Interactive Generation of 3D Volumetric Models from Point Clouds: Application to Mughal Architecture

Paper ID: briefs_0071

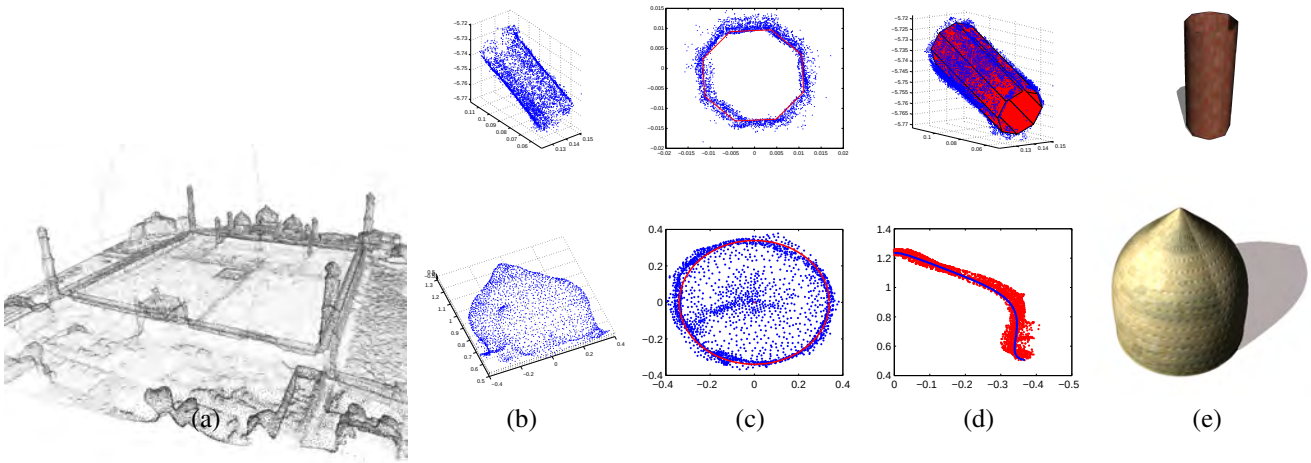


Figure 1: *(a) Input noisy point cloud. (b) Segmented point cloud. (c-d) Primitive fitting on 2D Projection. (e) Final rendered components of model.*

Abstract

We present a volumetric modeling approach to interactively fit 3D primitives to point clouds in order to generate a CAD like model. Our model consists of two categories of volumetric primitives: n-gonal prisms and surfaces of revolution with B-spline profiles. By directly fitting volume primitives, our approach can effectively handle a great degree of noise. Our interactive approach compares favorably both with manually and automatically generated models. Not only is it much more time efficient than manual modeling but also gives significantly better output than state-of-the-art automatic method. Since the focal technique of our approach is the generation of 3D models at a primitive level, our results are ideal in the domain of architecture and preservation of heritage. Modeling of primitives such as domes and minarets allow the user to relate to the very basics of the architecture prevalent in daily lives and amalgamate those perceptions in a virtual 3D model.

CR Categories: I.3.3 [Computer Graphics]: Modeling—Solid Modeling

Keywords: Modeling, 3D architecture, point cloud, CAD

1 Introduction

3D reconstruction of architecture has been a popular research area in computer vision and computer graphics. One way of modeling 3D structures is to use highly sophisticated modeling tools which require significant time and specific human expertise. Automatic approaches, on the other hand can be applied at a large-scale with the cost of compromising precision, quality and detailed features. The key is to strike a balance between automatic and manual approaches that amalgamate in a semi-automatic interactive mechanism reflecting an ideal trade-off between quality and ease of modeling. Over the years, significant advances have been made in this domain in the form of semi-automatic and automatic algorithms for 3D reconstruction using input data acquired from photographs, LiDAR scans or aerial imagery (much of which is stored in the form of point clouds).

Existing approaches are based on the technique of plane fitting and hence, focus primarily on planar structures [Sinha et al. 2008; Furukawa and Ponce 2010; Arikan et al. 2013]. This approach limits their capability in modeling non-planar components such as arches and domes which are common in heritage structures. Exceptions include [Schnabel et al. 2007] which focus more on CAD-based modeling of mechanical parts, leaving CAD-based modeling of architectural structures from point-cloud still limited to planar structures alone.

Unlike prior work, we have pursued the goal of obtaining 3D CAD like models from a sparse point cloud using the idea of parametric shape fitting. We present a semi-automatic algorithm that takes a sparse, noisy point cloud produced from structure from motion (SfM) as input and outputs a coarse CAD like model by finding the best fit parameters of individual shape primitives and assembling these locally fitted primitives together. Shape fitting possesses a greater power of generating a complete and solid CAD-like model as its closed nature and geometric regularities can be exploited at multiple levels. Moreover, unlike other approaches which depend on planar structural configurations, our approach also deals with non-planar elements of architecture, a characteristic feature of heritage architecture. We claim that 3D shape primitives for structural elements like domes, minarets and walls, are more expressive and intuitive for a user, rather than simple planes or curves.

In our work, we have successfully modeled planar structures such as regular n-sided 3D polyhedrons using a RANSAC based novel

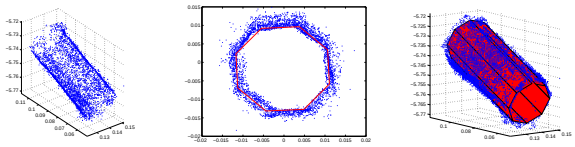


Figure 2: *N-gon fitting pipeline. a) Input 3D point cloud. b) Shape fitting on 2D projection. c) Output 3D n-gon primitive.*

shape fitting strategy as described in section 3.1. For non-planar dome shaped structures, the concept of swept surfaces [Wu et al. 2012] is extended whereby a profile curve, recovered by fitting piecewise Bezier curves, is swept onto a circular transport curve to generate a best fit dome. What sets our algorithm apart from other existing shape fitting approaches is that it can recover the correct parameters of shapes even in the presence of noise, outliers or erroneous shape geometry due to loss of information inherent in SfM point cloud. This is because the correct shape label for individual primitives is known already from user input. Prior knowledge of primitive shape type makes the algorithm robust to any misleading geometry information that may be present in the SfM point cloud as opposed to algorithms that detect shape primitive automatically.

2 Related Work

Digital Preservation of heritage structure is commonly performed by creating 3D Models of the architecture. There is a need for an efficient and automatic alternative that creates such models without incurring the cost of time and labor required for manual modeling. Digital Preservation is thus commonly performed by automatically generating 3D point cloud data using laser scanning or photogrammetry. Since laser scanners are expensive, photogrammetric approaches have been developed.

In photogrammetry, set of images of a site are first obtained. Point and lines in these images are then manually marked to create CAD-like models as proposed in [Debevec et al. 1996] and [Sinha et al. 2008]. Debevecs seminal work [Debevec et al. 1996] effectively recovers model parameters and camera positions from manual matching of edges on photographs to edges of an approximate geometric model. Though Debevecs work produces phenomenal results, reconstructing complicated structures like Arc de Triomphe and Taj Mahal, the approach is manually intensive. Likewise, [Sinha et al. 2008] employs user-assistance to mark all plane boundaries on a collection of unordered photographs and combines it with the constraints imposed by vanishing points to generate complete, polygonal models of planar buildings. Compared to Debevec's approach, [Sinha et al. 2008] requires relatively less user involvement, but its scope is limited to buildings with planar facades.

Amongst fully-automatic approaches, Schematic Surface Reconstruction [Wu et al. 2012] effectively describes the architecture with a concise network of horizontal transport curves (floor plan) associated with vertical profile curves to form swept surfaces. However, schematic representation too has its limitations as it fails to create a solid, compact 3D CAD-like model, deployable in graphics applications.

Recently, [Arikan et al. 2013] presented the state-of-the-art in semi-automatic modeling tool, O-Snap. O-Snap introduces a pipeline for generating complete, polygonal models that fit onto a SfM point cloud, while employing minimal user interaction. It locally fits planes onto segments of point cloud and automatically discovers adjacency relations, optimized by an interactive snapping tool. However, O-Snap also depends upon the assumption of planarity of

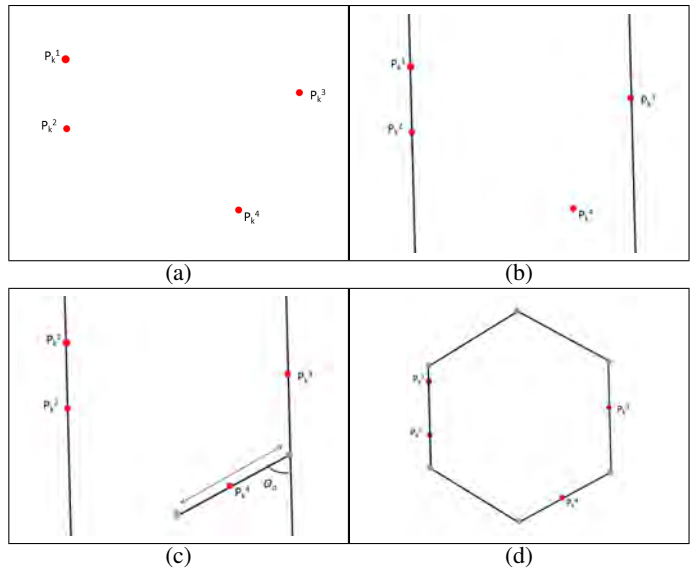


Figure 3: *Visualization of the generate shape hypothesis algorithm as described in Algorithm 1. a) Four initial sample points for n-gon hypothesis generation . b) First two points lie on one side of n-gon and third on a parallel side. c) Fourth point lies on an adjacent line. d) n-gon shape parametrized by the input points.*

all building structures and does not cater for non-planar features of architecture.

In this paper, unlike existing approaches, we propose a novel semi-automatic interactive pipeline that would be capable of modeling non-planar geometries, characteristic of heritage architecture, along with planar geometry, while utilizing minimal user input.

3 Our Approach

Given a noisy point cloud from a state-of-the-art SfM algorithm, and assuming that it has been segmented into its corresponding shape components, the user identifies the primitive type that is to be fitted onto each component. Our shape fitting algorithm then finds the optimal primitive parameters that locally fit this data and produces a CAD like model. We produce solid 3D models for all the individual primitives, but do not apply global continuity constraints on the architecture. Throughout our pipeline, we assume a user marked ground plane which helps us simplify the approach.

3.1 Planar Primitive Fitting

Piecewise planar structures such as hallways and minarets are a major component of Mughal Architecture. We model such closed structures using regular 3D n-gonal prisms. In this paper, we present a novel shape based RANSAC to fit any regular n-gonal primitive onto an input point cloud. We show that in a two dimensional space, all regular n-gons can be represented unambiguously using only four points. This is also supported by the fact that all regular n-gons have only four degrees of freedom in 2D; namely scale, rotation and two parameters of translation.

Four points must be in a specific formation to constrain a regular two dimensional n-gon. To model an n-gon where n is even, we need two points on any one of its side, say l_1. l_1 constrains the rotation of n-gon. Similarly, we need the third point on the side l_2 such that $l_2 \parallel l_1$. Knowing l_2, along with l_1 adds the constraint

scale and one translation parameter of the shape. To constrain the last free parameter of translation, we now need a point on a side adjacent to $l_1 \vee l_2$. Figure 3 shows the n-gon shape generation from four points as described above. This four point approach can be easily adapted for odd n, by restricting the third point to be on some known side instead of parallel one. While picking four points at random does not guarantee that they will lie on the desired configuration of sides, we try all permutations of these points and, in the spirit of RANSAC, simply discard the hypotheses which are not consistent with observed data.

The input point cloud is first projected onto the ground plane. Our RANSAC-based shape fitting algorithm that finds the parameters of input n-sided polygon is then run on this 2D data. The fitted n-gon is extruded along the direction of ground plane normal to obtain the 3D coarse model, fitting the input point cloud segment. This pipeline is illustrated in Figure 2. A similar shape fit algorithm can also be formulated in a three dimensional space using planes and points, but reducing the dimensionality to 2D decreases the complexity of our shape fit algorithm exponentially by imposing fewer degrees of freedom. A visualization of generating shape hypothesis with four sampled points is shown in Figure 3.

3.2 Non-Planar Primitive Fitting

Non-planar structures are usually defined with unique curves. In our work, we focused on non-planar structures which can be described as a surface of revolution around a single axis, such as domes, which are a distinctive feature of Mughal architecture. Adapting from [Wu et al. 2012], we derived a reconstruction pipeline, given in Figure 4, which detects a circular transport curve from ground plane projection and estimates the vertical profile curve associated with it. The estimated 2-dimensional profile curve, which effectively describes the structure curvature, is rotated along the direction of the transport curve to obtain a swept surface.

Given a point cloud segment, with known ground plane, we evaluate its ground plane projection. Using RANSAC, probabilistic-ally optimal center and radius parameters of the circular base are estimated to obtain the axis of rotation for the circular transport curve. Each point on the transport curve t_i has a set of corresponding profile points p_i associated with it. It is possible to evaluate the the final profile curve using only p_i for some specific t_i, but a single slice of profile points is insufficient to reconstruct the profile curve accurately because of noise and missing data inherent in point cloud. Hence, all profiles are projected onto a common plane first and then the final profile curve is modeled using Bezier Splines on that plane. In order to achieve this goal, a plane perpendicular to the ground and passing through the axis of rotation is sliced through the point cloud at transport point t_i as shown in Figure 4 (d). Subsequently, a cluster of profile slices, sampled at an interval of $\delta\theta$, are accumulated by collapsing them onto a common canonical profile plane.

Once all profile points have been accumulated, the profile curve is estimated from plane projected data by estimating a smooth concatenation of cubic Bezier Curves i.e. a B-spline, on profile points as shown in Figure 4 (e). A single Cubic Bezier curve is generally not expressive enough to model most of the curves perfectly. There are two ways of solving the issue, one is to use a higher order Bezier and the other is to concatenate multiple Cubic Bezier curves continuously, which leads to B-splines. As the former method is computationally expensive, the latter is generally used in literature. We used B-splines to model the profile curve as in Figure 4 (e). Rotating the B-spline along the transport curve yields a 3D model as shown in Figure 4 (f).

Algorithm 1 Generate Shape Hypothesis

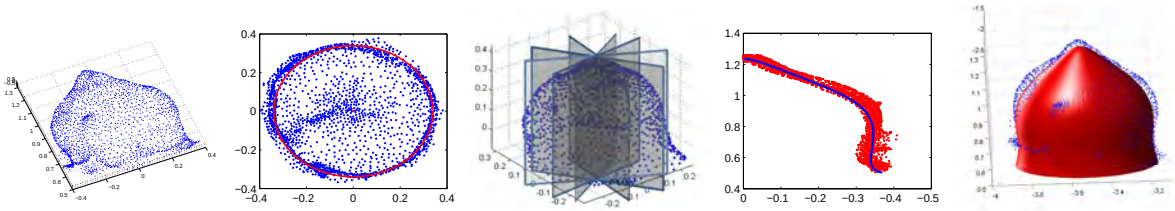
```
procedure GENERATESHAPEHYPOTHESIS(p, n)
▷ p is vector of non-collinear and randomly picked four points
                                    ▷ n is number of sides of the regular n-gon
    θ_0 ← (n − 2) ∗ 180/n                                  ▷ interior angle
    C ← φ                                   ▷ Shape Hypothesis Candidates
    for each permutation P_k of p do
        compute the line l_1 from points P_k^1 and P_k^2
        compute the line l_2 passing through P_k^3 such that l_2 ∥ l_1
        d ← distance_⊥(l_1, l_2)
        compute length of each n-gon side |S| using d and θ_0
        if P_k^4 does not lie on a line adjacent to l_1 ∨ l_2 then
            continue;                                      ▷ Invalid permutation
        end if
        choose line l_m from l_1 ∨ l_2 with minimum distance to P_k^4
        compute l_3 passing through P_k^4 and l_m at (180 - θ_0) °
        C_k^1 ← l_3          ▷ 1^st side of k^th candidate hypothesis C_k
        for i ← 2, n do
            compute gradient of side s_i
            compute intersection point of s_i with C_k^{i−1}
            C_k^i ← s_i
        end for
        if all points in p do not lie on C_k then
            C ← C \ C_k                                    ▷ Remove C_k from C
        end if
    end for
end procedure
```

4 Experimental Set-up and Results

We used an SfM point cloud generated by state-of-the-art Autodesk 123D Catch, using **12** self-taken aerial photographs. The point cloud and corresponding mesh generated were very noisy and had many artifacts as shown in Figure 5(a). This point cloud was then segmented into distinctive structural elements. The dome shown in Figure 5(b) was generated in 8.3 seconds with an input noisy segment of $2,089$ points only. Similarly, the minaret in Figure 5(a) was generated in 128 seconds, and consisted of a hexagonal base with 2,163 points and a dome on top having 302 points. Compared to the automatic model generated by Autodesk 123D Catch, our semi-automated approach provides a regularized hexagonal minaret of much better quality.

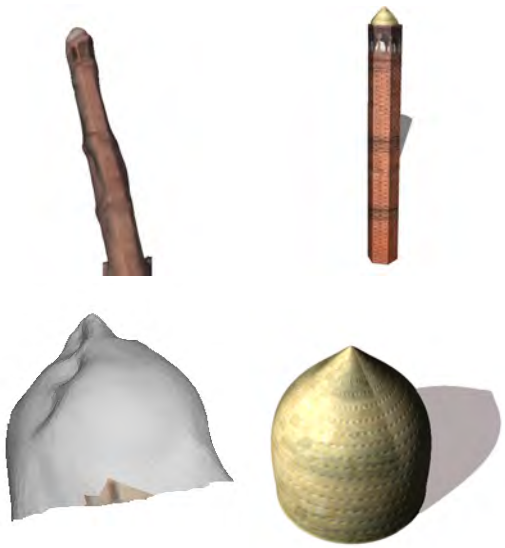
To compare with a manually generated model, we used an ideal synthetic point cloud from an existing CAD model to reconstruct the complete Badshah-i-Masjid, as shown in Figure 6 (Row 1). The synthetic point cloud had $6,27,326$ points and was divided into a total of 27 primitive shapes. The complete model in Figure 6 (Row 1) was generated in around 70 minutes and used only three basic primitive shapes; namely 11 domes, 8 hexagonal prisms and 8 cuboids. While the manual model has much more detail, our semi-automated model is a good enough approximation completed in a fraction of time need to create a model manually.

During experimentation, we used points on the courtyard to mark the ground plane. We restricted the number of RANSAC iterations to be 10% the size of the input data and the minimum size of inlier set for a successful hypothesis was 5% of the input. All these results were generated using Matlab 2012b, running on an intel core i7, 2.67 GHz with 8GB RAM. After generating the best fit primitives, the results were manually texture mapped in Autodesk Maya 2015.



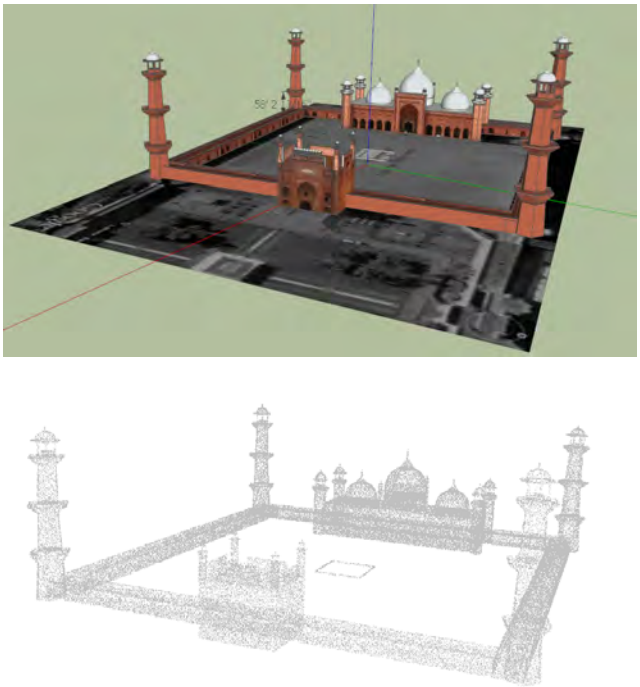
(a) (b) (c) (d) (e)

Figure 4: *Dome fitting pipeline. a) Input 3D point cloud. b) 2D ground plane projection. c) Finding circle parameters of dome transport curve. d) Cluster of profile slices e) B-spline modeling profile curve. f) Output 3D dome model.*



(a) (b)

Figure 5: *a) Model generated using Autodesk 123D Catch. b) Automatically generated CAD-like model using proposed approach.*



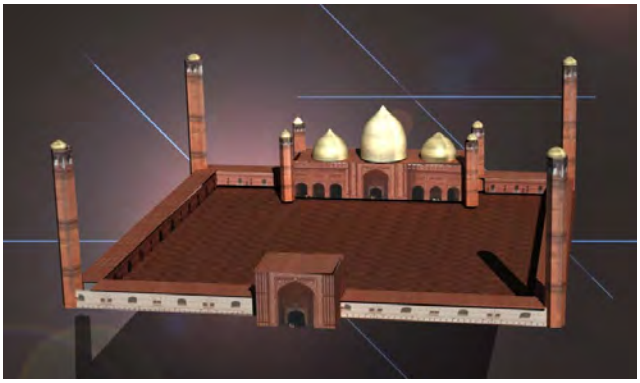


Figure 6: *Row 1) Manually generated model from 3D warehouse-Sketch Row 2) Synthetic point cloud from Sketchup model. Row 3) CAD Model generated using proposed approach.*

References

ARIKAN, M., SCHWÄRZLER, M., FLÖRY, S., WIMMER, M., AND MAIERHOFER, S. 2013. O-snap: Optimization-based snapping for modeling architecture. *ACM Trans. Graph. 32*, 1 (Feb.), 6:1–6:15.

DEBEVEC, P. E., TAYLOR, C. J., AND MALIK, J. 1996. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, ACM, New York, NY, USA, SIGGRAPH '96, 11–20.

FURUKAWA, Y., AND PONCE, J. 2010. Accurate, dense, and robust multiview stereopsis. *IEEE Transaction on Pattern Analysis and Machine Intelligence 32*, 8 (Aug), 1362–1376.

SCHNABEL, R., WAHL, R., AND KLEIN, R. 2007. Efficient ransac for point-cloud shape detection. *Computer Graphics Forum 26*, 2 (June), 214–226.

SINHA, S. N., STEEDLY, D., SZELISKI, R., AGRAWALA, M., AND POLLEFEYS, M. 2008. Interactive 3d architectural modeling from unordered photo collections. *ACM Trans. Graph. 27*, 5 (Dec.), 159:1–159:10.

WU, C., AGARWAL, S., CURLESS, B., AND SEITZ, S. M. 2012. Schematic surface reconstruction. In *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*.