipped from the final presentation for brevity
- Mostly more in depth information about the papers from the initial presentation

Proposed Approach - LIMAP

- Assumes we have the following data:
    - Set of input images with accurate pose information (either from prior knowledge or a SfM/SLAM estimate)
    - ***Existing lines***: 2D segments in the images have already been detected using any line-based feature identification algorithm (LSD, DeepLSD, etc…)
    - Optional: Point based SfM features (can be used to support robust triangulation and improve scoring)



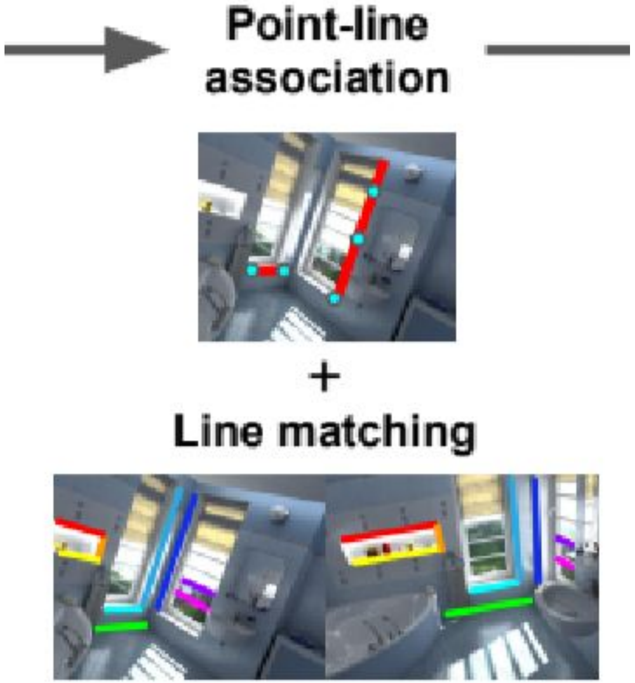
Liu, Shaohui, Yifan Yu, Rémi Pautrat, Marc Pollefeys, and Viktor Larsson. "3d line mapping revisited." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21445-21455. 2023.

Proposed Approach - LIMAP

- With the line segments, we use any existing line matcher to identify the top K matches in *other* images (use the n_v closest images)

- We take all of these potential matches to triangulation, they are filtered later in the process

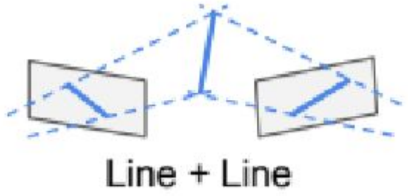
- 3D lines are defined by 2 3D endpoints
- These have (currently unknown) depths λ_1,λ_2



Liu, Shaohui, Yifan Yu, Rémi Pautrat, Marc Pollefeys, and Viktor Larsson. "3d line mapping revisited." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21445-21455. 2023.

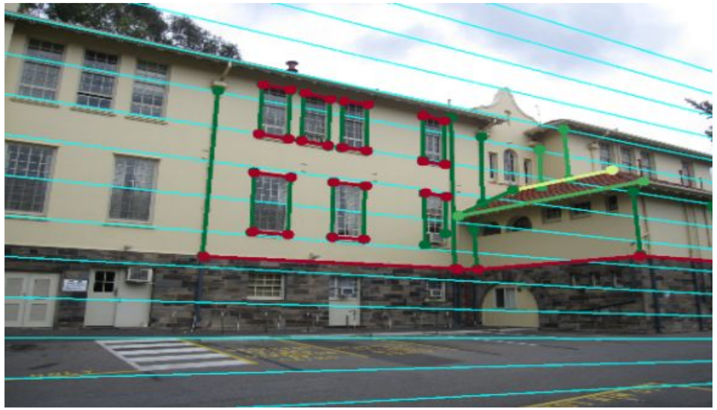
Proposed Approach - LIMAP

- A major difficulty with line triangulation is degeneracy
  - If lines are parallel to epipolar lines, no one solution can be found
  - This results in no identified depth for that particular point



- For 1 or 2 missed depths, line-to-line matching is unreliable
  - 0 Degenerate Endpoints: “Not degenerate”
  - 1 Degenerate Endpoint: “Weakly Degenerate”
  - 2 Degenerate Endpoints: “Fully Degenerate”

- This is where LIMAP comes in



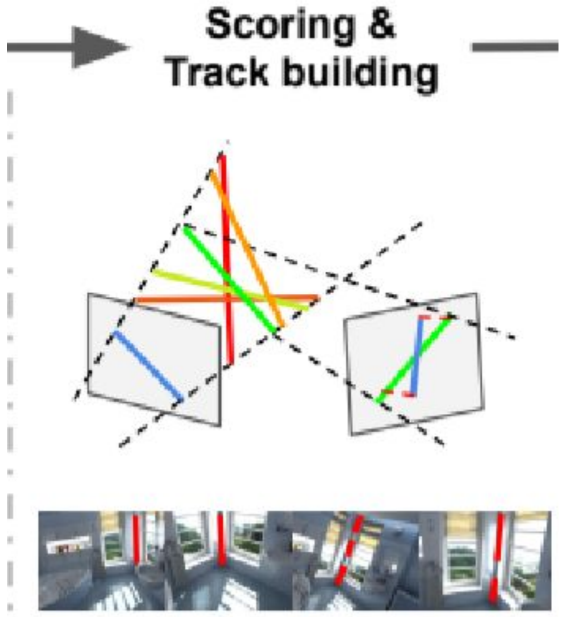
Liu, Shaohui, Yifan Yu, Rémi Pautrat, Marc Pollefeys, and Viktor Larsson. "3d line mapping revisited." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21445-21455. 2023.

Proposed Approach - LIMAP

- At this point, each initial segment *l* is associated with K 3D line segment proposals per neighboring image

- Score each proposal to get one value by combining:
- 2 traditional metrics:
  - Angular Distance between lines
  - Perpendicular Distance between lines
- 1 novel metric:
  - Perspective Distance (definition omitted for brevity)
    - Used to filter out bad matches from before

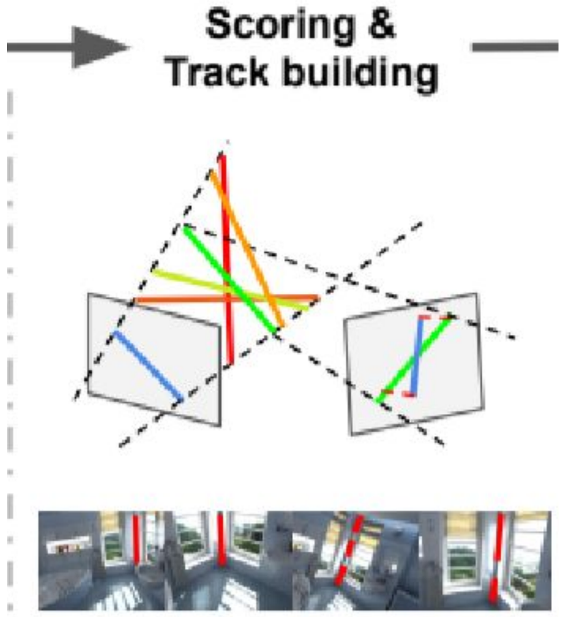
- After normalizing, the *consistency score* of one proposed line is the sum of the maximum agreements of neighboring images



Proposed Approach - LIMAP

- A consistency score threshold is set to ensure that at least 2 other images support that line proposal and it is used in the track building process

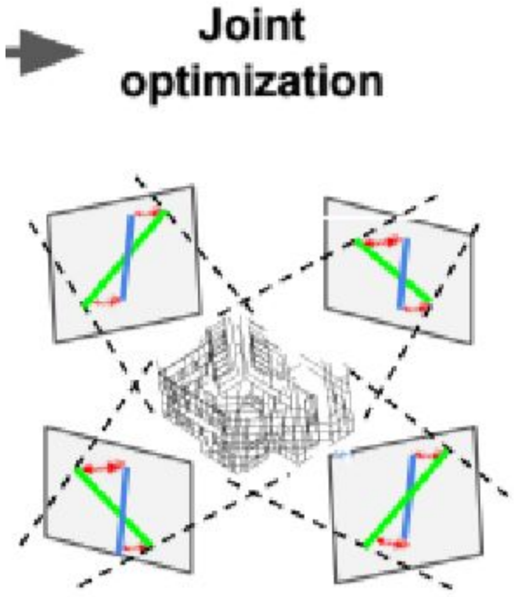
- A rough overview of track building is given below:
  - Convert into a graph: set 2D segments as nodes, if two nodes share a 3D assignment connect them with an edge
  - Prune 3D edges which don’t reach a certain 2D overlap score
  - For the remaining sections with more than 3 nodes, run PCA on the 3D endpoints to estimate the best infinite line



Liu, Shaohui, Yifan Yu, Rémi Pautrat, Marc Pollefeys, and Viktor Larsson. "3d line mapping revisited." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21445-21455. 2023.

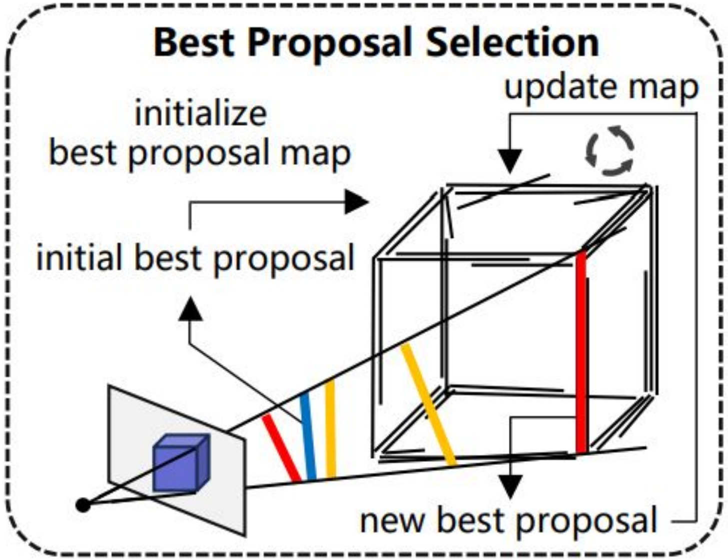
Proposed Approach - LIMAP

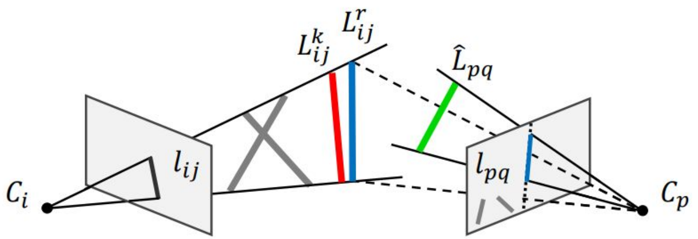
- Once we have tracks, we can do further refinement
- The energy function for this joint optimization has terms for the following (formulas omitted for brevity):
    - Line Reprojection Error
    - Point Reprojection Error
    - Soft Association Error: Number of Supports per Line Distance
        - More supports means more confidence
- Optimize on 4-DOF infinite lines
    - Plücker coordinate with Robust Huber Loss



Proposed Additions - CLMAP

- CLMAP's proposal selection has the main addition onto LIMAPs classical methodology
- The LIMAP selection function just chose the highest score according to the support relationships from nearby images. However, that could lead to inconsistencies such as missed collinearity
- CLMAP runs multiple rounds of scoring to improve consistency



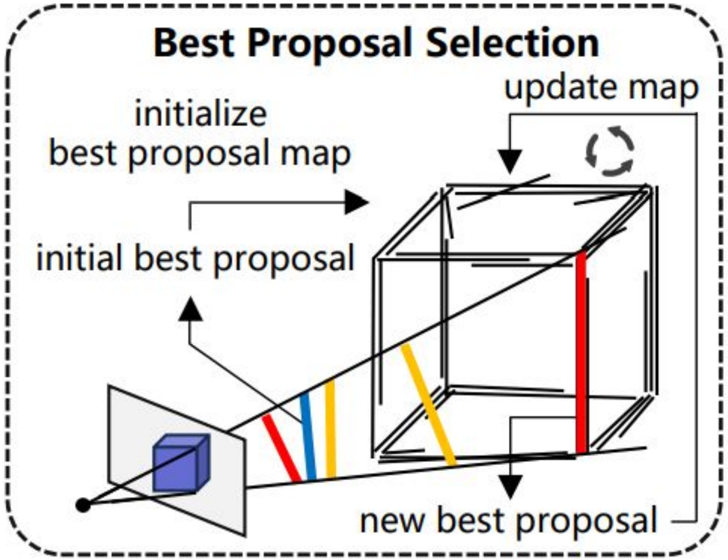


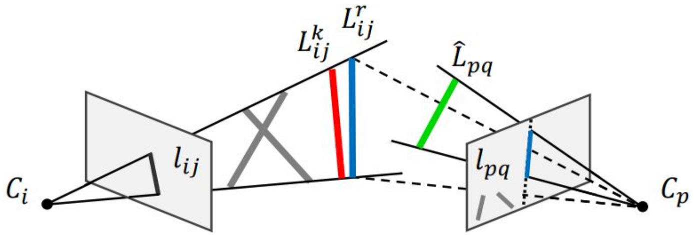
Example LIMAP Inconsistency

Bai, Xulong, Hainan Cui, and Shuhan Shen. "Consistent 3D Line Mapping." In *European Conference on Computer Vision*, pp. 57-74. Cham: Springer Nature Switzerland, 2024.

Proposed Additions - CLMAP

- The initial proposal is the same as LIMAP
- However, CLMAP then starts iterating, using the previous best proposal to score the next
- This process keeps iterating, stopping when the best proposal is no longer changing
- While the paper makes no claim of an optimal solution, this iterative process mitigates the worst of the inconsistencies



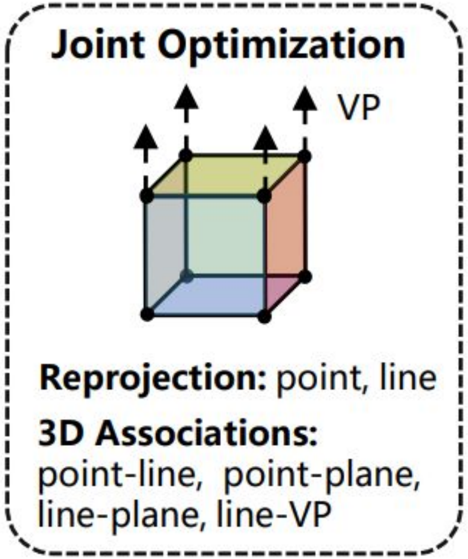


Example LIMAP Inconsistency

Bai, Xulong, Hainan Cui, and Shuhan Shen. "Consistent 3D Line Mapping." In *European Conference on Computer Vision*, pp. 57-74. Cham: Springer Nature Switzerland, 2024.

Proposed Additions - CLMAP

- The other main difference from LIMAP falls in the joint optimization section, as it considers ***line coplanarity***
- Additional terms are added to the energy function
  - Point & Plane Soft Association
  - Line & Plane Soft Association
- CLMAP also introduces adaptive weighting of each energy term depending on the initial value
- Optimization is run in largely the same way as LIMAP



Bai, Xulong, Hainan Cui, and Shuhan Shen. "Consistent 3D Line Mapping." In *European Conference on Computer Vision*, pp. 57-74. Cham: Springer Nature Switzerland, 2024.

Implementation - Running Package

- The only prepared example in the repo is for the Hypersim dataset scene 1, “ai_001_001”, which contains 100 pictures of a bathroom w/ pose information taken from the Evermotion archinteriors dataset

- For each new dataset: we need to
    - Create a configuration file
    - Create a Python function to set up and format the input data
    - Create a Python function to set up the evaluation procedures given that data’s ground truth

- This process is very slow going from a development standpoint, but understandable given the complexity of the varying 3D input data types

https://www.turbosquid.com/3d-models/archinteriors-vol-1-scenes-3d/343621

Codebase Areas for Improvement (cont.)

- SfM Dependence
  - The runner script for running LIMAP on only M3 (just using VP points, assuming SfM unavailable) is not included. Repo dependence on SfM inputs is very strict

- Visualization
  - Custom visualization tool built to overlay Ground Truth
  - Due to space constraints, the paper could not properly visualize the range of successes and failures in LIMAP. Modifications were made to the code base to better communicate accuracy and these were run for all available scenes in the dataset