From Multiview Image Curves to 3D Drawings

Supplementary Material

Anil Usumezbas
SRI International
anil.usumezbas@sri.com,

Ricardo Fabbri
State University of Rio de Janeiro
Polytechnic Institute
rfabbri@iprj.uerj.br, and

Benjamin B. Kimia
Brown University
School of Engineering
benjamin_kimia@brown.edu

SRI International, State University of Rio de Janeiro and Brown University

1 Overview of this document

This document presents additional descriptions, details and results that could not go into the main paper due to space constraints. In Section 2 we expand on the details for our algorithm, and present a complete table of system parameters we used in all our experiments. Section 3 discusses the 3D ground truth benchmarks that were used in the quantitative evaluation of our results, and details the process with which 3D curvilinear ground truth models were obtained with the aid of Blender, for both synthetic and real data. In Section 4 we present additional figures for our results as well as visual comparisons to the results of PMVS [4]. Other details can be found in Section 5.

2 Algorithm details

See Algorithm 1 for a detailed pseudo-code of our algorithm, as well as Table 1 for an index of system parameters used in our experiments.

Algorithm 1: Multiview Curve Drawing Graph (MDG) Construction

input:
- Multiview Local geometric consistency Network (MLN)
- Multiview Curve-level Consistency Network (MCCN)

output: $MDG = (J, B)$, where J encodes junctions and B curve fragment branches

Visited $\longleftarrow \emptyset$
while $Visited \neq S_{\boldsymbol{\Gamma}}$, *where* $S_{\boldsymbol{\Gamma}}$ *is the set of curves indexed by* S **do**
  for *each cluster* C *of the* MCCN, **do**
    $b_0 \longleftarrow \underset{\boldsymbol{\Gamma} \in C}{\operatorname{argmax}} \operatorname{Length}(\boldsymbol{\Gamma})$
    $\text{MDG}_0 \longleftarrow (J_0, B_0)$, where $B_0 = b_0$ and $J_0 = \emptyset$
    Given a partial curve drawing graph $\text{MDG}_i = (J_i, B_i)$, $B_i = \{b_0, \ldots, b_{n_i}\}$
    begin Construct B_{i+1}
      for *each other curve* $\boldsymbol{\Gamma} \in C$, $\boldsymbol{\Gamma} \notin$ *Visited* **do**
        for *each* s **do**
          $B_\tau(s) \doteq \{\tilde{\boldsymbol{\Gamma}} \in B_i : d(\tilde{\boldsymbol{\Gamma}}(s), \boldsymbol{\Gamma}(s)) < \tau_m\}$
          if $B_\tau(s) = \emptyset$ **then**
            `// Sample` $\boldsymbol{\Gamma}(s)$ `has no corresponding branch`
            if *no branch started* **then**
              Start new branch $\tilde{b}$
            else if *branch has started and* s *at endpoint of* $\boldsymbol{\Gamma}$ **then**
              `// The new branch is added with the correct topology, either breaking or elongating existing branches and creating junctions, according to Figure 9 in the paper`
              $B_{i+1} \longleftarrow B_i \cup \{\tilde{b}\}$
          else `// Sample` $\boldsymbol{\Gamma}(s)$ `has corresponding branches`
            begin Insert this sample into the corresponding curve
              for *each* $\tilde{\boldsymbol{\Gamma}} \in B_\tau(s)$ **do**
                Update $\tilde{\boldsymbol{\Gamma}}$ to contain $\boldsymbol{\Gamma}(s)$
                if *branch* $\tilde{b}$ *had started* **then**
                  `// The new branch is added with the correct topology, either breaking or elongating existing branches and creating junctions, according to Figure 9 in the paper`
                  $B_{i+1} \longleftarrow B_i \cup \{\tilde{b}\}$

Symbol	Description	Default Value
τ_l	minimum length of curve fragments	40 pixels
τ_e	minimum epipolar overlap	5 pixels
τ_d	maximum distance of supporting edgels	5 pixels
τ_m	sample merge threshold	0.001 world units
τ_s	minimum strength for a strong local link	3
τ_{sl}	minimum number of strong links to connect curves in the MCCN	5
τ_{prox}	used for evaluation	3D sample rate
τ_θ	maximum orientation difference of supporting edgels	10°
τ_α	minimum angle of curve segments with epipolar lines to reconstruct	10°
τ_t	minimum total inliers for a match to be considered reliable	5 edgels
τ_v	minimum inliers of a supporting view	10 edgels
τ_r	minimum best to second best ratio	1.5
b	baseline between consecutive views	20°
N_c	number of confirmation views	all views

Table 1: Table of parameters of the system.

3 Obtaining Ground Truth for Quantitative Evaluation

Quantitative evaluation of 3D models reconstructed from a sequence of images is a non-trivial task due to the difficulties involved in obtaining clean and accurate ground truth 3D models for physical objects in the world, as well as precise calibration for each of the images in the sequence. The well-known Middlebury benchmark [1] evaluates full surface reconstructions, and the ground truth 3D models are not made public; therefore it is not possible to appropriate them for quantitative evaluation of curve reconstructions. The EPFL benchmark [2] makes the ground truth 3D models publicly available, but these datasets are limited in the number of views in the image sequence, as well object types and illumination conditions captured in the scene. In our case, the difficulty is compounded by the fact that our reconstruction is a wireframe representation, whereas almost all existing ground truth for multiview stereo is for evaluating dense surface reconstruction algorithms.

Our first approach for reliable and fair evaluation of our 3D drawing algorithm is to utilize a synthetic 3D model and a rendering software to factor out calibration and reconstruction errors common among ground truth models obtained from real world objects. Here, the realistically-rendered images for this scene, Figure 1, as well as the precisely calibrated views, Figure 2, are obtained using Blender. Three different illumination conditions were rendered, and these can be mixed up to test any given algorithm's robustness under varying illumination, such as a slow sunset. This synthetic data was modeled after a real scene in Barcelona, see Figure 3 for some real imagery from this location.

To the best of our knowledge, there is no popular, publicly-available multiview stereo ground truth that is based on a precise and complex 3D model and its rendered images. We have made two versions of our Barcelona Pavilion dataset available for the evaluation of 3D reconstruction algorithms: i) The full mesh version for evaluating dense surface reconstruction algorithms, ii) 3D curve version for evaluating curvilinear models, such as the 3D drawing presented in this work, Figure 4. The latter version was obtained by a Blender-aided process of manually deleting surface meshes until only the outline of the objects remained, see Figure 5 and Figure 6.

Although the Barcelona Pavilion dataset allows for a very precise and reliable way of evaluating 3D models, a point can be made about the necessity of testing any reconstruction algorithm in the context of real world objects and real camera imagery to get a real sense of its performance. Our second approach, therefore, is to appropriate one of the many scenes present in DTU Robot Dataset [3] to the task of evaluating 3D curvilinear reconstructions. This is a significantly harder task than eliminating the surface meshes in the synthetic case, since the ground truth representation is a 3D point cloud, and no explicit distinction is made between curve outlines and surface geometry. We therefore use Blender to project the 3D point cloud ground truth for our selected scene onto several different images, correct for calibration errors to the best of our capacity, then remove all the internal surface points to end up with a subset of 3D points which are in the proximity of curved structures in the scene, see Figure 7 and Figure 8.

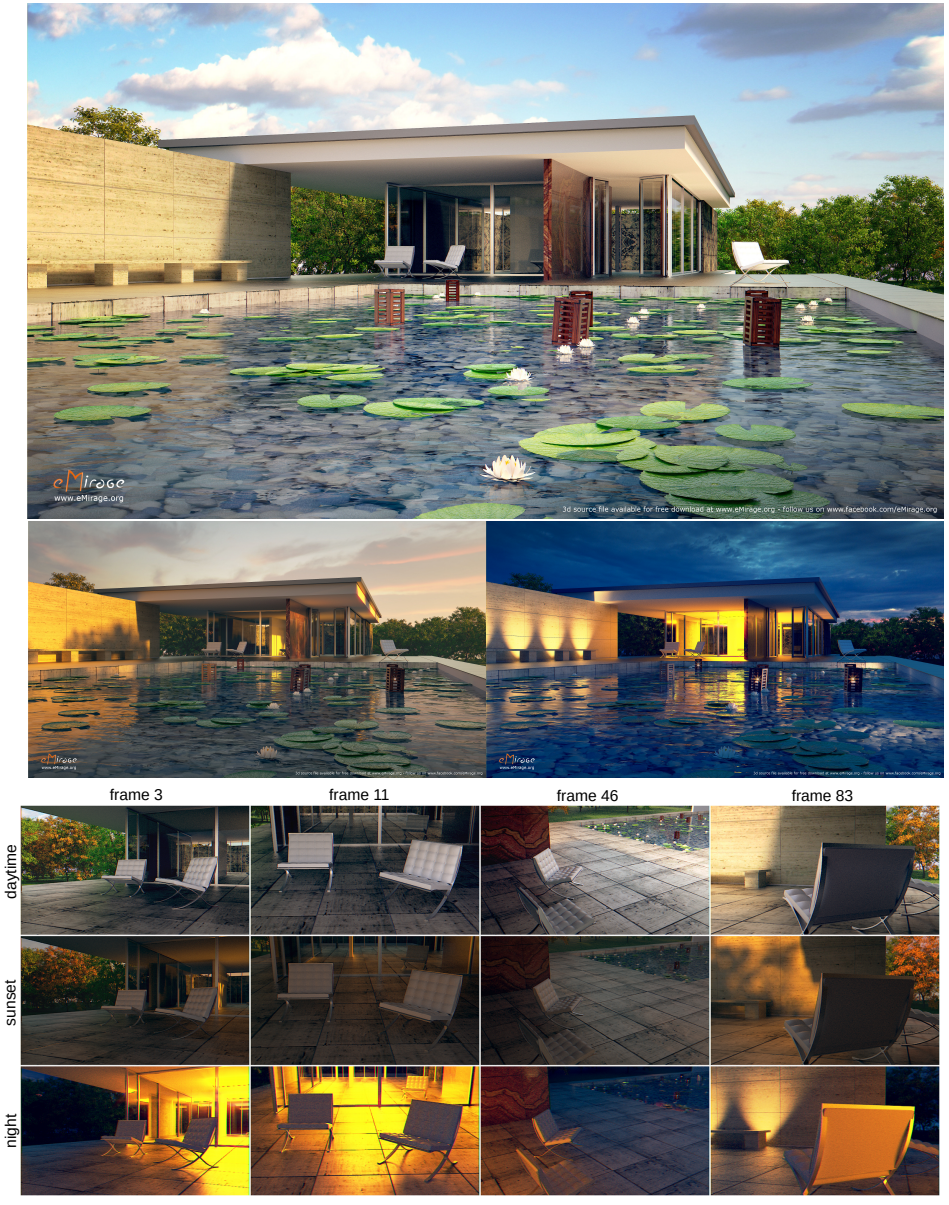


Fig. 1: Our synthetic truth modeled and rendered using Blender for the present work. The bottom images are sample frames of three different videos for different illumination conditions. A fourth sequence is also used in the experiments, mixing up frames from the three conditions.

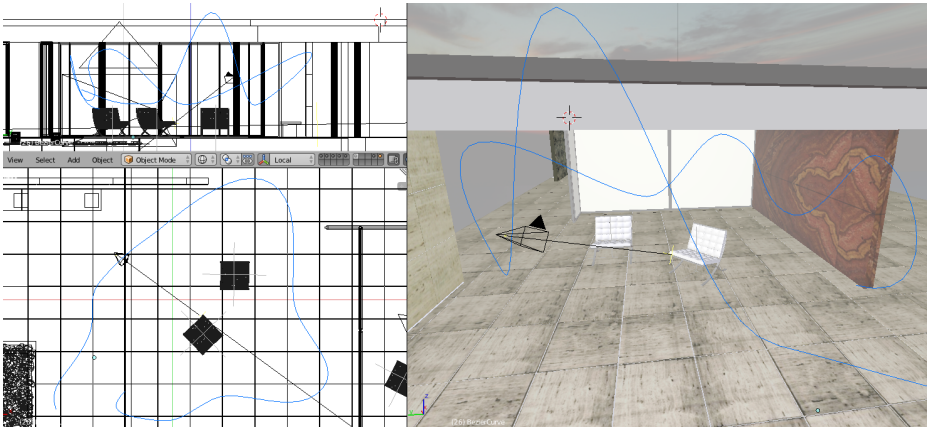


Fig. 2: Camera path (blue) used to render a realistic synthetic video and generate ground-truth cameras in our evaluation, as seen overlayed on the ground truth (left) and on a draft rendition of the scene (right).

Fig. 3: Sample of real imagery that accompany our ground truth synthetic dataset as it was modeled with high fidelity after a real scene. In the present work this was used as reference for selecting ground truth edges, but it could also be used for validating reconstruction form non-synthetic images.

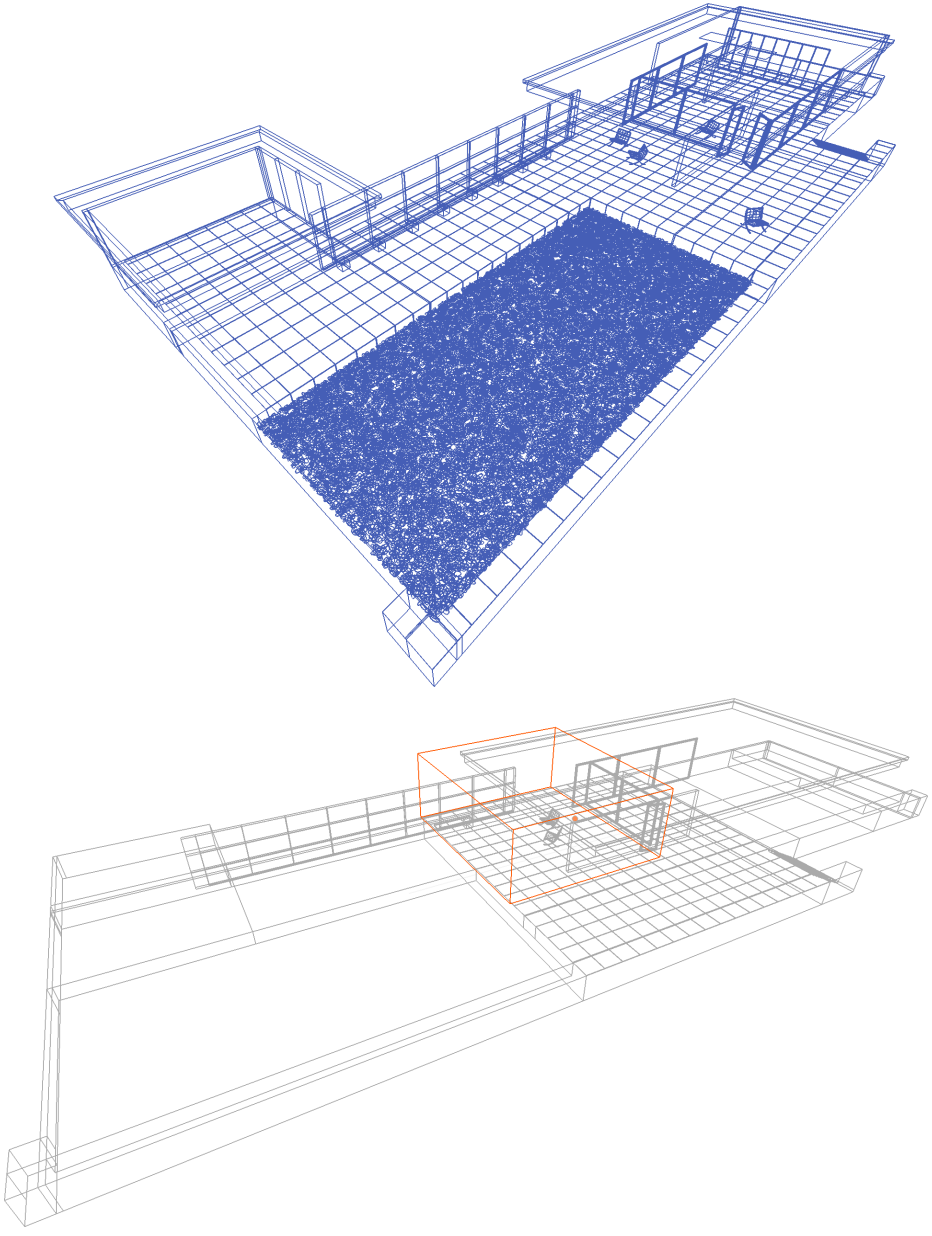


Fig. 4: The full Barcelona Pavilion synthetic ground truth (top) and the bounding box (bottom) corresponding to Figure 10 in the paper.

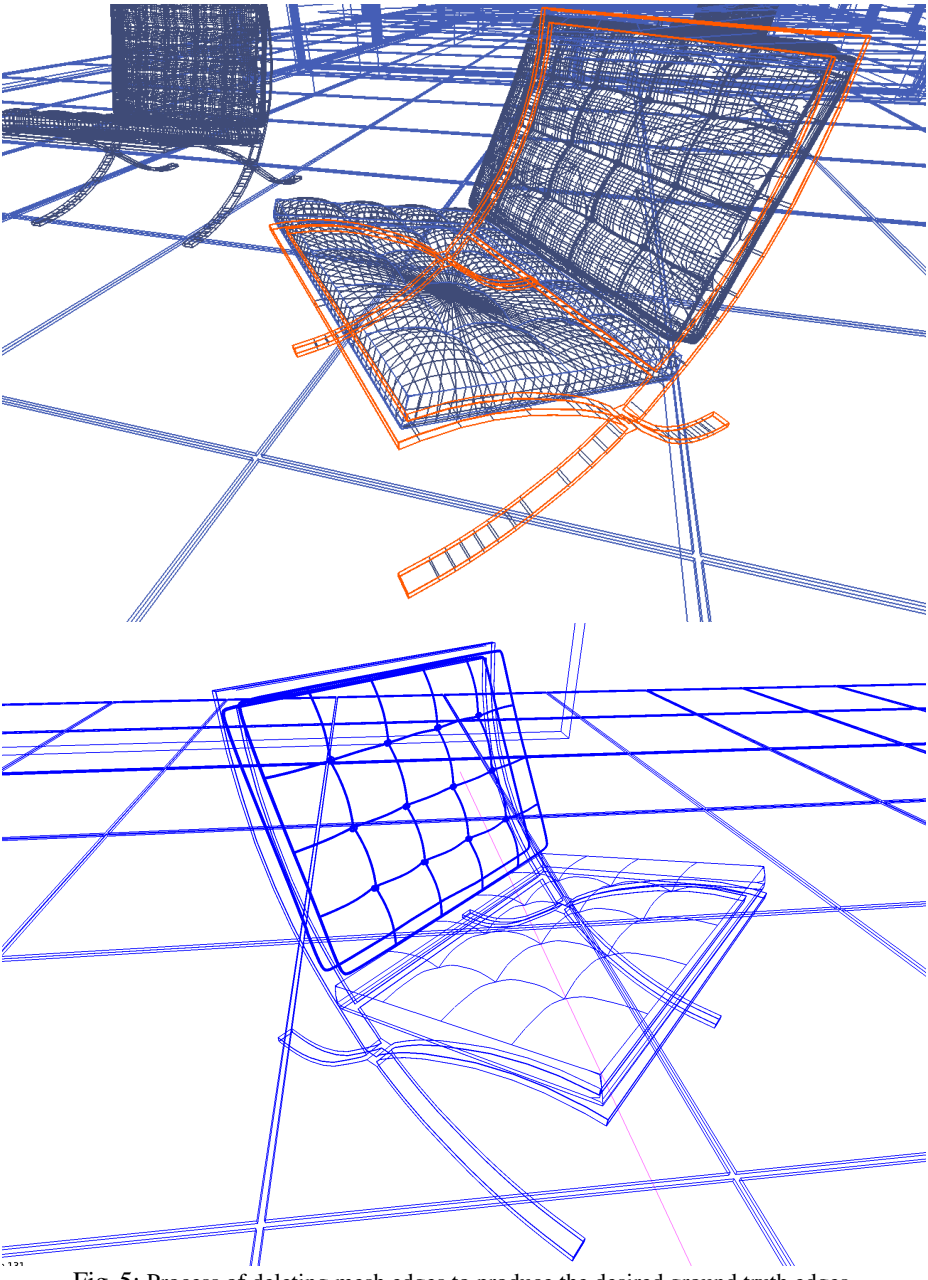


Fig. 5: Process of deleting mesh edges to produce the desired ground truth edges.

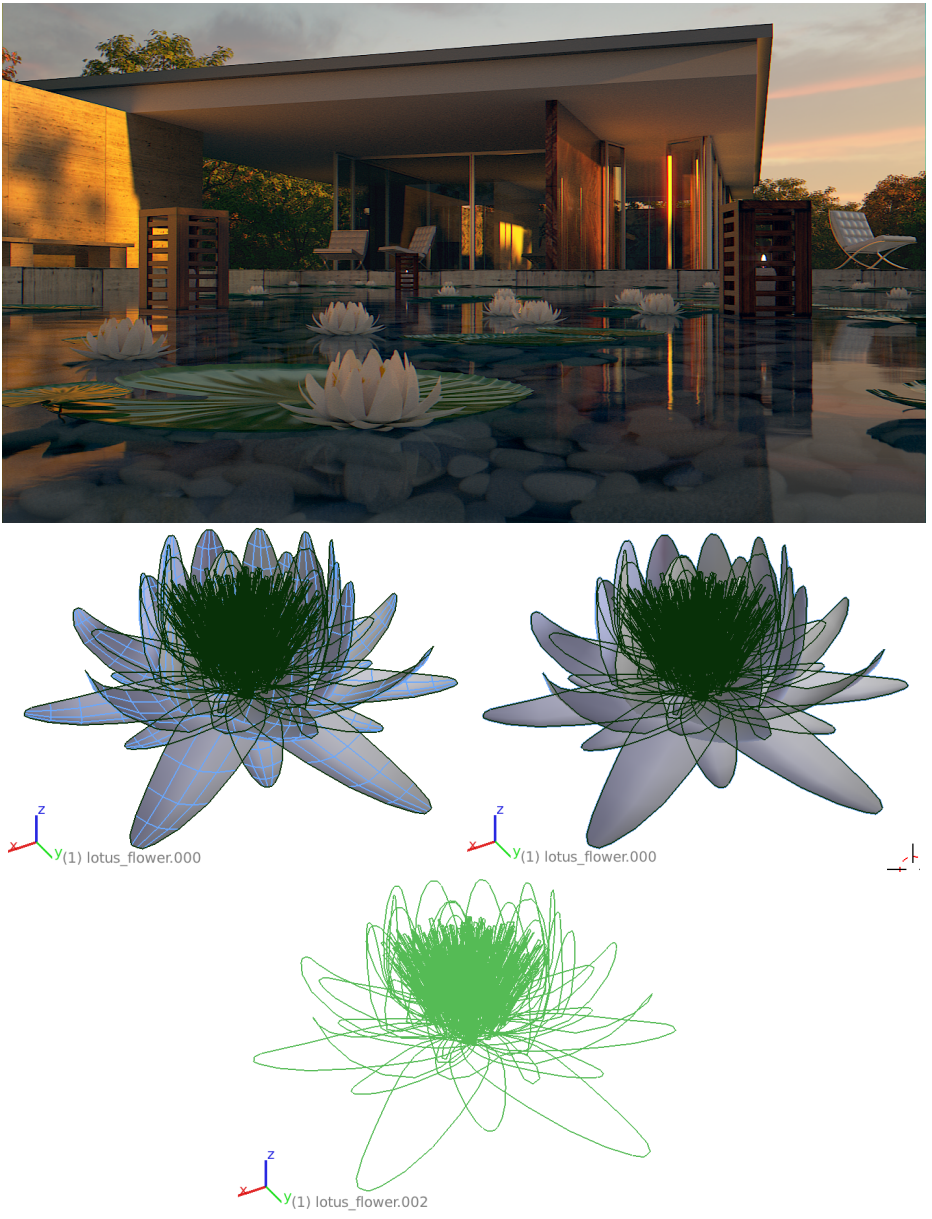


Fig. 6: Detail of our ground truth generation. Even minute objects were modeled by discarding internal mesh edges (blue).

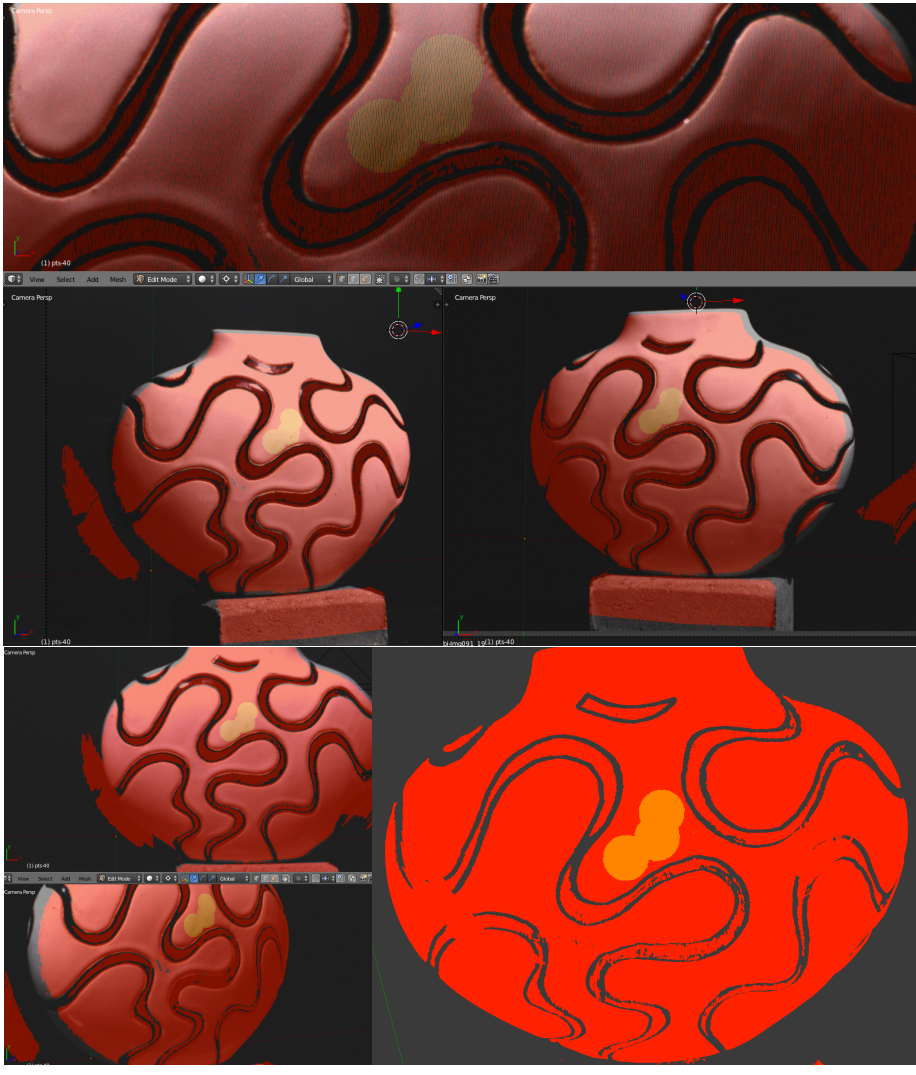


Fig. 7: Construction of the 3D groundtruth by manually deleting points of the dense point cloud from structured lighting, which is reprojected onto reference images during editing for disambiguating edges. The unstructured point cloud from structured lighting tends to suffer from over-smoothing, and lacks structure near edges, which we recover from the reference images registered using the ground truth camera parameters. Yellow shows the selected points which simultaneously show in all views and on the 3D reconstruction itself (bottom-right).

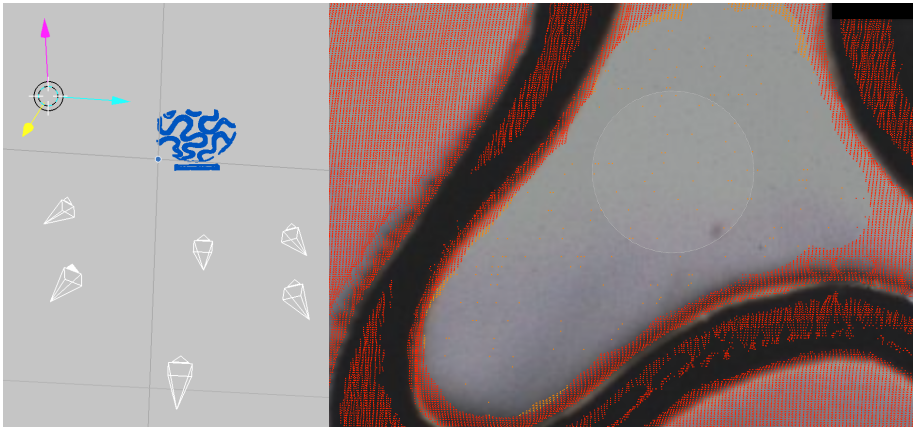


Fig. 8: Cameras used as references for sharp edges during manual construction of the ground truth from structured lighting scans (left). Zoomed-in process of registered editing of structured light scan, showing a selection disc together with deleted region, with registered image texture showing through.

4 Additional Results

In this Section we present more detailed figures for our results presented in the main paper, Figures 9, 10, 13 and 15; as well as visual comparisons to the results of PMVS [4], arguably the most well-known dense reconstruction algorithm to date, Figures 11, 12, 14 and 16

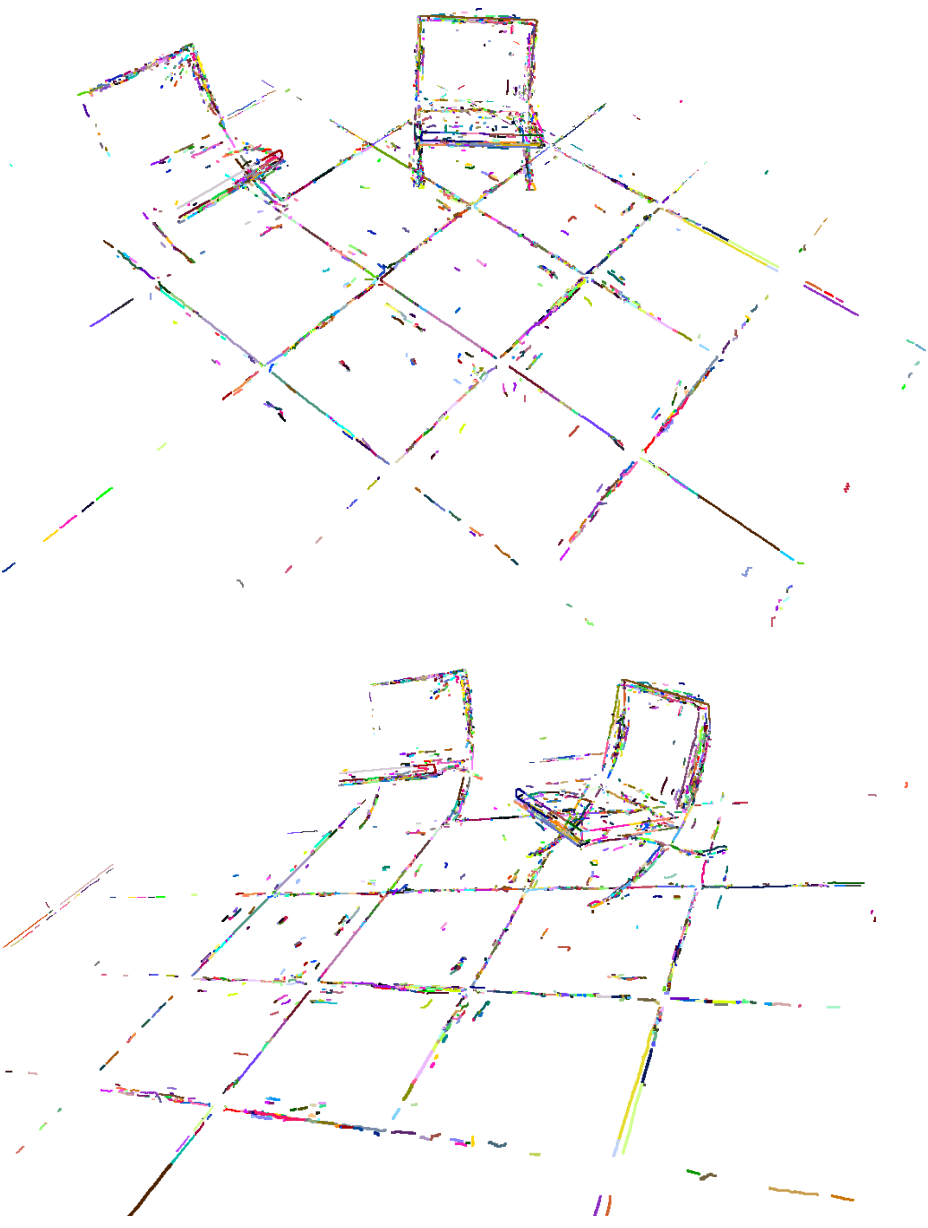


Fig. 9: Curve drawing results for the Barcelona Pavillion dataset, Mid day illumination.

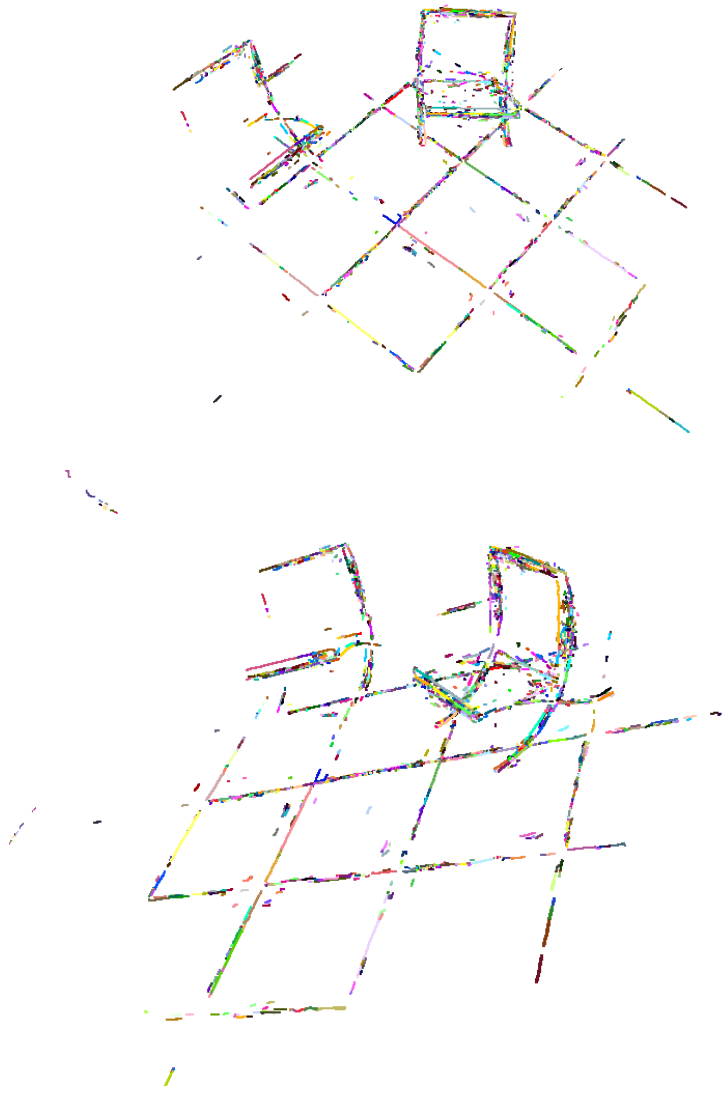


Fig. 10: Curve drawing results for the Barcelona Pavillion dataset, mixed illumination.

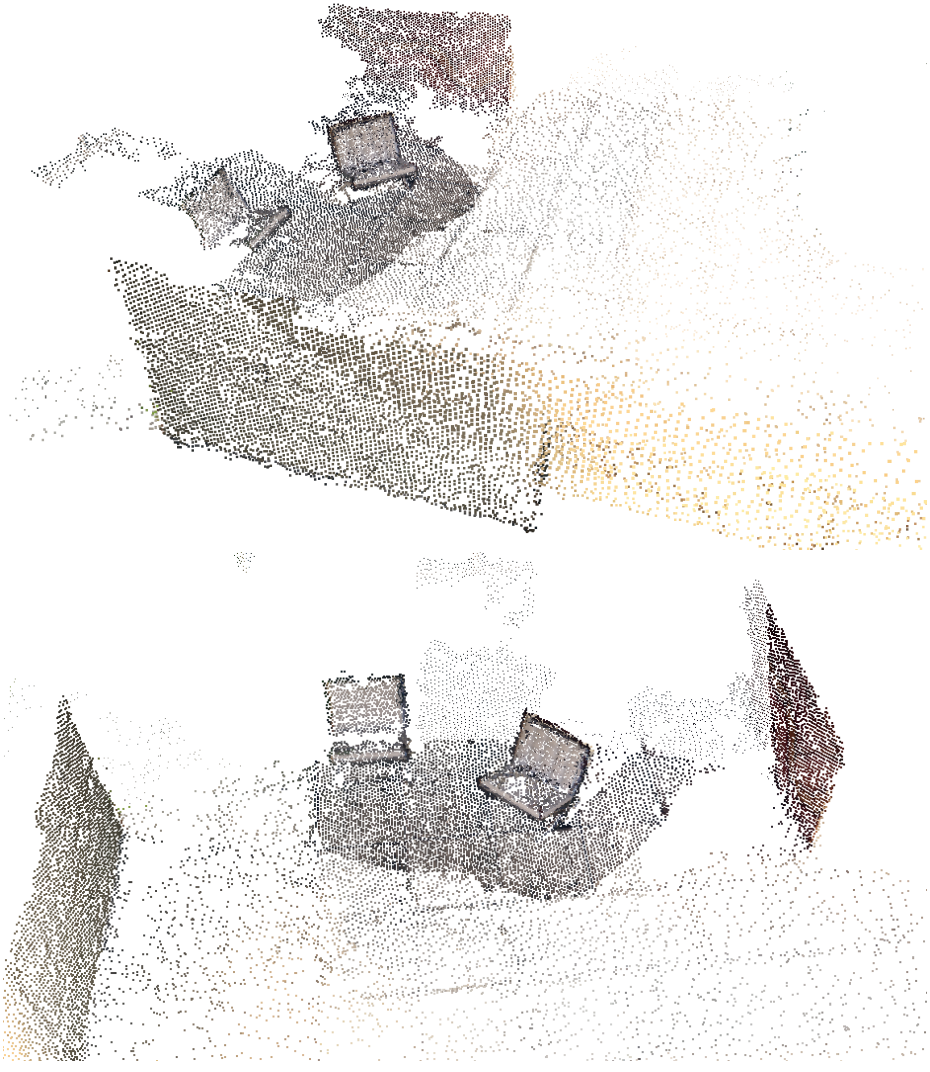


Fig. 11: Reference PMVS results for the Barcelona Pavilion dataset, mid day illumination.

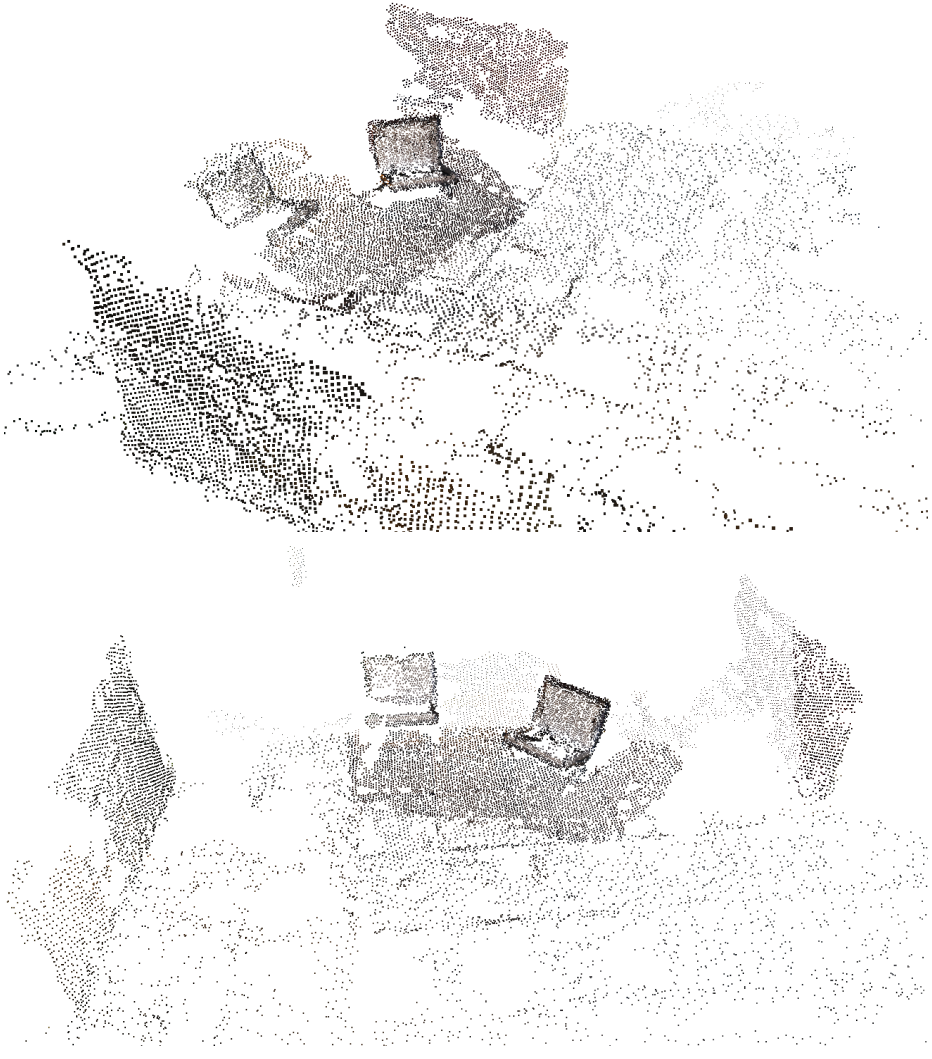


Fig. 12: Reference PMVS results for the Barcelona Pavilion dataset, mixed illumination.

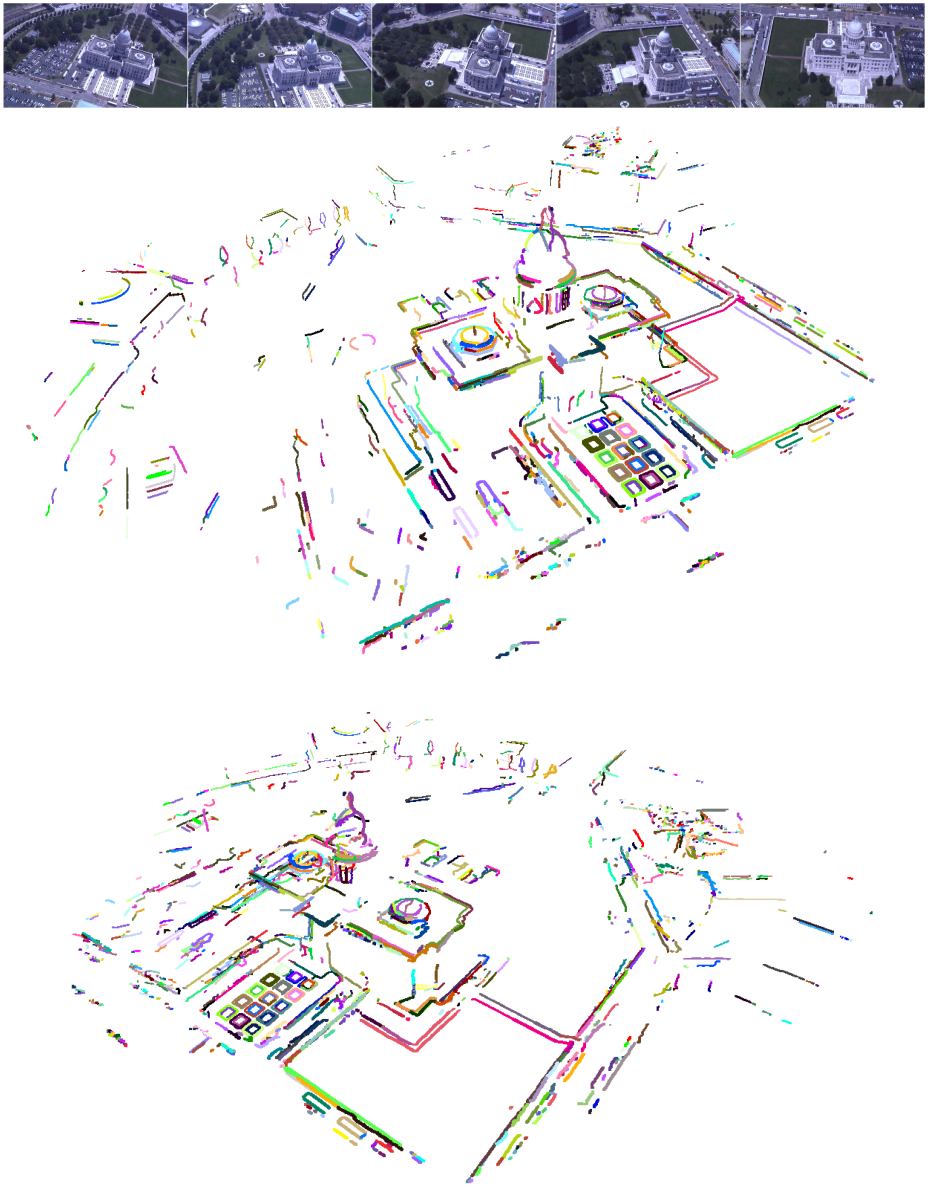


Fig. 13: Curve drawing results for the Capitol High dataset.

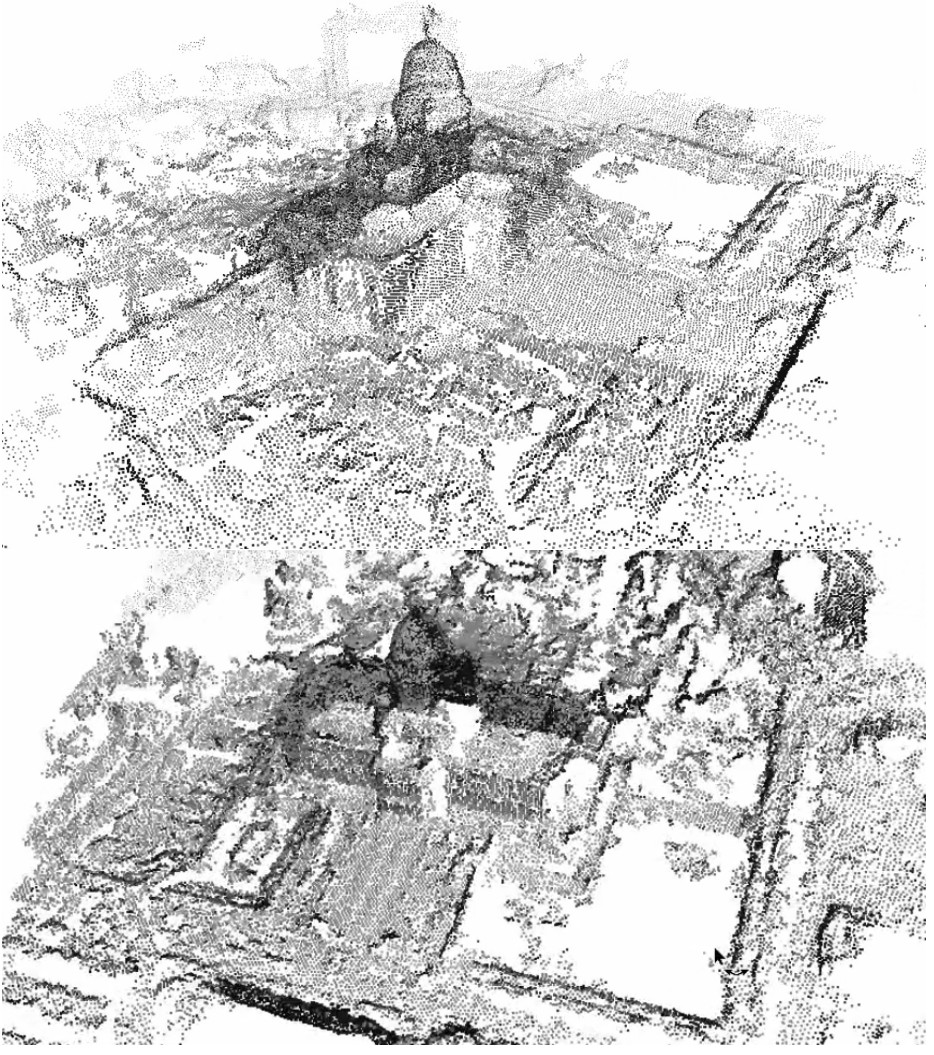
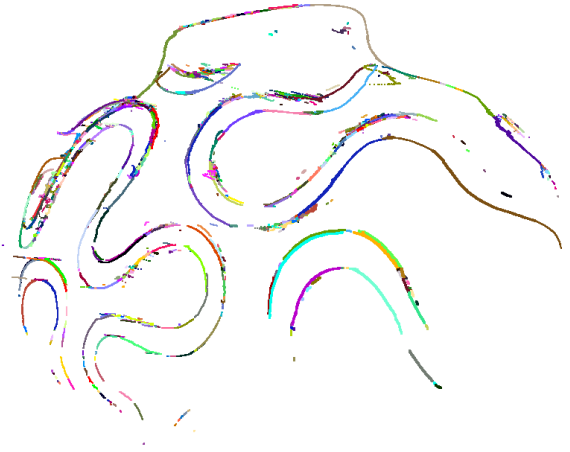


Fig. 14: Reference PMVS results for the Capitol High dataset.



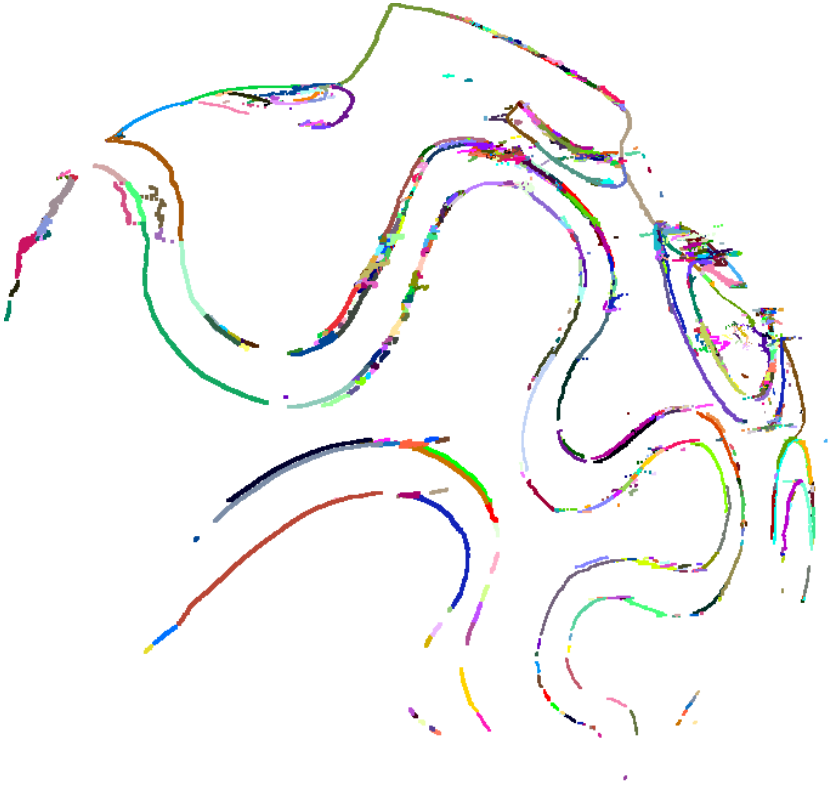


Fig. 16: Reference PMVS results for the Vase dataset.

5 Additional Details

5.1 Language

We used C++ to implement the base system up to the enhanced curve sketch, using widely-available open source libraries, such as Boost (www.boost.org), and VXL (vxl.sourceforge.net). The curve drawing stage is implemented in Matlab. The experiments ran on Linux but the code is very portable.

5.2 Addditional Supplementary Material

Other than this pdf document, the supplementary materials package contains, among others: i) Two mp4 videos comparing reconstructions of Curve Sketch, Enhanced Curve Sketch, 3D Drawing and PMVS on Amsterdam House Dataset, and ii) A .PLY file which contains the 3D Drawing results on the Amsterdam House Dataset. You can view this model in MeshLab or any other software that supports .PLY file format.

5.3 Availability

The C++ and Matlab source code are available to the public at multiview-3d-drawing.sourceforge.net, as well as the ground truth datasets and additional supplementary material.

5.4 Runtime

On a single core, our algorithm takes several hours to process images into 3D Drawings. This profiling includes edge detection, curve linking, automated calibration of cameras, hypothesis-testing and graph organization. The algorithm speed can be further increased by parallel processing and GPU implementation of certain high-demand processes.

References

1. S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms *CVPR'06*, pp. 519–528. 4
2. C. Strecha, W. von Hansen, L. Van Gool, P. Fua, and U. Thoennessen On Benchmarking Camera Calibration and Multi-View Stereo for High Resolution Imagery *CVPR'08* 4
3. H. Aanæs, A.L. Dahl, and K. Pedersen Interesting Interest Points *IJCV*, 2012. 4
4. Y. Furukawa and J. Ponce. Accurate, dense, and robust multi-view stereopsis. *CVPR'07*. 1, 13