
Reconstructing 3D scenes from multiple views has made impressive strides in recent years, chiefly by correlating isolated feature points, intensity patterns, or curvilinear structures. In the general setting – without controlled acquisition, abundant texture, curves and surfaces following specific models or limiting scene complexity – most methods produce unorganized point clouds, meshes, or voxel representations, with some exceptions producing unorganized clouds of 3D curve fragments or surface patches. Ideally, many applications require structured representations of curves, surfaces and their spatial relationships. The work presented in this thesis is a step in this direction by formulating an approach that combines 2D image curves into a collection of 3D curves, with topological connectivity between them represented as a 3D graph. This results in a 3D drawing, which is complementary to surface representations in the same sense as a 3D scaffold complements a tent taut over it. Furthermore, this 3D graph serves as a boundary condition for a novel dense surface reconstruction algorithm based on occlusion reasoning and lofting, a computer graphics method for curve interpolation. Because reconstructed surface representations are constrained by the 3D drawing acting like a scaffold to hang on the computed representations, the resulting surfaces are consistent with the curve boundaries of the 3D drawing, and do not smooth over important features such as ridges or corners. The 3D models obtained this way are capable of addressing some of the shortcomings of the multiview stereo state-of-the-art. We evaluate all our results against ground truth on synthetic and real datasets.
Dense and Accurate Multiview Stereo Reconstruction
Using Differential Geometry of Curves/Surfaces, Lofting
and Occlusion Reasoning

by
Anil Usumezbas
B.Sc., Electronics Engineering, Sabanci University, 2008

A dissertation submitted in partial fulfillment of the
requirements for the Degree of Doctor of Philosophy
in the School of Engineering at Brown University

Providence, Rhode Island
May 2017
© Copyright 2017 by Anil Usumezbas
This dissertation by Anil Usumezbas is accepted in its present form by the School of Engineering as satisfying the dissertation requirement for the degree of Doctor of Philosophy.

Date ___________  __________________________________________
Professor Benjamin B Kimia, Advisor

Recommended to the Graduate Council

Date ___________  __________________________________________
Professor Gabriel Taubin, Reader

Date ___________  __________________________________________
Professor Ricardo Fabbri, Reader

Approved by the Graduate Council

Date ___________  __________________________________________
Professor Lawrence Larson
Dean of the Graduate School
Vita

Anil Usumezbas began his bachelor’s studies in Electronics Engineering at the Sabanci University, Istanbul in 2004 and received his degree in May, 2008. During this time he also worked first as an Intern, then as a Part-time Design Engineer in VESTEK R&D, Istanbul, where he focused on HDL implementations of vision enhancement algorithms for televisions. Upon graduation, he joined Laboratory for Engineering and Man-Machine Sciences (LEMS) at Brown University for a Ph.D. degree in School of Engineering, focusing on 3D computer vision. En route to his Ph.D. degree, he completed his master’s degree in Applied Mathematics in May 2012. He worked as a Student Associate at SRI International, Princeton where he started employment as a Computer Scientist upon the completion of his Ph.D. studies.
Acknowledgements

Firstly, I would like to thank my advisor, Prof. Benjamin B. Kimia, for providing me with the opportunity to work on automated 3D modeling - the exact field of computer vision that attracted me to my Ph.D. study in the first place. He has imparted on me a unique and exciting vision for solving this problem and pushing our field forward with novel ideas. He entrusted me with dictating the direction of projects and given me a level of academic freedom that few graduate students enjoy. I would like to express my gratitude to my thesis readers, Prof. Gabriel Taubin and Prof. Ricardo Fabbri, for taking the time to review and comment on my research work. I would also like to express my deepest appreciation for the close research collaboration that I’ve enjoyed with Prof. Ricardo Fabbri during my thesis work.

I would like to thank my colleagues at Brown University who have helped me numerous times along the way: Maruthi Narayanan, Brandon Mayer, Amir Tamrakar, Ibrahim Eden, Eduardo Almeida, Fatih Calakli, Ali Osman Ulusoy, Firat Kalaycilar, Octavian Biris and Yuliang Guo. My time at Brown would certainly have been very dull and unfulfilling had it not been for my great friends, and so I must thank Jessica Jalbert, Sinan Karaveli, Brandon Mayer, Avram Levi, Dhiraj Catoor, Ilke Kaya, Rassin Grantab, Silja Maehl, Elif Yamangil, Elif Rodriguez, Gavril Bilev, Tatiana Vasilieva, Alexey Rybakov, Jen Corbett, Marco Donato and countless others.

I also owe my most sincere gratitude for my colleagues Ryan Villamil, Andy Coppock, Josh Chattin, Nhon Trinh as well as my managers Raia Hadsell, Bogdan Matei, and Supun
Samarasekera at SRI International, where I worked as a Student Associate during my Ph.D. years and where I am currently employed. They have had a tremendous positively impact on my Ph.D. by helping me overcome a lot of the roadblocks regarding my thesis research and helping me maintain my motivation to conclude my Ph.D. studies in every possible way.

Last and absolutely-not-the-least, my parents, my brother and all my extended family have always been very supportive of me and what I chose to do with my life. None of what I’ve achieved would’ve been possible without their selfless devotion to my success.
Contents

List of Tables x

List of Figures xi

1 Introduction and Overview 1

2 From Multiview Image Curves to 3D Drawings 10
   2.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10
   2.2 Enhanced 3D Curve Sketch . . . . . . . . . . . . . . . . . . . . . . . . . . 12
   2.3 From 3D Curve Sketch to 3D Drawing . . . . . . . . . . . . . . . . . . . . 15
   2.4 Constructing Synthetic Ground Truth Models for Quantitative Evaluation . 26
   2.5 Experiments and Evaluation . . . . . . . . . . . . . . . . . . . . . . . . . 29

3 The Surfacing of Multiview 3D Drawings via Lofting and Occlusion Reasoning 38
   3.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 38
   3.2 Bringing Lofting Into Multiview Stereo . . . . . . . . . . . . . . . . . . . . 42
   3.3 Automated Multiview Reconstruction Using Lofting . . . . . . . . . . . . 46
   3.4 Reorganization of Input Curve Graph/Network Using 1st-Order Differen-
      tial Geometry . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 55
   3.5 Experiments and Results . . . . . . . . . . . . . . . . . . . . . . . . . . . 60
4 Conclusions

Bibliography

* Parts of this thesis have appeared in the following peer-reviewed publications:


List of Tables

2.1 Table of parameters for the curve drawing system. . . . . . . . . . . . . . 31
3.1 Table of parameters for the lofting system. . . . . . . . . . . . . . . . . . 61
List of Figures

1.1 **3D drawings for urban planning and industrial design.** A process from professional practice for communicating solution concepts with a blend of computer and handcrafted renderings [Leggitt (2015), Yee (2012)]. New designs are often based off real object references, mockups or massing models for selecting viewpoints and rough shapes. These can be modeled *manually* in, *e.g.*, Google Sketchup (top-left), in some cases from reference imagery. The desired 2D views are rendered and *manually* traced into a reference curve sketch (center-left, bottom-left) easily modifiable to the designer’s vision. The stylized drawings to be presented to a client are often produced by *manually* tracing and painting over the reference sketch (right). Curves and surfaces can be used to generate reference 3D curve drawings from video footage of the real site for urban planning, saving manual interaction, providing initial information such as rough dimensions, and aiding the selection of pose, editing and tracing. The condensed 3D curve drawings make room for the artist to overlay his concept and harness imagery as a clean reference, clear from details to be redesigned. . . . . . 4
1.2 A series of visualizations for a robot-built 3D map, taken from Hadsell et al. (2012). The much needed semantic information and organization of 3D data is lacking, so the human agents interact with a series of color-coded 3D maps instead for a more intuitive interface. Such applications would greatly benefit from a richer representation with semantically significant features explicitly outlined.

2.1 The 3D curve drawing transforms the curves in calibrated views of a scene (top 2 rows) into a “3D drawing” (bottom 2 rows) – a graph of 3D curves meeting at junctions. Each curve is shown in a different color.

2.2 (a) Due to a lack of consistency in grouping of edges at the image level, a correct 3D curve reconstruction, shown here in blue, can be erroneously grouped with an erroneous reconstruction, shown here in red, leading to partially correct reconstructions. When such a 3D curve is projected in its entirety to a number of image views, we only expect the correct portion to gather sustained image evidence, which argues for a hypothesis verification method that can distinguish between supported segments and outlier segments; (b) An incorrect hypothesis can at times coincidentally gather an extremely high degree of support from a limited set of views. The red 3D line shown here might be an erroneous hypothesis, but because parallel linear structures are common in man-made environments, such an incorrect hypothesis often gathers coincidental strong support from a particular view or two. Our hypothesis verification approach is able to handle such cases by requiring explicit support from a minimum number of viewpoints simultaneously.
2.3 (a) Redundant 3D curve reconstructions (orange, green and blue) can arise from a single 2D image curve in the primary hypothesis view. If the redundant curves are put in one-to-one correspondence and averaged, the resulting curve is shown in (b) in purple. Our robust averaging approach, on the other hand, is able to get rid of that bump by eliminating outlier segments, producing the purple curve shown in (c). ........................................ 15

2.4 A visual comparison of: (top) the curve sketch results of Fabbri and Kimia (2010) run at high-recall settings, with (bottom) the results of our enhanced curve sketch algorithm presented in Section 2.2. Notice the significant reduction in both outliers and duplicate reconstructions, without sacrificing coverage. .................................................. 16

2.5 The four bottlenecks of Fig. 2.9 are resolved by integration of information/cues from all views. (a) The shared supporting edges, which are marked with circles, create the purple links between the corresponding samples of the 3D curves. These purple bonds will then be used to pull the redundant segments together and reorganize the 3D model into a clean 3D graph. Observe how the determination of common image support can identify portions of the green and blue curves as identical while differentiating the red one as distinct. A real example for a bundle of related curves is shown in (b) and the links among their edges in (c). ................................. 18

2.6 The correspondence between 3D edge samples is skewed along a curve, a direct indication that these links cannot be used as-is when averaging and fusing redundant curve reconstructions. Instead, each point is assumed to be in correspondence with the point closest to it on another overlapping curve, during the iterative averaging step. Observe that corrections can be partial along related curves. ........................................ 20
2.7 (a) A schematic of sample correspondence along two related 3D curves, showing skewed correspondences that may not be one-to-one. (b) A sketch of how two curves are integrated. Bottom 2 rows show real examples where the left column is the result of enhanced curve sketch, middle column shows the robust averaging output, and the right column shows the curve drawing result.

2.8 The complete set of merging primitives, which were systematically worked out to cover all possible merging topologies between a pair of curves whose overlap regions are computed beforehand. We claim that any configuration of overlap between two curves can be broken down into a series of these primitives along the length of one of the curves. The 5th primitive is representative of a bridge situation, where the connection at either end of the yellow curve can be any one of the first four cases shown, and 6th primitive is representative of a situation where only one end of the yellow curve connects to multiple existing curves, but not necessarily just two.

2.9 (a) The four main issues with the enhanced curve sketch: (b) localization errors along the camera principal axis, which cause loss in accuracy if not corrected, (c) redundant reconstructions due to a lack of integration across different views, (d) the reconstruction of a single long curve as multiple, disconnected (but perhaps overlapping) short curve segments, and (e) the lack of connectivity among distinct 3D curves which naturally form junctions. (f-i) shows the 3D drawing reconstructed from this enhanced curve sketch, as described in Section 2.3. Observe how each of the four bottlenecks have been resolved. Additional results are evaluated visually and quantitatively, and are reported in Section 2.5.
2.10 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.

2.11 The full Barcelona Pavilion synthetic ground truth (top) and the bounding box that we have limited our quantitative evaluation to (bottom).

2.12 Process of deleting mesh edges to produce the desired ground truth edges.

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

2.14 Our publicly-available synthetic (left and top-right) and real (bottom-right) 3D ground truths modeled and rendered using Blender for the present work.

2.15 The 3D drawing results, together with PMVS results, on the Brown Capitol dataset.

2.16 The 3D drawing results, together with PMVS results, on the Barcelona pavilion dataset.

2.17 The 3D drawing results, together with PMVS results, on the Vase dataset.

2.18 Precision-recall curves for quantitative evaluation of 3D curve drawing algorithm:
   (a) Curve sketch, enhanced curve sketch and curve drawing results are compared on Barcelona Pavilion dataset with afternoon rendering, showing significant improvements in reconstruction quality; (b) A comparison of 3D curve drawing results on fixed and varying illumination version of Barcelona Pavilion dataset proves that 3D drawing quality does not get adversely affected by varying illumination; (c) 3D drawing improves reconstruction quality by a large margin in Vase dataset, which consists of images of a real object under slight illumination variation.

3.1 The proposed approach transforms a 3D curve drawing (top) obtained from a fully calibrated set of 27 views, into a collection of dense surface patches (bottom) obtained via lofting and occlusion reasoning.
3.2 From open and closed curves (left), lofting produces smooth surfaces (right).

3.3 Application of subdivision resulting in a high-poly surface (manually marked hard edges in red) Lavou et al. (2006).

3.4 Three possible outcomes of our lofting approach are illustrated using simple, analytical 3D shapes: a set of stationary boundary curves accurately capturing the geometry of a surface patch produces an accurate reconstruction via lofting (a); if the geometry of a surface patch is only roughly approximated due to non-stationary boundaries (b) or unrelated stationary boundaries (c), a simplified approximate shape is generated via lofting. Left column shows the ground truth shapes together with stationary boundary curves marked in red. Right column shows the loft surfaces obtained by running our approach on the red curves. Notice that even when geometric errors are present, a coarse-level geometry of the surface is still reconstructed.

3.5 Quadrangulation in lofting; depending on the configuration of special interior vertices on a chain, one of these four edits are applied to obtain a base mesh topology [Schaefer et al. (2004)].

3.6 A schematic of a simple shapes where a surface patch (green) is represented by a pair of curves (red); in the case of closed curves, a pair is not necessary.
3.7 There is an inherent ambiguity in reconstructing a surface from two curve fragments arising from which endpoints are paired (top row vs. bottom row). When two curve fragments do belong to a veridical surface, one of the two reconstructions generally has much lower average Gaussian curvature than the other and this is a cue as to which one is veridical. When the pairing of curve fragments is incorrect in that no surface exists between them, both reconstructs have high average Gaussian curvature, a cue to remove outliers.

3.8 Some example loft surfaces of various geometries that our reconstruction algorithm generates.

3.9 A 3D surface patch $S$ occludes all 3D curve fragments that lie behind it. Thus, the 3D curve fragments between $\Gamma_1$ and $\Gamma_4$ are partially obstructed so that only portions between $(\Gamma_1, \Gamma_2)$ and $(\Gamma_3, \Gamma_4)$ are visible as $(\gamma_1, \gamma_2)$ and $(\gamma_3, \gamma_4)$ in the image. The projections of $(\Gamma_2, \Gamma_3)$ should have no edge evidence in the image. On the other hand, the 3D curve fragments $(\Gamma_5, \Gamma_6)$ is fully unoccluded and edge evidence for it is expected. The presence of edge evidence in the portion $(\gamma_2, \gamma_3)$ is grounds for invalidating the 3D surface hypothesis $S$.

3.10 A visual illustration of our dense surface reconstruction pipeline.

3.11 Examples to surface hypotheses being confirmed by the confirmation views shown in this here. Left column:Projected surface hypothesis is shown in green, projected curve drawing is shown in blue and occluded segments are shown in purple. Right column: Same surface and occluded segments are shown with image edges in blue. Notice the lack of any edge presence whatsoever around most of the purple segments, which is clear indication of occlusion consistency between the images and the hypothesis surface.
3.12 An example outlier surface hypothesis ruled out by detected edge structures. Left column: projected surface hypothesis is shown in green, projected curve drawing is shown in blue and occluded segments are shown in purple. Right column: Same surface and occluded segments are shown with image edges in blue. Notice how most of the purple segments are barely visible from all the edges that match in both location and orientation.

3.13 (a) Overgrouping of two curve fragments $C_2$ and $C_3$ into $C_4$ can lead to nonsensical lofting results in the pair $(C_1, C_4)$ in contrast to the close-to-veridical results of lofting $(C_1, C_2)$; (b) lofting is sensitive to loop-like noise or excessive perturbations; (c) lofting with overfragmented curves produces suboptimal lofting results as well as redundant surface proposals, and leads to a combinatorial increase in the number of lofting applications and post-processing.

3.14 The original input 3d curve drawing (top row), which is the direct output of the 3D curve drawing approach, the result of our reorganization algorithm before breaking sharp corners (middle row), and after the sharp corners are broken (bottom row). The level of granularity displayed in the last row is the most appropriate for our lofting approach, as most surfaces are bounded by entire curves rather than subsegments.

3.15 The left column represents four views of the PMVS reconstruction results on the Amsterdam House Dataset. Observe the wide gaps on homogeneous surfaces. The right column represents the results of our algorithm from the same views, obtained from a set of mere 27 curve fragments and without using appearance. Note that the PMVS gaps are filled in our results. Our algorithm errs in reconstructing the back of the can as a flat surface. This can easily be corrected via integration of appearance cues in the reconstruction process.
3.16 Visual results for the Barcelona Pavilion sequence. Left column: Result of our lofting-based reconstruction algorithm. Right column: PMVS results on the same sequence. Notice that we have no coverage of the floor surface due to the lack of occlusion information, but that our approach is better at capturing fine geometric detail and features a lot less holes on the surfaces of the primary objects, i.e., the chairs.

3.17 (a) The precision-recall curves for Amsterdam House Dataset, corresponding to post hypothesis-formation surfaces (green), confirmed surface (blue), and confirmed surfaces after occlusion-based cleanup (red). These results provide quantitative proof for the necessity of all steps in our reconstruction algorithm; (b) The precision-recall curve for Barcelona Pavilion Dataset, evaluating the geometric accuracy of the entire pipeline.
Chapter 1

Introduction and Overview

The automated 3D reconstruction of *general* scenes from multiple views obtained using conventional cameras, under uncontrolled acquisition, is a paramount goal of computer vision, ambitious even by modern standards. While a fully complete working system addressing all the underlying challenges is beyond current technology, significant progress has been made in the past few years using approaches that fall into three broad classes, depending on whether one focuses on correlating isolated points, surface patches, or curvilinear structures across views, as described below.

A vast majority of multiview reconstruction methods rely on correlating isolated interest points across views to produce an unorganized 3D cloud of points. The *interest-point-based approach* has been highly successful in reconstructing large-scale scenes with *texture-rich images*, in systems such as in Phototourism and recent large-scale 3D reconstruction work by Heinly et al. (2015), Pollefeys et al. (2004), Agarwal et al. (2009), Diskin and Asari (2015). Despite their manifest usefulness, these methods generally cannot represent smooth, textureless regions (due to the sparsity of interest points in image regions with homogeneous appearance), or regions that change appearance drastically across views. This limits their applicability, especially in man-made environments [Simoes et al. (2012)], and objects such as cars [Shinozuka and Saito (2014)], non-Lambertian surfaces
such as that of the sea, appearance variation due to changing weather [Baatz et al. (2012)], and wide baseline [Moreels and Perona (2007)].

A natural extension to this idea is to match intensity patterns across views using multiview stereo, producing denser point clouds or mesh reconstructions. **Dense multi-view stereo** produces detailed 3D reconstructions of objects imaged by a large number of precisely calibrated cameras, as in Furukawa and Ponce (2007), Habbecke and Kobbelt (2007), Esteban and Schmitt (2004), Goesele et al. (2007), Seitz et al. (2006), Calakli et al. (2012), Restrepo et al. (2014). For general, complex scenes with various kinds of objects and surface properties, this approach has shown most promise towards obtaining an accurate and dense 3D model of a given scene. Homogeneous areas, such as walls of a corridor, repeated texture, and areas with view-dependent intensities create challenges for these methods.

Another class of methods rely on optimizing either voxel occupancy [Miller et al. (2011)] or a global inverse-rendering function that optimizes for lighting and a single, dense surface simultaneously, given an initial guess [Jin et al. (2008), Wu et al. (2011)]. These approaches have a number of drawbacks: i) They often need to be initialized by the visual hull of the object, a good initial guess or a bounded 3D voxel volume, all of which can create practical limitations for general scenery; ii) they typically make strong assumptions about the number of objects in the scene or their attributes (e.g. man-made structures, planar surfaces, single object of uniform material etc.) iii) they tend to smooth out semantic details of the scene, like ridges and corners; iv) they sometimes require controlled acquisition, known illumination [Yoon et al. (2010)] or require camera calibration on an intensity level [Wu et al. (2011)].

A smaller number of techniques correlate and reconstruct image **curvilinear structure** across views, resulting in 3D curvilinear structure. Pipelines based on straight lines (see Lebeda et al. (2014), Zhang (2013), Fathi et al. (2015) for recent reviews), algebraic and general curve features [Teney and Piater (2012), Litvinov et al. (2012), Fabbri and
Kimia (2010), Fabbri et al. (2012), Pötsch and Pinz (2011), Berthilsson et al. (2001), Fabbri and Kimia (2005)] have been proposed, but some lack generality, e.g., requiring specific curve models [Rico Espino et al. (2012)]. The 3D Curve Sketch system [Fabbri (2010), Fabbri et al. (2012), Fabbri and Kimia (2010)] operates on multiple views by pairing curves from two arbitrary “hypothesis views” at a time via epipolar-geometric consistency. A curve pair reconstructs to a 3D curve fragment hypothesis, whose reprojection onto several other “confirmation views” gathers support from subpixel 2D edges. The curve pair hypotheses with enough support result in an unorganized set of 3D curve fragments, the “3D Curve Sketch”. While the resulting 3D curve segments are visually appealing, they are fragmented, redundant, and lack explicit inter-curve organization.

The plethora of multiview representations, as documented above, arise because 3D structures are geometrically and semantically rich [Zia et al. (2015), Feng et al. (2014)]. A building, for example, has walls, windows, doorways, roof, chimneys, etc. The structure can be represented by sample points (i.e., unorganized cloud of points) or a surface mesh where connectivity among points is captured. This representation, especially when rendered with surface albedo or texture, is visually appealing. However, the representation also leaves out a great deal of semantic information: which points or mesh areas represent a window or a wall? Which two walls are adjacent? The representation of such components, or parts, requires an explicit representation of part boundaries such as ridges, as well as where these boundaries come together, such as junctions.

The same point can equally arise if objects in the scene were solely defined by their curve structures. A representation of a building by its ridges may usually give an appealing impression of its structure, but it fails to identify the walls, i.e., which collection of 3D curves bound a wall and what its geometry is.

Many applications such as robotics [Carlson et al. (2014)], urban planning and industrial design [Yee (2012), Leggitt (2015)] require a structured 3D representation which
Figure 1.1: **3D drawings for urban planning and industrial design.** A process from professional practice for communicating solution concepts with a blend of computer and handcrafted renderings [Leggitt (2015), Yee (2012)]. New designs are often based off real object references, mockups or massing models for selecting viewpoints and rough shapes. These can be modeled *manually* in, e.g., Google Sketchup (top-left), in some cases from reference imagery. The desired 2D views are rendered and *manually* traced into a reference curve sketch (center-left, bottom-left) easily modifiable to the designer’s vision. The stylized drawings to be presented to a client are often produced by *manually* tracing and painting over the reference sketch (right). Curves and surfaces can be used to generate reference **3D curve drawings** from video footage of the real site for urban planning, saving manual interaction, providing initial information such as rough dimensions, and aiding the selection of pose, editing and tracing. The condensed 3D curve drawings make room for the artist to overlay his concept and harness imagery as a clean reference, clear from details to be redesigned.

makes explicit 3D curves and 3D surfaces; spatial relationships among curves, among surfaces and between curves and surfaces; 3D components which make possible semantic interpretations of these structures, e.g., building ridges, building faces, chimneys, etc. Figure 1.1 shows an artist’s rendering of such relationships, the ultimate goal is to be able to obtain the 3D equivalent of these drawings in a fully automated fashion.

The previous approaches have focused on surface representation as a cloud of points or as meshes, but not as much work has focused on representing general 3D curves or
surface patches. This is a significant shortcoming because objects in the world are typically annotated by ridges, reflectance boundaries, texture boundaries, complex shapes and so on. Curves and surfaces, along with their differential geometric properties, are more structured and recognizable than point clouds, while lighter and more condensed than 3D meshes or voxel volumes for large-scale scenes. Furthermore, meshing a point cloud or a voxel representation often results in the oversmoothing of important 3D features such as ridges and corners.

In addition to architectural modeling, having such a rich, distinctive and multi-layered representation for 3D models would greatly benefit a wide range of 3D computer vision applications that aim to make higher-level inferences on 3D data in different ways.

**Efficient transmission of general 3D scenes:** Compared to triangular meshes and dense point clouds, 3D curves and surfaces are more compact because they only contain the most salient and robust features, as well as being more precise because they capture in fine detail the important 3D features that are often oversmoothed by reconstruction or meshing algorithms. Furthermore, the surfaces themselves can be entirely omitted during transmission to send only the 3D curves, and surfaces can be reconstructed on the receiver’s side, drastically shrinking the size of a package that needs to be transmitted. Such a representation would be a perfect fit for applications like terrain-mapping robots [Hadsell et al. (2012)] where compression is needed for efficient storage and/or transmission.

**Efficient interaction of a human agent with 3D data:** Occlusion and texture boundaries that form such a 3D representation would correspond to semantically meaningful features in a given scene and this is a distinct advantage in cases where a human agent will monitor and/or interact with the 3D models created, as is the case for most surveillance and military applications. Hadsell et al. (2012) show such a practical system where the necessary semantic cues are provided in the form of color-coding and the agent is asked to continuously switch between different color modes, a more arduous task than working
Figure 1.2: A series of visualizations for a robot-built 3D map, taken from Hadsell et al. (2012). The much needed semantic information and organization of 3D data is lacking, so the human agents interact with a series of color-coded 3D maps instead for a more intuitive interface. Such applications would greatly benefit from a richer representation with semantically significant features explicitly outlined.

with a representation that is capable of distinguishing semantic significance, Figure 1.2.

**Dense Surface Reconstruction:** A wireframe scaffolding can guide or complement dense surface reconstruction to be able to reconstruct textureless or shiny surfaces, precise ridges and semantically-meaningful features of a given scene. We describe our own dense reconstruction algorithm, which was built in this spirit, in Chapter 3.

**Fast Recognition/Registration of General 3D Scenery and General Scene Understanding:** The field of 3D vision has been expanding beyond 3D reconstruction algorithms to attempt higher-level vision tasks such as registration [Baatz et al. (2012), Restrepo et al. (2014), Hadsell et al. (2012)], recognition/classification [Restrepo et al. (2012), Aldoma et al. (2012)], segmentation [Kowdle et al. (2012), Matei et al. (2008)], scene understanding [Zia et al. (2015)] and tracking/navigation [Lebeda et al. (2014), Rao et al. (2012) in 3D. Given this increasing emphasis on processing 3D models for higher-level understanding of what they represent, the relative richness of curves and surfaces is a distinct advantage.
Furthermore, 3D junctions naturally arise as part of the 3D curve reconstruction process, as described in Chapter 2 and serve as 3D feature points of sorts, which can encode various salient geometric properties of the curves meeting at that junction, such as the pairwise angles between them or histograms of their lengths, tangents and/or curvature. Essentially, these properties form new and lightweight 3D geometric descriptors, which have the potential to replace the existing geometric descriptors such as the Viewpoint Feature Histogram [Rusu et al. (2010)], or Fast Point Feature Histogram [Rusu et al. (2009)], both of which introduce additional, non-negligible computational costs to the reconstruction process and therefore are rarely used in practical applications. A RANSAC-based algorithm that operates on these junctions and their attached descriptors can match, register or recognize 3D models with a lot less points than, say, what would be required for an ICP-like algorithm to perform the same task.

In general, image curve fragments are attractive because they have good localization, they have greater invariance than interest points to changes in illumination, are stable over a greater range of baselines, and are typically denser than interest points. Furthermore, the reflectance or ridge curves provide boundary condition for surface reconstruction, while occluding contour variations across views lead to surfaces [Cipolla and Giblin (2000), Liu et al. (2008), Taubin et al. (2008)]. Recent studies strongly support the notion that image curves contain much of the image information, see Koenderink et al. (2013), Zucker (2014), Künsberg and Zucker (2014), Cole et al. (2009). Moreover, curves are structurally rich as reflected by their differential geometry, a fact which is exploited both in recent computer systems [Zucker (2014), Abuhashim and Sukkarieh (2012), Fabbri et al. (2012), Fabbri and Kimia (2010)] and perception studies [Fleming et al. (2011), Zucker (2014)].

Finally, an advantage of working with curved structures instead of unorganized isolated points is that they are more intimately tied with the eventual surface models we are aiming to obtain, *i.e.*, most curved structures in a 3D model either exist on a surface or its boundary,
and therefore can serve as a boundary condition to solve for the missing surfaces between them.

In computer vision and graphics literature, there has been scattered but persistent interest in using 3D curves to infer aspects of an underlying shape [Maekawa and Ko (2002), Zorin (2006)], shape-related features linked to shading [Bui et al. (2015)], or closed 3D curves [Zhuang et al. (2013)]. For example, the approach in Sadri and Singh (2014) exploits the flow complex, a structure that captures both the topology and the geometry of a set of 3D curves, to construct an intersection-free triangulated 3D shape. Similarly, the approach in Pan et al. (2015) explores a similar concept with flow lines, which are designed to encapsulate principal curvature lines on a surface. As another example, the approach in Abbasinejad et al. (2012) identifies potential surface patches delineated by a 3D curve network, breaking them into smaller, planar patches to represent a complex surface. These methods are completely automated and yield impressive results on a wide range of objects. However, they require a complete and accurate input curve network, which is very difficult to obtain in a bottom-up fashion from image data: there will always be holes, missed curves, incorrect groupings, noise, outliers, and other real-world imperfections. Furthermore, these methods are not general, but rather tailored for scenes with objects of relatively clean geometry. Thus, they are not suitable for more general, large-scale complex scenes that the multiview stereo community tackles on a regular basis.

In this work, we present two major contributions to the field of multiview stereo reconstruction. In chapter 2 a novel approach is presented, which builds a collection of 3D curves from images, with topological connectivity between them represented as a 3D graph. This results in a 3D drawing, which is complementary to surface representations in the same sense as a 3D scaffold complements a tent taut over it. In chapter 3 a novel, fully-automated dense surface reconstruction approach is presented, which fills this scaffold with surfaces consistent with the 3D curves serving as their boundary conditions, by exploiting a method
called “Lofting”, which not well-known in the computer vision community, in conjunction with occlusion cues. Chapter 4 discusses conclusions and future research directions.
Chapter 2

From Multiview Image Curves to 3D Drawings

2.1 Introduction

In this section, we describe the technology to process a series of (intrinsic and extrinsically) calibrated multiview images to generate a 3D curve drawing as a graph of 3D curve segments meeting at junctions. The ultimate goal of this approach is to integrate the 3D curve drawing with the traditional recovery of surfaces so that 3D curves bound the 3D surface patches towards a more semantic representation of 3D structures. The 3D curve drawing can also be of independent value in applications such as fast recognition of general 3D scenery [Pötsch and Pinz (2011)], efficient transmission of general 3D scenes, scene understanding and modeling by reasoning at junctions [Mattingly et al. (2015)], consistent non-photorealistic rendering from video [Chen and Klette (2014)], modeling of branching structures, among others [Rao et al. (2012), Kowdle et al. (2012), Wang et al. (2014)].

The rest of this section is organized as follows. In Section 2.2 we review Enhanced Curve Sketch, our 3D curve sketching algorithm that is heavily based on the work [Fabbri
Figure 2.1: The 3D curve drawing transforms the curves in calibrated views of a scene (top 2 rows) into a “3D drawing” (bottom 2 rows) – a graph of 3D curves meeting at junctions. Each curve is shown in a different color.
and Kimia (2010)]. We identify three shortcomings in their curve representation and suggest solutions and improvements for each. Section 2.3 presents as our main contribution the multiview integration of information both at edge- and curve-level, which naturally leads to junctions. Section 2.4 discusses the process with which a synthetic 3D ground truth model was constructed, together with the accompanying calibrated image sequence. Finally, section 2.5 validates the approach using real and synthetic datasets.

2.2 Enhanced 3D Curve Sketch

Image curve fragments formed from grouped edges are central to our framework. Each image $V^v$ at view $v = 1, \ldots, N$ contains a number of curves $\gamma^v_i$, $i = 1, \ldots, M^v$. Reconstructed 3D curve fragments are referred as $\Gamma_k$, $k = 1, \ldots, K$, whose reprojection onto view $v$ is $\gamma^{k,v}$. Indices may be omitted where clear from context.

The initial stage of our framework is built as an extension of the hypothesize-and-verify 3D Curve Sketch approach [Fabbri and Kimia (2010)]. We use the same hypothesis generation mechanism with a novel verification step performing a finer-level analysis of image evidence and significantly reducing the fragmentation and redundancy in the 3D models.

Two image curves $\gamma^{v_1}_{l_1}$ and $\gamma^{v_2}_{l_2}$, each at least $\tau_l$ pixels long in their respective hypothesis views $v_1$ and $v_2$, are paired at a time, provided they have sufficient epipolar overlap of at least $\tau_e$ pixels [Fabbri and Kimia (2010)]. The verification of these $K$ curve pair hypotheses, represented as $\omega_k$, $k = 1, \ldots, K$ with the corresponding 3D reconstruction denoted as $\Gamma_k$, gauges the extent of edge support for the reprojection $\gamma^{k,v}$ of $\Gamma_k$ onto another set of confirmation views, $v = v_{i_3}, \ldots, v_{i_n}$. An image edge in view $v$ supports $\gamma^{k,v}$ if it is closer than $\tau_d$ pixels in distance and $\tau_\theta$ degrees in orientation. The total support a hypothesis $\omega$
receives from view $v$ is

$$S_{\omega_k}^v = \int_0^{L_{k,v}} \phi(\gamma_{k,v}(s)) ds,$$

(2.1)

where $L_{k,v}$ is the length of $\gamma_{k,v}$, and $\phi(\gamma(s))$ is the extent of edge support at $\gamma(s)$. A view is considered a supporting view for $\omega_k$ if $S_{\omega_k}^v > \tau_v$. Evidence from confirmation views is aggregated in the form

$$S_{\omega_k} = \sum_{v=1}^{i_v} [S_{\omega_k}^v > \tau_v].$$

(2.2)

The set of hypotheses $\omega_k$ whose number of inlier views $S_{\omega_k}$ exceeds the threshold $\tau_t$ are kept and the resulting $\Gamma_k$ form the unorganized 3D curves.

We identify three major shortcomings in this methodology: (i) some 3D curve fragments are correct for certain portions of the underlying curve and erroneous in other parts, due to multiview grouping inconsistencies; (ii) gaps in the 3D model, typically due to unreliable reconstructions near epipolar tangencies, where epipolar lines are nearly tangent to the curves; and (iii) multiple, redundant 3D structures. We now document each issue and describe our solutions.

**Erroneous grouping:** inconsistent multiview grouping of edges can lead to reconstructed curves which are veridical only along some portion, which are nevertheless wholly admitted, Fig. 2.2(a). Also, fully-incorrect hypotheses can accrue support coincidentally, as with repeated patterns or linear structures, Fig. 2.2(b). Both issues can be addressed by allowing for selective local reconstructions: only those portions of the curve receiving adequate edge support from sufficient views are reconstructed. This ensures that inconsistent 2D groupings do not produce spurious 3D reconstructions. The shift from cumulative global to multi-view local support results in greater selectivity and deals with coincidental alignment of edges with the reconstruction hypotheses.

**Gaps:** The geometric inaccuracy of curve segment reconstructions nearly parallel to
Figure 2.2: (a) Due to a lack of consistency in grouping of edges at the image level, a correct 3D curve reconstruction, shown here in blue, can be erroneously grouped with an erroneous reconstruction, shown here in red, leading to partially correct reconstructions. When such a 3D curve is projected in its entirety to a number of image views, we only expect the correct portion to gather sustained image evidence, which argues for a hypothesis verification method that can distinguish between supported segments and outlier segments; (b) An incorrect hypothesis can at times coincidentally gather an extremely high degree of support from a limited set of views. The red 3D line shown here might be an erroneous hypothesis, but because parallel linear structures are common in man-made environments, such an incorrect hypothesis often gathers coincidental strong support from a particular view or two. Our hypothesis verification approach is able to handle such cases by requiring explicit support from a minimum number of viewpoints simultaneously.

epipolar lines led Fabbri and Kimia (2010) to break off curves at epipolar tangencies, creating 2D gaps leading to gaps in 3D. We observe, however, that while reconstructions near epipolar tangency are geometrically unreliable, they are topologically correct in that they connect the reliable portions correctly but with highly inaccurate geometry. What is needed is to flag curve segments near epipolar tangency reconstructions as geometrically unreliable. We do this by the integration of support in Equation 2.1, giving significantly lower weight to these unreliable portions instead of fully discarding them, which greatly reduces the presence of gaps in the resulting reconstruction.

**Redundancy:** A 2D curve can pair up with dozens of curves from other views, all pointing to the same reconstruction, leading to redundant pairwise reconstructions as partially overlapping 3D curve segments, each localized slightly differently. Our solution is to detect and reconcile redundant reconstructions. Since redundancy changes as one traverses a 3D curve, we reconcile redundancy at the local level: each 3D edge is in one-to-one
Figure 2.3: (a) Redundant 3D curve reconstructions (orange, green and blue) can arise from a single 2D image curve in the primary hypothesis view. If the redundant curves are put in one-to-one correspondence and averaged, the resulting curve is shown in (b) in purple. Our robust averaging approach, on the other hand, is able to get rid of that bump by eliminating outlier segments, producing the purple curve shown in (c).

correspondence with a 2D edge of its primary hypothesis view (i.e., the first view from which it was reconstructed), hence 3D edges can be grouped in a one-to-one manner, all corresponding to a common 3D source. These are robustly averaged by data-driven outlier removal, where a Gaussian distribution is fit on all pairwise distances between corresponding samples, discarding samples farther than $2\sigma$ from the average, Fig. 2.3. Robust averaging improves localization accuracy, removes redundancy, and elongates shorter curve subsegments into longer 3D curves.

2.3 From 3D Curve Sketch to 3D Drawing

Despite the visible improvements of the Enhanced 3D Curve Sketch of Section 2.2, Fig. 2.4, curves are broken in many places, and there remains redundant overlap. The sketch representation as unorganized clouds of 3D curves are not able to capture the fine-level
Figure 2.4: A visual comparison of: (top) the curve sketch results of Fabbri and Kimia (2010) run at high-recall settings, with (bottom) the results of our enhanced curve sketch algorithm presented in Section 2.2. Notice the significant reduction in both outliers and duplicate reconstructions, without sacrificing coverage.
geometry or spatial organization of 3D curves, e.g. by using junction points to characterize proximity and neighborhood relations. The underlying cause of these issues is lack of integration across multiple views. The robust averaging approach of Section 2.2 is one step, anchored on one primary hypothesis view, but integrates evidence within that view only; a scene curve can be visible from multiple hypothesis view pairs, and some redundancy remains.

This lack of multiview integration is responsible for three problems observed in the enhanced curve sketch, Fig. 2.9: (i) localization inaccuracies, Fig. 2.9b, due to use of partial information; (ii) reconstruction redundancy, which lends to multiple curves with partial overlap, all arising from the same 3D structure, but remaining distinct, see Fig. 2.9c; (iii) excessive breaking because each curve segment arises from one curve in one initial view independently.

**Multiview Local Consistency Network:** The key idea underlying integration of reconstructions across views is the detection of a common image structure supporting two reconstruction hypotheses. Two 3D local curve segments depict the same single underlying 3D object feature if they are supported by the same 2D image edge structures. Since the identification of common image structure can vary along the curve, it must necessarily be a local process, operating at the level of a 3D local edge and not a 3D curve. Two 3D edge elements (edgels) depict the same 3D structure if they receive support from the same 2D edgels in a sufficient number of views, so 3D-2D links between a 2D edgel to the 3D edgel it supports must be kept. Typically, they share supporting image edges in many views; and the number of shared supporting edgels is the measure of strength for a 3D-3D link between them.

Formally, we define the Multiview Local geometric consistency Network (MLN) as pointwise alignments $\phi_{ij}$ between two 3D curves $\Gamma_i$ and $\Gamma_j$: let $\Gamma_i(s_i)$ and $\Gamma_j(s_j)$ be two
Figure 2.5: The four bottlenecks of Fig. 2.9 are resolved by integration of information/cues from all views. (a) The shared supporting edges, which are marked with circles, create the purple links between the corresponding samples of the 3D curves. These purple bonds will then be used to pull the redundant segments together and reorganize the 3D model into a clean 3D graph. Observe how the determination of common image support can identify portions of the green and blue curves as identical while differentiating the red one as distinct. A real example for a bundle of related curves is shown in (b) and the links among their edges in (c).

points in two 3D curves, and define

\[ S_{ij} = \{ v : \gamma^{i,v}(s_i) \text{ and } \gamma^{j,v}(s_j) \text{ share local support} \}. \]  

(2.3)

Then the a kernel function \( \phi \) defines a consistency link between these two points, weighted by the extent of multiview image support \( \phi_{ij}(s_i, s_j) = |S_{ij}| \). When the curves are sampled, \( \phi \) becomes an adjacency matrix of a graph representing links between individual curve samples. The implementation goes through each image edgel, and creates pairwise links between all 3D points that are supported by a given edgel. The algorithm automatically discards all links that are longer than \( 10\tau_m \) in order to reduce the number of outliers prior to the creation of MLN, reasoning that identical 3D samples could not possibly be that far apart.
Multiview Curve-level Consistency Network: The identification of 3D edges sharing 2D edges leads to high recall operating point with many false links due to accidental alignment of edge support. False positives can be reduced without affecting high recall by employing a notion of curve context for each 3D edgel: a link between two 3D edgels based on a supporting 2D edgel is more effective if the respective neighbors of the 3D edge on the underlying 3D curve are also linked.

The curve context idea requires establishing new pairwise links between 3D curves using MLN, when there are a sufficient number of links with $\phi_{ij} > \tau_c$ between their constituent 3D edges (in our implementation, $\tau_c = 3$ and we require $\tau_{sl}$ such edges, typically 5 or more). The linking of 3D curves is represented by the Multiview Curve-level Consistency network (MCCN), a graph whose nodes are the 3D curves $\Gamma_j$ and the edges represent the presence of high-weight 3D edge links between these 3D curves. The MCCN graph allows for a clustering of 3D curves by finding connected components; and once a link is established between two curves, there is a high likelihood of their edges corresponding in a regularized fashion, thus fewer common supporting 2D edges are required to establish a link between all their constituent 3D edges. This fact is used to perform gap filling, since even no edge support is acceptable to fill in small gaps and create a continuous and regularized correspondence if both neighbors of the gap are connected. The two stages in tandem, i.e., high recall linking of 3D edges and use of curve context to reduce false positives leads to high recall and high precision, i.e., all the 3D edges which need to be related are related and very few outlier connections remain.

Integrating information across related edges: The identification of a bundle of curves as arising from the same 3D source implies that we can improve the geometric accuracy of this bundle by allowing them to converge to a common solution. While this might appear straightforward, 3D edges are not consistently distributed along related curves, yielding a skew in the correspondence of related samples, Fig. 2.6, sometimes not a one-to-one
Figure 2.6: The correspondence between 3D edge samples is skewed along a curve, a direct indication that these links cannot be used as-is when averaging and fusing redundant curve reconstructions. Instead, each point is assumed to be in correspondence with the point closest to it on another overlapping curve, during the iterative averaging step. Observe that corrections can be partial along related curves.

This argues for averaging 3D curves and not 3D edge samples, which in turn requires finding a more regularized alignment between the 3D curves, without gaps; we find each curve samples’s closest point on the other curve.

When post averaging a sample with its closest points on related curves, the order of resulting averaged samples is not clear. The order should be inferred from the underlying curves, but this information can be conflicting, unless the distance between two curves is substantially smaller than the sampling distance along the curves. This requires first updating each curve’s geometry separately and iteratively, without merging curves until after convergence, Fig. 2.7d. This also improves the correspondence of samples at each iteration, as the closest points are continuously updated.

At each stage, the iterative averaging process simply replaces each 3D edge sample with the average of all closest points on curves related to it, Fig 2.7b–d. This can be formulated as evolving all 3D curves by averaging along the MCCN using closest points. Formally,
Figure 2.7: (a) A schematic of sample correspondence along two related 3D curves, showing skewed correspondences that may not be one-to-one. (b) A sketch of how two curves are integrated. Bottom 2 rows show real examples where the left column is the result of enhanced curve sketch, middle column shows the robust averaging output, and the right column shows the curve drawing result.
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 ←− ∅
while Visited $\neq S_T$, where $S_T$ is the set of curves indexed by $S$

do

for each cluster $C$ of the MCCN, do

$b_0 ←− \text{argmax}_{\Gamma \in C} \text{Length}(\Gamma)$

MDG$_0 ←− (J_0, B_0)$, where $B_0 = b_0$ and $J_0 = ∅$

Given a partial curve drawing graph $MDG_i = (J_i, B_i)$, $B_i = \{b_0, \ldots, b_{n_i}\}$

begin

Construct $B_{i+1}$

for each other curve $\Gamma \in C, \Gamma \notin \text{Visited}$ do

for each $s$ do

$B_\tau(s) = \{\tilde{\Gamma} \in B_i : d(\tilde{\Gamma}(s), \Gamma(s)) < \tau_m\}$

if $B_\tau(s) = ∅$ then

// Sample $\Gamma(s)$ has no corresponding branch

if no branch started then

Start new branch $\tilde{b}$

else if branch has started and $s$ at endpoint of $\Gamma$ then

// The new branch is added with the correct topology, either breaking or elongating existing branches and creating junctions, according to Figure 2.8

$B_{i+1} ←− B_i \cup \{\tilde{b}\}$

else // Sample $\Gamma(s)$ has corresponding branches

begin

Insert this sample into the corresponding curve

for each $\tilde{\Gamma} \in B_\tau(s)$ do

Update $\tilde{\Gamma}$ to contain $\Gamma(s)$

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 2.8

$B_{i+1} ←− B_i \cup \{\tilde{b}\}$

end

end
each $\Gamma_i$ is evolved according to

$$\frac{\partial \Gamma_i(s)}{\partial t} = \alpha \ \text{avg}_{(i,j) \in L} \left\{ \Gamma_j(r) : \Gamma_j(r) = \text{cp}_j(\Gamma_i(s)) \right\},$$

(2.4)

where $\text{cp}_i(p)$ is the closest point in $\Gamma_i$ to $p$ and $L$ is the link set defined as follows: Let the set $S_{ij}$ of so-called strong local links between curves $\Gamma_i$ and $\Gamma_j$ be

$$S_{ij} = \{(s,t) : \phi_{ij}(s,t) \geq \tau_e, \phi_{ij} \in \text{MLN}(\Gamma_1, \ldots, \Gamma_K)\}.$$

(2.5)

Then the set $L$ of the MCCN is defined as

$$L = \{(i,j) : |S_{ij}| \geq \tau_{sl}\}.$$

(2.6)

In practice, the averaging is robust and $\alpha$ is chosen such that in one step we move to the average.

**3D Curve Drawing Graph:** Once all related curves have converged, they can be merged into single curves, separated by junctions where 3 or more curves meet. The order along the resulting curve is also dictated by closest points: The immediate neighbors of any averaged 3D edge are the two closest 3D edges to it among all converged 3D edges in a given MCCN cluster.

This is where junctions naturally arise: as two distinct curves may merge along one portion they may diverge at one point, leaving two remaining, non-related subsegments behind, Fig 2.7e. This is a junction node relating three or more curve segments, and its detection is done using the merging primitives, whose complete set are shown in Fig. 2.8. The intuition is this: a complex merging problem along the full length of two 3D curves actually consists of smaller, simpler and independent merging operations between different segments of each curve. A full merging problem between two complete curves can be
Figure 2.8: The complete set of merging primitives, which were systematically worked out to cover all possible merging topologies between a pair of curves whose overlap regions are computed beforehand. We claim that any configuration of overlap between two curves can be broken down into a series of these primitives along the length of one of the curves. The 5th primitive is representative of a bridge situation, where the connection at either end of the yellow curve can be any one of the first four cases shown, and 6th primitive is representative of a situation where only one end of the yellow curve connects to multiple existing curves, but not necessarily just two.
Figure 2.9: (a) The four main issues with the enhanced curve sketch: (b) localization errors along the camera principal axis, which cause loss in accuracy if not corrected, (c) redundant reconstructions due to a lack of integration across different views, (d) the reconstruction of a single long curve as multiple, disconnected (but perhaps overlapping) short curve segments, and (e) the lack of connectivity among distinct 3D curves which naturally form junctions. (f-i) shows the 3D drawing reconstructed from this enhanced curve sketch, as described in Section 2.3. Observe how each of the four bottlenecks have been resolved. Additional results are evaluated visually and quantitatively, and are reported in Section 2.5.

expressed as a permutation of any number of simpler merging primitives. These primitives were worked out systematically to serve as the basic building blocks capable of constructing all possible configurations of our merging problem.

After iterative averaging, all resulting curves in any given cluster are processed in a pairwise fashion using these primitives: initialize the 3D graph with the longest curve in the cluster, and merge every curve in the cluster one by one into this graph. At each step, any number of these merging primitives arise and are handled appropriately. This process outputs the Multiview Curve Drawing Graph (MDG), which consists of multiple disconnected 3D graphs, one for each 3D curve cluster in the MCCN. The nodes of each graph are the junctions (with curve endpoints) and the links are curve fragment geometries. This structure is the final 3D curve drawing, see Algorithm 1 for a detailed pseudo-code of our entire algorithm.
2.4 Constructing Synthetic Ground Truth Models 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 [Seitz et al. (2006)] 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 [Strecha et al. (2008)] 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 2.10, as well as the precisely calibrated views, 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.

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
Figure 2.10: 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.
Figure 2.11: The full Barcelona Pavilion synthetic ground truth (top) and the bounding box that we have limited our quantitative evaluation to (bottom).
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 2.11. 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 2.12 and Figure 2.13.

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 [Aan et al. (2012)] 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.

2.5 Experiments and Evaluation

In Section 2.3 we have described an algorithm that relies on a number of system parameters to produce accurate results. The pipeline itself is not very fragile to changing parameters, however most parameters need to be assigned a value relative to certain attributes of the input data such as image resolution (1080p in our experiments), number of images in the input sequence (between 25 and 50 in our experiments) and average distance between curve sketch samples (measured to be 1mm in our experiments). In Table 2.1 we provide a summary of all system parameters along with the default values they were
Figure 2.12: Process of deleting mesh edges to produce the desired ground truth edges.
Figure 2.13: Detail of our ground truth generation. Even minute objects were modeled by discarding internal mesh edges (blue).

assigned for the input sequences in our experiments.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_l$</td>
<td>minimum length of curve fragments</td>
<td>40 pixels</td>
</tr>
<tr>
<td>$\tau_e$</td>
<td>minimum epipolar overlap</td>
<td>5 pixels</td>
</tr>
<tr>
<td>$\tau_d$</td>
<td>maximum distance of supporting edgels</td>
<td>5 pixels</td>
</tr>
<tr>
<td>$\tau_m$</td>
<td>sample merge threshold</td>
<td>1mm</td>
</tr>
<tr>
<td>$\tau_e$</td>
<td>minimum strength for a strong local link</td>
<td>3</td>
</tr>
<tr>
<td>$\tau_{sl}$</td>
<td>minimum number of strong links to connect curves in the MCCN</td>
<td>5</td>
</tr>
<tr>
<td>$\tau_\theta$</td>
<td>maximum orientation difference of supporting edgels</td>
<td>$10^\circ$</td>
</tr>
<tr>
<td>$\tau_t$</td>
<td>minimum total number of inlier views for a match to be considered reliable</td>
<td>5 edgels</td>
</tr>
<tr>
<td>$\tau_v$</td>
<td>minimum inliers of a supporting view</td>
<td>10 edgels</td>
</tr>
<tr>
<td>$b$</td>
<td>baseline between consecutive views</td>
<td>$20^\circ$</td>
</tr>
</tbody>
</table>

Table 2.1: Table of parameters for the curve drawing system.

We have devised a number of large real and synthetic multiview datasets, available at multiview-3d-drawing.sourceforge.net.

The Barcelona Pavilion Dataset: a realistic synthetic dataset we created for validating the present approach with control over illumination, geometry and cameras. It consists of: 3D models composing a large, mostly man-made, scene professionally composed by eMirage studios using the 3D modeling software Blender; ground-truth cameras fly-by’s around chairs with varied reflectance models and cluttered background; (iii) ground-truth
Figure 2.14: Our publicly-available synthetic (left and top-right) and real (bottom-right) 3D ground truths modeled and rendered using Blender for the present work.

videos realistically rendered with high quality ray tracing under 3 extreme illumination conditions (morning, afternoon, and night); (iv) ground-truth 3D curve geometry obtained by manually tracing over the meshes. This is the first synthetic 3D ground truth for evaluating multiview reconstruction algorithms that is realistically complex – most existing ground truth is obtained using either laser or structured light methods, both of which suffer from reconstruction inaccuracies and calibration errors. Starting from an existing 3D model ensures that our ground truth is not polluted by any such errors, since both 3D model and the calibration parameters are obtained from the 3D modeling software, Fig. 2.14. The result is the first publicly available, high-precision 3D curve ground truth dataset to be used in the evaluation of curve-based multiview stereo algorithms. For the experiments reported in the main manuscript we use 25 views out of 100 from this dataset, evenly distributed around the primary objects of interest, namely the two chairs, see Fig. 2.14.

**The Vase Dataset:** constructed for this research from the DTU Point Feature Dataset with calibration and 3D ground truth from structured light Aan et al. (2012), Jensen et al. (2014). The images were taken using an automated robot arm from pre-calibrated positions and our test sequence was constructed using views from different illumination conditions
to simulate varying illumination. To the best of our knowledge, these are the most exhaustive public multiview ground truth datasets. To generate ground-truth for curves, we have constructed a GUI based on Blender to manually remove all points of the ground-truth 3D point-cloud that correspond to homogeneous scene structures as observed when projected on all views, Fig. 2.14(bottom). What remains is a dense 3D point cloud ground truth where the points are restricted to be near abrupt intensity changes on the object, i.e. edges and curves. Our results on this real dataset showcase our algorithm’s robustness under varying illumination.

**The Amsterdam House Dataset:** 50 calibrated multiview images, also developed for this research, comprising a wide variety of object properties, including but not limited to smooth surfaces, shiny surfaces, specific close-curve geometries, text, texture, clutter and cast shadows, Fig. 2.1. The camera reprojection error obtained by Bundler Agarwal et al. (2009) is on average subpixel. There is no ground truth 3D geometry for this dataset; the intent here is: to qualitatively test on a scene that is challenging to approaches that rely on, e.g., point features; and to be able to closely inspect expected geometries and junction arising from simple, known shapes of scene objects.

**The Capitol High Building:** 256 HD frames from a high $270^\circ$ helicopter fly-by of the Rhode Island State Capitol Fabbri and Kimia (2010). Camera parameters are from the Matlab Calibration toolbox and tracking 30 points.

**Qualitative Evaluation:** The enhancements of Section 2.2 lead to significant improvements to the 3D curve sketch of Fabbri and Kimia (2010) in increasing recall while maintaining precision. See Fig. 2.4 for a qualitative comparison. When the clean clouds of curves are organized into a set of connected 3D graphs, the results are more accurate, more visually pleasing and not redundant, Fig. 2.9(f-i). Each of the issues in Fig. 2.9(a-e) have been resolved and spatial organization of 3D curves have been captured as junctions, represented by small white spheres. See Figures 2.15, 2.16 and 2.17 for our 3d drawing results.
Figure 2.15: The 3D drawing results, together with PMVS results, on the Brown Capitol dataset

Figure 2.16: The 3D drawing results, together with PMVS results, on the Barcelona pavilion dataset
on the Capitol, Barcelona Pavilion and Vase datasets respectively, along with PMVS results provided for comparison.

**Quantitative Evaluation:** Accuracy and coverage of 3D curve reconstructions is evaluated against ground truth. We compare 3 different results to quantify our improvements: (i) Original Curve Sketch Fabbri and Kimia (2010) run exhaustively on all views, (ii) Enhanced Curve Sketch, Section 2.2, and (iii) Curve Drawing, Section 2.3. Edge maps are obtained using Third-Order Color Edge Detector Tamrakar and Kimia (2007), and are linked using Symbolic Linker Guo et al. (2014) to extract curve fragments for each view. Edge support thresholds are varied during reconstruction for each method, to obtain precision-recall curves. Here, **precision** is the percentage of accurately reconstructed curve samples: a ground truth curve sample is a true positive if its closer than a proximity threshold to the reconstructed 3D model. A reconstructed 3D sample is deemed a false positive.
Figure 2.18: Precision-recall curves for quantitative evaluation of 3D curve drawing algorithm: (a) Curve sketch, enhanced curve sketch and curve drawing results are compared on Barcelona Pavilion dataset with afternoon rendering, showing significant improvements in reconstruction quality; (b) A comparison of 3D curve drawing results on fixed and varying illumination version of Barcelona Pavilion dataset proves that 3D drawing quality does not get adversely affected by varying illumination; (c) 3D drawing improves reconstruction quality by a large margin in Vase dataset, which consists of images of a real object under slight illumination variation.
if its not closer than $\tau_m$ to any ground truth curve. This method ensures that redundant reconstructions aren’t rewarded multiple times. All remaining curve samples in the reconstruction are false positives. **Recall** is the fraction of ground truth curve samples covered by the reconstruction. A ground truth sample is marked as a false negative if its farther than $\tau_m$ to the test reconstruction. The precision-recall curves shown in Fig. 2.18 quantitatively measure the improvements of our algorithm and showcase its robustness under varying illumination.
Chapter 3

The Surfacing of Multiview 3D Drawings via Lofting and Occlusion Reasoning

3.1 Introduction

Dense 3D surface reconstruction is an important problem in computer vision which remains challenging in general scenarios. Most existing multiview reconstruction methods suffer from some common problems such as: (i) Holes in the 3D model corresponding to homogeneous/reflective/transparent image regions, (ii) Oversmoothing of semantically-important details such as ridges, (iii) Lack of semantically meaningful surface features, organization and geometric detail.

We propose a novel and complementary dense 3D reconstruction approach based on occlusion reasoning and a CAD method called lofting, which is the process of obtaining 3D surfaces through the interpolation of 3D structure curves. Even though lofting is a very powerful tool, it does not appear to be used very much in the multiview geometry
Figure 3.1: The proposed approach transforms a 3D curve drawing (top) obtained from a fully calibrated set of 27 views, into a collection of dense surface patches (bottom) obtained via lofting and occlusion reasoning.

applications. Employing an existing curve-based reconstruction method, we start with a calibrated image sequence to build a 3D drawing of the scene in the form of a 3D graph, where graph links contain curve geometries and graph nodes contain junctions where curve endpoints meet. We propose to use the 3D drawing of a scene as a scaffold on which dense surface patches can be placed on. Our approach relies on the availability of a “3D drawing” of the surface, a graph of 3D curve fragments reconstructed from calibrated multiview observations of an object [Usumezbas et al. (2016)]. Observe that such a 3D drawing acts as a scaffold for the surface of the object in that the drawing breaks the object surfaces into 3D surface patches, which are glued on and supported by the 3D drawing scaffold. Our approach then is based on selecting some 3D curve fragments from the 3D drawing, forming surface hypotheses from these curve fragments, and using occlusion reasoning to discard inconsistent hypotheses.

The first task, then, is to select curve fragments from the 3D drawing to form a surface patch hypothesis. Aside from closed curves, which independently yield such hypotheses, the remaining veridical surface patches can be constructed from pairs of curve fragments.
Since it is not clear which pairs of curve fragments support surfaces, we hypothesize a set of candidate surface hypotheses on sufficiently long curve fragments, paired with other likely curve fragments in proximity that give rise to low-curvature surface hypotheses. The main cue to ruling out non-veridical hypotheses is occlusion consistency: Surface hypotheses occlude the 3D curve fragments; if image data invalidates an expected occlusion, the hypothesis under question is discarded.

Second, the main technology for constructing surface patch hypotheses from a pair (or a few) curve fragments is “lofting”, a drafting technique for generating streamlined objects from curved line drawings that was initially used to design and build ships and aircrafts. More recently, lofting has become a common technique in computer graphics and computer-aided design (CAD) applications where a collection of surface curves are used to define the surface through interpolation. To the best of our knowledge, lofting has not been used in multiview geometry reconstruction of surfaces. In this chapter, we advocate the use of this powerful technique in multiview geometry, in the context of proposing surface patch geometries consistent with a few selected curve fragments from a 3D drawing of the scene.

Aside from yielding a useful and semantically-meaningful intermediate representation, reconstructing surfaces by going through curved structures closely replicates the human act of drawing: As in a progressive drawing, the basis is independent of illumination conditions and other details. For instance, photometry/shading/reflectance can be incorporated later on either as hatchings or progressively refined as fine shading; multiple renderings can be performed from the same basis. Even challenging materials such as the ocean surface can be rendered on top of a curve basis. This approach also has the advantage of scalability, since it allows for a very large 3D scene to be selectively and progressively reconstructed.

In our vision, the use of image curves as a basic construct of multiview geometry complements the feature-based approach that is commonly used to reconstruct an unorganized cloud of points. These points can be constrained to lie on surface hypotheses, and then
can be viewed as isolated anchor points that complement the scaffold of 3D curves. On the other hand, dense multiview stereo and other shape-from-X approaches, which aim to reconstruct accurate 3D surface geometries, can also play a role that complement the current “topological” reconstruction in geometric reconstruction accuracy. We envision that when such information is reliably available, it can be used to guide the surface hypothesis formation process.

This chapter is organized as follows: Section 3.2 reviews lofting and describes how a surface is generated from a few curve fragments lying on the surface. Section 3.3 describes how 3D surface patch hypotheses are generated from a 3D drawing, and how occlusion consistency is used to take out non-veridical hypotheses. Section 3.5 deals with several technical challenges, which require a regularization of the 3D drawing so that surface patches can be robustly inferred. Section 3.5 presents experimental results, a comparison with PMVS by Furukawa and Ponce (2007, 2010), and quantification of reconstruction accuracy.
3.2 Bringing Lofting Into Multiview Stereo

Lofting is a graphics technique for shape inference from a set of 3D curves, a term with roots in shipbuilding to describe the molding of a hull from curves [Bole (2015)]. Designers often use such intermediate, curve-based representations (sketches, graphs, drawings) to outline 3D shape, as they compactly capture rich 3D information and are easy to customize. Through lofting, these 3D curves are used to interactively model smooth surfaces. Implementations of lofting are commonplace in interactive CAD [Blender Online Community (2016), Wu et al. (1977), Nealen et al. (2007), Morigi and Rucci (2011), Grimm and Joshi (2012), Abbasinejad et al. (2012), Das et al. (2005), Nam and Chai (2012)], and applications [Lin et al. (1997), Beccari et al. (2010), Tustison et al. (2004)]. Lofting has not yet spread to 3D computer vision, where fully-automated image-based modeling is the norm. This work leverages lofting to build a fully-automated, dense multiview stereo reconstruction pipeline.

Given 3D curves $\Gamma_1, \Gamma_2, \ldots, \Gamma_n$ forming the partial boundary of a surface, lofting produces a smooth surface passing through them which is sought to be ‘simple’: smooth, avoiding holes and degeneracies such as self-intersections, as shown in Fig. 3.2. Earlier approaches formulated this as surface deformation with parameters estimated to fit the prior into a 3D curve outline as in Chiyokura and Kimura (1983), Kraevoy et al. (2009). Approaches using functional optimization, such as Morigi and Rucci (2011), Nealen et al. (2007), Sorkine and Cohen-Or (2004), Welch and Witkin (1994), Bobenko and Schröder (2005), Moreton and Séquin (1992), employ generic objectives, such as least squares and integral of squared principal curvatures, and the result depends on this choice, leading to overfitting or oversmoothing. These approaches cannot easily handle complex shapes with many self occlusions [Lin et al. (1997)]. Other algorithms include those based on B-splines, such as Woodward (1988), Park et al. (2004).

We have chosen lofting based on subdivision surfaces, a well-known graphics technique
that divides the faces of a coarse input mesh via a recursive sequence of transforms or subdivision schemes, yielding smooth high-poly meshes, Fig. 3.3. Subdivision is widely used in a number of graphics problems [Peters and Reif (1997), DeRose et al. (1998)], such as surface fitting [Suzuki et al. (1999), Takeuchi et al. (2000), Ma et al. (2004)], reconstruction [Hoppe et al. (1994), Maekawa and Ko (2002), Zorin (2006)], and lofting itself [Nasri (1997, 2000, 2003), Nasri et al. (2001), Nasri and Abbas (2002), Catalano et al. (2008)]. Combined subdivision schemes, like the ones presented by Levin (1999b,a) translate conditions on the limit surface to conditions on the scheme itself, and allow subdivision to be adjusted near the curve network and boundary conditions beyond subdivision or spline curves. Subdivision surfaces provide a simple standard framework, with more powerful schemes compared to other techniques; meshes with complex constraints at corners can be handled with greater ease [Schaefer et al. (2004)]. We leverage Schaefer et al. (2004), which takes open 3D polygonal lines terminating in a set of corners – as in our 3D drawing, but interactively generated. We have augmented it to automatically reorganize the curve network prior to lofting, and with additional heuristics to avoid degeneracies. The result is a lofting approach that can: i) take any number of boundary curves partially or completely covering the boundary of the desired surface, and ii) handles topological inconsistencies, self-intersections, discontinuities and other geometric artifacts. A brief description of our lofting stage follows.
All of these approaches demonstrate clearly that more than one solution exists for any given lofting problem, each with its own set of assumptions and drawbacks. The approach we will outline in this section opts to proceed with the simplest, geometrically-coherent option that accurately explain a given set of input boundary curves. When surface patches are bounded by stationary 3D curves, their geometry is captured to a large degree by the geometry of these bounding 3D curves, as shown in Figure 3.4(a). We observe that most surfaces in the real world fall inside this category and are captured accurately by our approach. On the other hand, when surface patches are smoother, two kinds of issues may arise: i) In some cases non-stationary curves may start emerging on the surface boundaries, which are not captured by the curve-based modeling step we described in Chapter 2, see Figure 3.4(b). ii) In other cases, the boundary of the surface may still be stationary but unrelated to the geometry of the surface as shown in Figure 3.4(c). In both of these cases, our approach still manages to generate a simpler approximate surface that is not entirely incorrect but rather lacking geometric detail.

Figure 3.16 in Section 3.5 shows an example 3D model resulting from our algorithm, where an overwhelming number of surface patches have accurately-reconstructed geometries, while a small number of them feature simplifying approximations of the kind discussed above. More specifically, our algorithm accurately reconstructs the surface details on the seat cushions, but the backsides of chairs come out mostly planar. This is because the curved boundaries of the seat section were accurately captured, yielding accurate curved surface geometries, whereas for the backside only linear outlines of the periphery were captured, which resulted in a simplified planar surface. These results prove that our lofting approach is very robust in identifying the correct location and rough geometry of 3D surfaces even when it misses the fine geometric detail.

**Skinning:** quadrangulates the input curves to construct a quad topology base mesh without the final geometry [Schaefer et al. (2004), Piegl and Tiller (1996), Kaklis and Ginnis
Figure 3.4: Three possible outcomes of our lofting approach are illustrated using simple, analytical 3D shapes: a set of stationary boundary curves accurately capturing the geometry of a surface patch produces an accurate reconstruction via lofting (a); if the geometry of a surface patch is only roughly approximated due to non-stationary boundaries (b) or unrelated stationary boundaries (c), a simplified approximate shape is generated via lofting. Left column shows the ground truth shapes together with stationary boundary curves marked in red. Right column shows the loft surfaces obtained by running our approach on the red curves. Notice that even when geometric errors are present, a coarse-level geometry of the surface is still reconstructed.
Figure 3.5: Quadrangulation in lofting; depending on the configuration of special interior vertices on a chain, one of these four edits are applied to obtain a base mesh topology [Schaefer et al. (2004)].

(1996), Nasri et al. (2003)). Skinning does not produce accurate shape approximation, but mainly avoids vertices lacking curvature continuity [Loop (2004)]. Given a closed 3D curve $\Gamma = (s_1, \ldots, s_n)$, a chain is a subsequence $\Gamma_i^{i+k} = (s_i, \ldots, s_{i+k})$, $i = 1, \ldots, n + k$. The topology of the base mesh $\lambda$ is constructed by a sequence of chain advances on $\Gamma$: given $\Gamma_i^{i+k}$, this adds a layer of $k$ quads to $\lambda$ bounded below by $\Gamma_i^{i+k}$ and above by a new chain $\Gamma_j^{j+k} = (s_j, \ldots, s_{j+k})$ on the interior of the resulting patch $\lambda$. $\Gamma$ is replaced by $\tilde{\Gamma} = \Gamma_{i-1}^{i-1} \cup \Gamma_j^{j+k} \cup \Gamma_{i+k+1}^{i+k+1}$. Depending on the configuration of special interior vertices, different types of advances apply [Schaefer et al. (2004)], see Figure 3.5.

**Fairing** computes the positions of the vertices in $\lambda$ by minimizing “fairness” energy, a thin-plate functional [Schaefer et al. (2004)]. **Subdivision** is then applied with a modified version of Catmull-Clark schemes [Schaefer et al. (2004)], yielding a fine mesh, see Figure 3.3.

### 3.3 Automated Multiview Reconstruction Using Lofting

In the previous two sections, we described: (i) The concept of a 3D curve drawing, a graph of 3D contour fragments and a method for deriving it from a set of calibrated multiview imagery, and (ii) the concept of lofting which reconstructs 3D surface meshes bounded
Figure 3.6: A schematic of a simple shapes where a surface patch (green) is represented by a pair of curves (red); in the case of closed curves, a pair is not necessary.

by a set of given contour fragments. We now describe how pairs of curve fragments selected from the 3D curve drawing give rise to 3D surface hypotheses. These hypotheses are then ruled out when they predict occlusions which are not consistent with the input data. The remaining hypotheses yield a set of occlusion-consistent surface patches. In the following, we first describe the process of hypothesis formation and then testing of formed hypotheses for occlusion consistency.

**Forming Surface Patch Hypotheses:** Ideally, any subset of curve fragments should be able to form surface hypotheses, but this is clearly intractable; even if curve fragments are long, noiseless and salient (a critical factor as we shall see in Section 3.4), they number in the order of 100 curves or so. Note that surface patches that arise from closed curves are a special case and these be identified and processed a priori. The remaining surface patches involve at least two curve fragments but typically more, say around 3-5. Then, pairs of curve fragments can be used as entry level hypotheses, Figure 3.6.

The pool of curve fragments from which pairs are selected is restricted to those with a minimal length constraint, $L > \tau_{\text{length}}$. This threshold is learned from data and is typically around a few centimeters for our data. The distance between two 3D curves is defined
as the average point-to-curve distance for all the samples on both curves. The typical 3D curve proximity threshold $\tau_\alpha$, which is also learned from data, is around 15-20 cm.

Third, in addition to length and pair proximity, curvature of the reconstructed surface is a cue to whether it is veridical. This is because object surfaces are typically not as convoluted as surfaces arising from unrelated cues. We use average Gaussian curvature, \textit{i.e.} Gaussian curvature at every point on the surface averaged over all surface points, and a threshold $\tau_G$, which is also learned. It should be noted that every curve pair generates two surface hypotheses: each endpoint in a given curve can pair with two possible endpoints on the other curve in the pair. The surface hypotheses with lower average Gaussian curvature is the one that is selected, if it is above $\tau_G$, Figure 3.7. See Figure 3.8 for a collection of sample surface hypotheses obtained this way.

Note that an alternate method for forming pairs of 3D curve fragments is to use the topology of 3D curve fragments as projected onto 2D views. The topology of 2D image curves is derived from the medial axis or Delaunay Triangulation to determine the neighboring curve fragments for any given curve. The topology of projected 3D curve fragments then induces a neighborhood relationship among 3D curve fragments: two 3D curve fragments are neighbors in 3D if their corresponding 2D image curves are neighbors in at least one view. This improves the performance in two ways: (i) veridical pairing which exceed the proximity threshold are restored to the pool of candidate pairs; (ii) non-veridical curve pairs which are not neighbors are correctly discarded. This is a significant factor in areas dense in 3D curves compared to the proximity threshold, which generates numerous non-veridical curve pairs.

**Hypothesis Viability Using Occlusion Consistency:** The most important cue in probing the viability of a 3D surface patch hypothesis is whether it is consistent with respect to the occlusions it predicts (it is assumed that surfaces are opaque). If an opaque 3D surface patch is veridical, then all 3D curve structures that are occluded by it in a given projected
Figure 3.7: There is an inherent ambiguity in reconstructing a surface from two curve fragments arising from which endpoints are paired (top row vs. bottom row). When two curve fragments do belong to a veridical surface, one of the two reconstructions generally has much lower average Gaussian curvature than the other and this is a cue as to which one is veridical. When the pairing of curve fragments is incorrect in that no surface exists between them, both reconstructs have high average Gaussian curvature, a cue to remove outliers.
Figure 3.8: Some example loft surfaces of various geometries that our reconstruction algorithm generates.
Figure 3.9: A 3D surface patch $S$ occludes all 3D curve fragments that lie behind it. Thus, the 3D curve fragments between $\Gamma_1$ and $\Gamma_4$ are partially obstructed so that only portions between $(\Gamma_1, \Gamma_2)$ and $(\Gamma_3, \Gamma_4)$ are visible as $(\gamma_1, \gamma_2)$ and $(\gamma_3, \gamma_4)$ in the image. The projections of $(\Gamma_2, \Gamma_3)$ should have no edge evidence in the image. On the other hand, the 3D curve fragments $(\Gamma_5, \Gamma_6)$ is fully unoccluded and edge evidence for it is expected. The presence of edge evidence in the portion $(\gamma_2, \gamma_3)$ is grounds for invalidating the 3D surface hypothesis $S$.

The technical approach to testing occlusion is based on ray tracing [Glassner (1989)]: A ray is connected from the camera center to each point on a 3D curve fragment belonging to the 3D curve drawing and the visibility of the point is tested against each surface hypothesis. Specifically, let $\{\Pi_1, \ldots, \Pi_N\}$ denote the set of hypothesized surface patches. Let the 3D curve drawing have curve fragments $\{\Gamma_1, \ldots, \Gamma_K\}$, each having image curve projections onto view $l$, $\gamma_k^l(s)$, where $s$ represents length parameter $s \in [0, L_k^l]$, where $L_k^l$ is the total length of the projected curve. Let the portion of the 3D curve that is occluded by the surface patch $\Pi_n$ be denoted by the interval $(a_{k,n}^l, b_{k,n}^l)$. Then, the evidence against surface hypothesis $\Pi_n$ provided by curve $\Gamma_k$ from view $l$, $E_{k,n}^l$, is the edge support for the invisible portion. This evidence is the sum of total edge support at sample point $s$, $\phi(\gamma_k^l(s))$, which is simply the number of image edges that have matching locations and orientations to the
curve $\gamma_k(s)$ at sample point $s$:

$$E_{n,k}^l = \int_{b_{k,n}}^{a_{k,n}} \phi(\gamma_k(s)) \, ds$$  \hspace{1cm} (3.1)

This evidence is then subjugated to a threshold of significance $\tau_E$; if significant, the evidence invalidates the hypothesis. On the other hand, if the evidence against the hypothesis for all the curves that should be occluded is indeed insignificant, i.e., $E_{n,k}^l < \tau_E$, $\forall k$, the lack of evidence in fact provides support for the surface hypothesis. This is to be distinguished from surface hypotheses that are not occluding any curves. The situation where $\Pi_n$ occludes $\Gamma_k$ and image evidence shows occlusion lends more evidence to $\Pi_n$ than the situation where $\Pi_n$ does not occlude any curves.

We now assume that all surface patches occlude at least one curve in at least one view; note that for polyhedral shapes, frontal patches occlude the contours of patches on the back, so this is not a stringent assumption. In fact, probing this assumption on both Amsterdam House Dataset and Barcelona Pavilion Dataset (which are described in Section 3.5) shows that this is the case for more than 90% of the surface hypotheses generated. This assumption implies that each surface hypothesis needs to be confirmed at least once against an occlusion hypothesis, i.e., $\forall n, \exists l, \exists k$, such that $E_{k,n}^l < \tau_E$.

The above process probes the implication of surface patch in relation to the 3D curve drawing. When introducing a multitude of surface patches, however, the issue of occlusion between two surface hypotheses arises. It is possible that one surface hypothesis is fully occluded by all other surfaces. Such a surface is then not visible in any view and is discarded.

**Redundant Hypotheses:** Since surface hypotheses are generated by pairs of 3D curve fragments, if a ground truth surface consists of multiple curve fragments, say a rectangular patch consisting of four curve fragments, then the same surface will likely be represented by a number of curve fragment pairs, six possible pairs in the case of a rectangular patch.
These redundant representations are detected in a post-processing stage and consolidated. When a large portion of a surface hypothesis (80% in our system) is subsumed by another surface, i.e., 80% of the points on it are closer than a proximity threshold to another surface, then this surface is discarded as a redundant hypothesis. A more principled approach is to merge two overlapping surfaces by forming curve triplet hypotheses: When two curve pairs have a curve fragment in common and their surface hypotheses overlap, as described above, the lofting approach is applied to the curve triplet and the resulting surface replaces the pair of surface hypotheses. And, of course, a curve triplet and a curve pair with a common curve fragment and overlapping surfaces result in curve quadruplet hypotheses, and so on as needed. This growth of surface hypotheses yields more accurate and less redundant surface patches, but results from this process are not ready for inclusion in this publication.

Figure 3.10 is a visual illustration of our entire surface reconstruction approach. Figure 3.11 demonstrates that our algorithm is very good at correlating image edges with 3D curve structures, accurately reasoning about occlusion and confirming an overwhelming majority of correct surfaces, as well rejecting almost all of the incorrect hypotheses. Figure 3.12. It should be noted that many surface hypotheses do not contain any portion of the curve drawing behind them from any given view. These hypotheses cannot be confirmed or denied, and, depending on the robustness of the hypothesis generation algorithm, they can be included in or discarded from the output as needed. In addition, many existing multiview stereo methods can be plugged into our system at the level of curve pairing and used as alternative ways to provide initial seeds for our surface hypotheses. As mentioned
Figure 3.11: Examples to surface hypotheses being confirmed by the confirmation views shown in this here. Left column: Projected surface hypothesis is shown in green, projected curve drawing is shown in blue and occluded segments are shown in purple. Right column: Same surface and occluded segments are shown with image edges in blue. Notice the lack of any edge presence whatsoever around most of the purple segments, which is clear indication of occlusion consistency between the images and the hypothesis surface.
earlier, our lofting algorithm scales well to a large number of input 3D curves, which are
provided either simultaneously or sequentially.

3.4 Reorganization of Input Curve Graph/Network Using
1st-Order Differential Geometry

Four important technical issues arise in the application of lofting to reconstruct surface
patches from 3D drawings.
Problem 1: Lofting sensitivity to overgrouping: First, lofting is highly sensitive to overgrouping in grouping edges into curves. If some parts of a curve belong to a veridical surface patch but another part does not, then the lofting results experience significant and irreversible geometric errors, e.g., as in Figure 3.13a where two curve fragments $C_1$ and $C_2$ belong to a side of the house and correctly hypothesize a surface patch through lofting. However, if $C_2$ is grouped with an adjacent curve fragment $C_3$ belonging to an adjacent face of the house that $C_2$ belongs to (let $C_4$ denote $C_2 \cup C_3$), then the lofting results based on $(C_1, C_4)$ do not produce a meaningful surface patch. The core of this problem is that the curve $C_2$ is shared by two surface hypotheses, but if grouped with $C_3$, it can no longer represent the frontal surface hypothesis created by $C_1$ and $C_2$. This transition in the ability to represent multiple surface hypotheses happens at junctions. Thus, breaking all curves at corners, i.e., high-curvature points, should remedy this problem. Unfortunately, it is difficult to output curvature for noisy curves, thus requiring a smoothing algorithm before the curve can be broken at high-curvature points. This smoothing algorithm is described below in the context of curve noise.

Problem 2: Lofting sensitivity to curve noise: Second, curve fragments of the 3D drawing can have excessive noise, depicting loop-like structures and local perturbations, Figure 3.13b. These degeneracies in the local form of a curve fragment often result in failures in the lofting algorithm to produce a surface hypothesis, or result in surfaces featuring geometric degeneracies. There are a number of smoothing methods, and we use a relatively recent robust algorithm that is based on B-spline models [Garcia (2010, 2011)], balancing data fidelity term with a smoothness term. The ratio of those two terms determine the degree of smoothing. The advantage of this method is that the polyline representation of the curve can be maintained after smoothing.

Problem 3: Lofting sensitivity to overfragmentation and gaps: Third, lack of edges or undergrouping in the edge grouping stage can lead to gaps and overfragmentation. In
Figure 3.13: (a) Overgrouping of two curve fragments $C_2$ and $C_3$ into $C_4$ can lead to nonsensical lofting results in the pair $(C_1, C_4)$ in contrast to the close-to-veridical results of lofting $(C_1, C_2)$; (b) lofting is sensitive to loop-like noise or excessive perturbations; (c) lofting with overfragmented curves produces suboptimal lofting results as well as redundant surface proposals, and leads to a combinatorial increase in the number of lofting applications and postprocessing.

both cases, a long veridical curve is represented as multiple smaller curve fragments, Figure 3.13c. As a result, what would have been a single surface patch now needs to be covered by a suboptimal set of smaller, overlapping surface hypotheses. In addition, the increased number of curve fragments increases the number of curve pairs to be considered, and lead to a combinatorial increase in computational cost. Curve fragments that are coincidental at a point can be grouped if they show good continuity of tangents at endpoints. Similarly, gaps between two curve fragments $\Gamma_1(s)$ and $\Gamma_2(s)$ can be bridged between endpoint $\Gamma_1(s_1)$ and $\Gamma_2(s_2)$ if: (i) These endpoints are sufficiently close, i.e., $|\Gamma_1(s_1) - \Gamma_2(s_2)| < \tau_{dist}$, where $\tau_{dist}$ is a gap proximity threshold, and (ii) $CC((\Gamma_1(s_1), T_1(s_1)), (\Gamma_2(s_2), T_2(s_2))) < \tau_{cocirc}$ where $CC$ is the co-circularity measure, characterizing good continuation from one point-tangent pair $(P_1, T_1)$ to another pair $(P_2, T_2)$ [Parent and Zucker (1989)].

**Problem 4: Duplications due to curve fragment overlaps:** Fourth, there is some duplication in 3D curve fragments in that two curves can overlap along portions, thus creating duplicate surface representations. While this duplication may not be an issue for some applications, better results can be obtained if the duplication is removed: When two curves overlap, the longer curve is unaltered and the overlapping segment is removed from the shorter curve. The curves are also downsampled since the initial curve drawing is dense in
sample points.

The resolution of the above four problems significantly improves the performance of our algorithm. Note that these steps are applied in sequence: Pruning small curves, smoothing curve fragments, gap filling and grouping overfragmented segments, eliminating duplications and downsampling. In addition, it is judicious to iteratively apply these steps in sequence, starting with small parameters and increasing the parameters in steps (typically 3-4). This is crucial because all of these steps run the risk of distorting the 3D data in significant ways if pursued too aggressively in a single iteration, e.g., corners can be over-smoothed, wrong gaps can be filled, meaningful but relatively short curve fragments can get pruned without getting a chance to be merged into a larger curve fragment etc.

It should be noted that aforementioned problems do not arise in the plethora of interactive surface lofting approaches, as a human agent is available to break or group 3D structures to obtain geometrically accurate 3D surfaces [Nealen et al. (2007)]. Some of the lofting approaches try to get around this problem by constraining the input curves to be closed curves [Zhuang et al. (2013), Schaefer et al. (2004)] but a fully automated, bottom-up lofting system like ours has to be able to handle such grouping inconsistencies algorithmically.

In summary, this regrouping algorithm exploits the underlying organization, as well as the rich differential geometric properties embedded in any sufficiently-smooth, 3D curve representation, to adjust the granularity and connectivity of any input curve graph or network to suit the needs of a wide variety of applications. In the case of surface lofting, the quality of the resulting reconstructions are significantly improved if the input curves that have 3D surfaces between them have their samples more or less linearly aligned with each other, resulting in a more robust quadrangulation step that kickstarts most lofting approaches. We therefore use the 1st and 2nd order differential geometric cues, namely tangents and curvatures, to full extent in order to aggressively group smooth segments and
Figure 3.14: The original input 3d curve drawing (top row), which is the direct output of the 3D curve drawing approach, the result of our reorganization algorithm before breaking sharp corners (middle row), and after the sharp corners are broken (bottom row). The level of granularity displayed in the last row is the most appropriate for our lofting approach, as most surfaces are bounded by entire curves rather than subsegments.
break curves at high-curvature points, maximizing the likelihood that the lofting algorithm will receive a set of 3D curves best suited for its capabilities.

3.5 Experiments and Results

**Implementation:** The 3D drawing is computed using code made available by the authors of Usumezbas et al. (2016). Smoothing code was made available by Garcia (2010). The selection of lofting software is among the most critical; we have selected one of the most robust lofting implementations, BSurfaces, a part of the Blender software [Eclectiel and Blender Online Community (2016)], a well-known, professional-grade CAD tool which is in widespread use. BSurfaces is able to work on multiple curves with arbitrary topology and configurations, either simultaneously or incrementally, and produces simple and smooth surfaces that accurately interpolate input curves, even if they only partially cover the boundary of the surface to be reconstructed. The use of BSurfaces has been limited to interactive modeling, where a human agent provides clean and well-connected curves to the system. To the best of our knowledge, a fully-automated 3D modeling pipeline that first obtains a 3D curve network, and then uses lofting to surface this network in a fully-automated fashion using lofting, is novel.

**System parameters:** We utilize Gaussian distributions to perform straightforward statistical analyses on the input data to automatically set system parameters. Curve-based thresholds such as $\tau_{\text{length}}, \tau_{\text{dist}}, \tau_{\alpha}$ are learned using the 3D input curves: We separately fit a Gaussian to the distribution of these attributes in the input curves to determine reasonable thresholds. For example, the mean of distances between consecutive curve samples in the input curves determine $\tau_{\text{dist}}$, which in turn determines if a gap between two curve endpoints is small enough to be bridged. Similarly, we fit a Gaussian distribution on input curve lengths and $\tau_{\text{length}}$ is set to $\mu - 2\sigma$ and curves shorter than $\tau_{\text{length}}$ are discarded as unusually short. Surface-related thresholds such as $\tau_{G}, \tau_{E}$ are set by statistically observing
a number of ground truth surface geometries in a similar fashion. Table 3.1 summarizes these parameters and provides some of the default values we’ve used in our experiments. In the interest of succinctness, discussion regarding the numerical defaults for some of these parameters is omitted; in those cases, the default value is marked as “learned from data”. While the parameters of our system are important to obtaining accurate results, they don’t make our approach excessively fragile because they are easily adaptable to various kinds of input data.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_{\text{length}}$</td>
<td>minimum length of 3D curve fragments to be merged into the drawing</td>
<td>3cm</td>
</tr>
<tr>
<td>$\tau_{\alpha}$</td>
<td>maximum average point-to-curve distance allowed for a pair of 3D curve fragments to be included in the hypothesis set</td>
<td>30cm</td>
</tr>
<tr>
<td>$\tau_{G}$</td>
<td>maximum average Gaussian curvature allowed for 3D surfaces</td>
<td>learned from data</td>
</tr>
<tr>
<td>$\tau_{E}$</td>
<td>minimum number of structured edge support required to rule out a hypothesis surface</td>
<td>10 edgels</td>
</tr>
<tr>
<td>$\tau_{\text{dist}}$</td>
<td>maximum length for 3D curve gaps to be bridged</td>
<td>learned from data</td>
</tr>
<tr>
<td>$\tau_{\text{cocirc}}$</td>
<td>minimum amount of co-circularity required for bridging gaps</td>
<td>learned from data</td>
</tr>
</tbody>
</table>

Table 3.1: Table of parameters for the lofting system.

**Datasets:** We use two datasets to quantify experimental results. First, the Amsterdam House Dataset consists of 50 fully calibrated multiview images and comprises a wide variety of object properties, including but not limited to smooth surfaces, shiny surfaces, specific close-curve geometries, text, texture, clutter and cast shadows. This dataset is used to evaluate the occlusion and visibility reasoning part of our pipeline, described in Section 3.3. Second, the Barcelona Pavilion Dataset is a realistic synthetic dataset created for validating the present approach with complete control over illumination, 3D geometry and cameras. This dataset was used together with its 3D mesh ground truth to evaluate the geometric accuracy of the pull pipeline. These datasets are challenging because they feature multiple objects, self-occlusions, various surface properties and materials. Most energy-minimization approaches are not suited for this kind of data, and PMVS performance has
limitations as illustrated in our results.

**Qualitative Evaluation:** Figure 3.15 and Figure 3.16 show our algorithm’s reconstruction and compares it to PMVS [Furukawa and Ponce (2007)] on two different datasets. Observe that the reconstructed surface patches are glued onto the 3D drawing so that the topological relationship among surface patches is explicitly captured and represented. A key point to keep in mind is that the two approaches are not compared to see which is better. Rather, the intent is to show the complementary nature of the two approaches and the promise of even greater performance when appearance, the backbone of PMVS, is integrated into our approach.

**Quantitative Evaluation:** The algorithm is quantitatively evaluated in two ways. First, we assume the input to the algorithm, the 3D curve drawing, is correct and compare ground truth to the algorithm’s results based on a common 3D drawing. Specifically, we manually construct a surface model using the curve drawing in an interactive design and modeling context using Blender. The resulting surface model then serves as ground truth (GT) since it is the best possible expected outcome of our algorithm. Both GT and algorithm surface models are sampled and a proximity threshold is used to determine if a sample belongs to the other and vice versa. Three stages of surface reconstruction are then evaluated as a precision-recall curve, Figure 3.17a, namely: (i) All surface hypotheses satisfying formation constraints; (ii) surface hypotheses that survive the occlusion constraint; (iii) surface hypotheses that further satisfy the visibility constraint with duplications removed. The algorithm recovers 90% of the surfaces with nearly 100% precision. The missing surfaces are those that do not occlude any structures, and therefore cannot be validated with our approach. Clearly, the use of appearance would a long way towards recovering these missing surfaces.
Figure 3.15: The left column represents four views of the PMVS reconstruction results on the Amsterdam House Dataset. Observe the wide gaps on homogeneous surfaces. The right column represents the results of our algorithm from the same views, obtained from a set of mere 27 curve fragments and without using appearance. Note that the PMVS gaps are filled in our results. Our algorithm errs in reconstructing the back of the can as a flat surface. This can easily be corrected via integration of appearance cues in the reconstruction process.
Figure 3.16: Visual results for the Barcelona Pavilion sequence. Left column: Result of our lofting-based reconstruction algorithm. Right column: PMVS results on the same sequence. Notice that we have no coverage of the floor surface due to the lack of occlusion information, but that our approach is better at capturing fine geometric detail and features a lot less holes on the surfaces of the primary objects, i.e., the chairs.

Second, we also quantitatively evaluate the algorithm in an end-to-end fashion, including the 3D drawing stage. Since the ground truth surfaces are not available from Amsterdam House Dataset, we resort to using Barcelona Pavilion Dataset, which has GT surfaces. Since this dataset is large, we focus our evaluation on a specific area with two chair objects. We use the same strategy to compare the final outcome of our algorithm, Figure 3.17b. The results show that despite a complete disregard for appearance, geometry of the surfaces together with occlusion constraint is able to recover a significant number of surface patches accurately. The recall does not reach 100% because the ground truth floor surfaces do not occlude any curves and therefore cannot be recovered.
Figure 3.17: (a) The precision-recall curves for Amsterdam House Dataset, corresponding to post hypothesis-formation surfaces (green), confirmed surface (blue), and confirmed surfaces after occlusion-based cleanup (red). These results provide quantitative proof for the necessity of all steps in our reconstruction algorithm; (b) The precision-recall curve for Barcelona Pavilion Dataset, evaluating the geometric accuracy of the entire pipeline.
Chapter 4

Conclusions

We have presented a method to extract a 3D drawing from 2D images as a graph of 3D curve fragments to represent a scene from a large number of multiview imagery. The 3D drawing is able to pick up contours of objects with homogeneous surfaces where feature and intensity based correlation methods fail. The 3D drawing can act as a scaffold to complement and assist existing feature and intensity based methods. Since image curves are generally invariant to image transformations such as illumination changes, the 3D drawing is stable under such changes. The approach does not require controlled acquisition, does not restrict the number of objects or object properties.

A fully-automated dense surface reconstruction approach is developed using geometry of curvilinear structure evident in wide baseline calibrated views of a scene. The algorithm relies on the aforementioned 3D drawing, a graph-based representation of reconstructed 3D curve fragments which annotate meaningful structure in the scene, and on lofting to create surface patch hypotheses which are glued onto the 3D drawing, viewed as a scaffold of the scene. The algorithm validates these hypotheses by reasoning about occlusion among curves and surfaces. Thus it requires views from a wide range of camera angles and performs best if there are multiple objects to afford the opportunity for inter-object
and intra-object occlusion. Qualitative and quantitative evaluations shows that a significant portion of the scene surface structure can be recovered and its topological structure is made explicit, a clear advantage. This is significant considering this is only the first step in our approach, namely, using geometry without using appearance which is the core idea underlying successful dense reconstruction systems like PMVS.

One fundamental limitation of our dense surface reconstruction approach is that we require the scenes or the objects to be imaged from a wide range of viewpoints - ideally at least 270 degrees of coverage around the scene of interest or more. This is because our primary cues for surfaces are coming from occlusion information in the scene, and the input image sequence needs to cover enough ground in terms of different viewpoints for our method to perform best.

Our entire pipeline can be thought of as 3D modeling approach that pushes the limit of what can be inferred about shape using geometry alone, avoiding any appearance-based image cues such as intensity patterns or color. Aside from providing a rich and descriptive new representations for 3D models, this approach also clarifies the most promising points of interaction between traditional, appearance-based multiview stereo methods, and those utilizing geometry instead. A typical user of our system is an artist, graphical designer, computer graphics specialist or a computer vision researcher, who needs fully automated, easy-to-use recovery of 3D information from image sequences or video, and needs to overcome certain limitations of the dense 3D modeling state-of-the-art using cues from input images that are complementary to using similarities between appearance patterns.

The primary future research direction from here is to integrate the use of appearance in the approach presented here, an idea that holds significant promise for refining the surface geometries wherever the lofting results are too simplistic due to missing or non-stationary boundary curves, improving the overall reconstruction quality. Other possible directions are making our system work for less views and smaller curves, as this would enable more
geometries and acquisition conditions; and using higher order geometry in order to further refine the hypothesis testing step involved in nearly all stages of our pipeline. In terms of evaluation, a possible future work is to assess more exhaustively how much precision we can obtain with our system for different ranges of baseline and various kinds of surface geometries.
Bibliography


URL: http://doi.acm.org/10.1145/2261250.2261314


URL: https://www.microsoft.com/en-us/research/publication/building-rome-in-a-day/


URL: http://dx.doi.org/10.1023/A:1011104020586

Blender Online Community (2016), Blender - a 3D modelling and rendering package, Blender Foundation, Blender Institute, Amsterdam.

URL: http://www.blender.org


URL: http://www.sciencedirect.com/science/article/pii/S0097849315000734


URL: http://doi.acm.org/10.1145/1531326.1531334


URL: http://doi.acm.org/10.1145/280814.280826


URL: http://dx.doi.org/10.1117/1.JEI.24.2.023003

Eclectiel, L. and Blender Online Community (2016), Blender BSurfaces – 3D modeling and retopology software, Blender Foundation, Blender Institute, Amsterdam.

URL: http://www.blender.org


URL: http://dx.doi.org/10.1016/j.cviu.2004.03.016


infrastructure using computer vision: Achievements, opportunities, and challenges’, *Advanced Engineering Informatics*.


**URL:** [http://doi.acm.org/10.1145/2614106.2614182](http://doi.acm.org/10.1145/2614106.2614182)


**URL:** [http://dx.doi.org/10.1109/CVPR.2007.383246](http://dx.doi.org/10.1109/CVPR.2007.383246)


**URL:** [http://dx.doi.org/10.1109/TPAMI.2009.161](http://dx.doi.org/10.1109/TPAMI.2009.161)


**URL:** [http://dx.doi.org/10.1007/s00348-010-0985-y](http://dx.doi.org/10.1007/s00348-010-0985-y)


**URL:** [http://dx.doi.org/10.1109/ICCV.2007.4408933](http://dx.doi.org/10.1109/ICCV.2007.4408933)


**URL:** [http://dl.acm.org/citation.cfm?id=2331067.2331084](http://dl.acm.org/citation.cfm?id=2331067.2331084)


**URL:** [http://dx.doi.org/10.1109/CVPR.2007.383195](http://dx.doi.org/10.1109/CVPR.2007.383195)


**URL:** [http://dx.doi.org/10.1109/3DIMPVT.2012.76](http://dx.doi.org/10.1109/3DIMPVT.2012.76)


URL: http://doi.acm.org/10.1145/192161.192233


URL: http://dx.doi.org/10.1007/s11263-007-0055-y


URL: http://www.sciencedirect.com/science/article/pii/0167839695000488


URL: http://liris.cnrs.fr/publis/?id=2407

Lebeda, K., Hadfield, S. and Bowden, R. (2014), 2D or not 2D: Bridging the gap between tracking and structure from motion, in ‘Proc. of ACCV’.


URL: https://books.google.com/books?id=ZrDzCQAAQBAJ


URL: http://dx.doi.org/10.1145/311535.311541


URL: http://www.sciencedirect.com/science/article/pii/S0166361596000826


URL: http://dx.doi.org/10.1109/TPAMI.2007.1143


URL: http://doi.acm.org/10.1145/1057432.1057454


URL: http://www.sciencedirect.com/science/article/pii/S1077316902000060


URL: http://dx.doi.org/10.1109/CVPR.2008.4587458


URL: http://www.sciencedirect.com/science/article/pii/S1045926X15000191

Miller, A., Jain, V. and Mundy, J. L. (2011), Real-time rendering and dynamic updating of

URL: http://doi.acm.org/10.1145/1964179.1964190


URL: http://dx.doi.org/10.1007/s11263-006-9967-1


URL: http://doi.acm.org/10.1145/2021164.2021171


URL: http://www.sciencedirect.com/science/article/pii/S0097849312000519


URL: http://www.sciencedirect.com/science/article/pii/S0167839696000180

URL: http://www.sciencedirect.com/science/article/pii/S0167839600000157

URL: http://dx.doi.org/10.1111/1467-8659.t01-1-00648


URL: http://doi.acm.org/10.1145/1275808.1276429

URL: http://doi.acm.org/10.1145/2766990


URL: [http://doi.acm.org/10.1145/263834.263851](http://doi.acm.org/10.1145/263834.263851)


URL: [http://dx.doi.org/10.1023/B:VISI.0000025798.50602.3a](http://dx.doi.org/10.1023/B:VISI.0000025798.50602.3a)


**URL:** http://dx.doi.org/10.1109/ROBOT.2009.5152473

**URL:** http://dx.doi.org/10.1109/IROS.2010.5651280

**URL:** http://doi.acm.org/10.1145/2560328

**URL:** http://doi.acm.org/10.1145/1057432.1057447


**URL:** http://doi.acm.org/10.1145/2617841.2620723


**URL:** http://dx.doi.org/10.1109/CVPR.2008.4587706


URL: http://dx.doi.org/10.1109/ICCV.2007.4408919


URL: http://dx.doi.org/10.1109/3DIMPVT.2012.28


**URL:** http://dx.doi.org/10.1109/CVPR.2011.5995388


**URL:** http://doi.acm.org/10.1145/359842.359848


**URL:** https://books.google.com/books?id=UwWai1uM-HsC


**URL:** http://dx.doi.org/10.1007/s11263-009-0222-4

Zhang, L. (2013), Line Primitives and Their Applications in Geometric Computer Vision, PhD thesis, Department of Computer Science, Univ.


**URL:** http://dx.doi.org/10.1007/s11263-014-0780-y
URL: http://doi.acm.org/10.1145/1185657.1185673