University of Bath

User-Guided Free-Form Asset Modelling

Daniel Beale *

January 23, 2017

Abstract

In this paper a new system for piecewise primitive surface recovery on point clouds is presented, which allows a novice user to sketch areas of interest in order to guide the fitting process. The algorithm is demonstrated against a benchmark technique for autonomous surface fitting, and, contrasted against existing literature in user guided surface recovery, with empirical evidence. It is concluded that the system is an improvement to the current documented literature for its visual quality when modelling objects which are composed of piecewise primitive shapes, and, in its ability to fill large holes on occluded surfaces using free-form input.

1 Introduction

Creating models of existing buildings and man made structures is a problem for a range of different industries such as visual effects, gaming and architecture, and is one of the most important challenges for computer graphics. It is possible for highly skilled artists to reconstruct an existing building if reference photographs and plans are available. This process can be expensive, motivating the search for alternatives that build models directly from image or video data with little or no demand placed on the user.

The need for 3D models from the real world has lead to a wide variety of research and systems. Previous work in this field is not far removed from the paradigm of traditional geometric modelling packages such as Maya[1] and Modo[2]; the user, at a conventional workstation, is able to perform standard operations such as selecting specific points and drawing precise curves. While this workflow leads to models which are visually pleasing from the point of view of an artist or director, they do not easily model the geometry of a real existing scene, leaving room for techniques for speeding up the process through automation.

Standard industrial reconstruction pipelines recreate surfaces from photographs by first building a point cloud, which can be obtained from a number of different technologies such as photographs or range scanners. A common problem among all of these technologies is the undesired production of holes in regions which are occluded, for example, or exhibit a high degree of specular reflection or, indeed, no reflection at all. Autonomous methods for filling holes rely on prediction and prior knowledge of the expected surface shape. User guided techniques are able to draw on human prior experience for identifying surfaces, but also require a way to interact with the system in a way which is not time consuming or difficult.

Rather than motivating the problem through the need for precise models, attention is drawn to the necessity of accurate and fast modelling using current point cloud technology. The demands of modern film production, and increasingly television and gaming, requires pre-visualisation[3][4] ; this is the task of rapidly prototyping and visualising the individual shots of a film before actual production. While pre-visualization has clear demands from a technical point of view; including the need for logistical planning, safety, rigging and setup, and managing costs; there are also many artistic advantages, for example, ensuring that you get the shots you want during production. With the rise of virtual production, the turn-around times between pre-visualisation and production are getting smaller. In the area of 3D modelling, these time pressures give rise to a need for rapid prototyping of geometry that can be used to visualise and plan shots before subsequent detailed modelling for final post-production and composting.

The need for methods which allow a user to quickly build surface models from point clouds are identified, while maintaining robustness to large holes and noise. The new system introduced in this paper allows a user to guide fitting through a collection of free form curves. The workflow first requires the user to highlight required points on the cloud, and then fit a collection of maximum-likelihood primitive surfaces over a number of user interactions. The final surfaces are closed using a well-known and contemporary iso-surface algorithm. It is concluded that the method presented in this paper produces high quality surfaces, and is an improvement beyond contemporary methods when compared on a dataset of man-made objects.

---

[1] http://www.autodesk.co.uk/products/maya/overview
[2] https://www.thefoundry.co.uk/products/modo/
[3] https://www.smpte.org/sections/hollywood/previsualization-current-filmmaking
[4] http://digitalfilmfarmworkshops.com/pre-visualize-directing-video-film/

2 Background

Reconstructing point clouds and surfaces has motivated the fields of computer vision and graphics for many years, and remains an area of active interest. Methods are developed which remain accurate in the presence of noise and occlusion, maintain a low algorithmic complexity and also allow creative but precise user input.

Since surface reconstruction has been successfully used in industry for some time now, the common types of software pipeline are detailed with allusion to creative contributions in the area. A large number of algorithms are documented in order to solve specific problems that a general reconstruction algorithm cannot. To begin with, major, fully autonomous systems for surface recovery from images are detailed, and also, alternative hardware such as range scanners. Methods for improving the process through user guidance are introduced towards the end of the section.

There is a large body of research in to surface reconstruction from point clouds, with algorithms that are used in modelling software such as The Foundry's Nuke[5]. The area has a well formed and standard pipeline, which is useful for the range of different surfaces that it can deal with, and also for its ability to process point clouds which are generated from alternative hardware. The pipeline consists of the following processes,

1. **Point cloud reconstruction**. A dense collection of points is reconstructed from images using available software e.g. [Changchang(2011a),Changchang(2011b),Furukawa(2010),Kolmogorov(2004)]. Alternatively one can use a different sensor such as a range finder, radar or sonar with varying reconstruction qualities.

2. **Normal estimation**. A consistent set of normals are estimated for each point in the cloud. This indicates both where the surface is and also which side of the surface a point is on [Mitra and Nguyen(2003),Hoppe et al.(1992)Hoppe, DeRose, Duchamp, McDonald, and Stuetzle].

3. **Surface fitting**. Finally, an implicit surface; such as a Poisson surface [Kazhdan et al.(2006)Kazhdan, Bolitho, and Hoppe,Kazhdan and Hoppe(2013),Bolitho et al.(2009)Bolitho, Kazhdan, Burns, and Hoppe], a Gaussian Process [Williams and Fitzgibbon(2007),Dragiev et al.(2011)Dragiev, Toussaint, and Gienger] or Delaunay triangulation [Gopi et al.(2000)Gopi, Krishnan, and Silva]; or a parametric surface such as a B-Spline [Krishnamurthy and Levoy(1996),Wang et al.(2006)Wang, Pottmann, and Liu] is fit to the data.

A main alternative to the method described above is to reconstruct a volume or surface directly from a collection of images, without the intermediate point cloud reconstruction. These methods produce excellent results when there are a large number of cameras viewing the object (e.g. [Furukawa et al.(2009)Furukawa, Curless, Seitz, and Szeliski,Campbell et al.(2010)Campbell, Vogiatzis, Hernández, and Cipolla,Starck et al.(2006)Starck, Miller, and Hilton]), but will not work when there are only a few images. They are a good candidate for reconstructing models of objects which are small enough to be viewed from a large number of angles, but are generally unsuitable for medium to large scale objects such as buildings and landscapes.

The primary concern of this paper is with man made objects, on which the assumption of 'piecewise primitive' surface structure is reasonable. It is beneficial to assume that the surface is piecewise primitive if there are large holes in the point cloud or it is extremely noisy, since the assumptions places a large enough constraint that it is possible to predict with a high degree of accuracy in unknown areas. There are methods which are able to fit primitives such as toroids, quadrics, or super-quadrics without user interaction [Debevec et al.(1996)Debevec, Taylor, and Malik,Musialski

---

[5]https://www.thefoundry.co.uk/products/nuke/

et al.(2013)Musialski, Wonka, Aliaga, Wimmer, Gool, and Purgathofer, Bo et al.(2012)Bo, Ling, and Wang]. For example, [Monszpart et al.(2015)Monszpart, Mellado, Brostow, and Mitra] fit a collection of planes to the surface using an energy minimisation framework; [Schnabel et al.(2007)Schnabel, Wahl, and Klein] use probabilistic RanSaC to fit primitives including toroids and cylinders; [Li et al.(2011)Li, Wu, Chrysathou, Sharf, Cohen-Or, and Mitra] optimise using a formulation which minimises global and pairwise energy; while [Labatut et al.(2009)Labatut, Pons, and Keriven] minimise energy using a binary space partitioning tree. For man made objects this approach is reasonable, and well studied, but complicated natural phenomena, such as tree bark or moss, break the assumptions and so produce poor quality results.

Incorporating the use of a small amount of human interaction can be a large benefit to the process (e.g. [van den Hengel et al.(2007)van den Hengel, Dick, Thormählen, Ward, and Torr, Portelli et al.(2010)Portelli, Ganovelli, Tarini, Cignoni, Dellepiane, and Scopigno]). One can provide extra points and normals to surface by selecting a plane which is orthogonal to a point on the medial axis and sketching on it [Yin et al.(2014)Yin, Huang, Zhang, Gong, Cohen-Or, and Chen], before sweeping along the skeleton to fit the surface. Methods which fit a collection of planes to the surface are useful since they easily snap together along the edges [Sinha et al.(2008)Sinha, Steedly, Szeliski, Agrawala, and Pollefeys, Nan et al.(2010)Nan, Sharf, Zhang, Cohen-Or, and Chen]. Using quadric fitting with interaction has been studied in [Andrews and Séquin(2014)]. The authors allow a user to identify a collection of points on a mesh from a single stroke and fit a quadric, displaying triangles which are near the original mesh. [Cao et al.(2014)Cao, Ju, Fu, and Hu] consider objects which are made by extruding 2D profiles, such as furniture and buildings. The work described in [Habbecke and Kobbelt(2012)] use software to apply constraints to designs that are made against pictures, but do not use computer vision in the fitting process.

Alternative to the multiple image / 3D-reconstruction pipeline, shape and volume can be quickly recovered from a single image, without any knowledge of the scene geometry [Lau et al.(2010)Lau, Saul, Mitani, and Igarashi, Chen et al.(2013)Chen, Zhu, Shamir, Hu, and Cohen-Or, Jiang et al.(2009)Jiang, Tan, and Cheong]. In 3-Sweep [Chen et al.(2013)Chen, Zhu, Shamir, Hu, and Cohen-Or] a user can quickly outline contours on objects and recover depth using high level knowledge about object ordering, the resulting models can then be recomposed into different photographs. Methods in this area can quickly recover plausible looking geometry. Without the addition of strong, and potentially unavailable, priors, they do not have the same metric accuracy as a point cloud reconstruction.

It is worth mentioning that editable assets can also be generated from scratch using interaction alone, without any computer vision (e.g. [Chen et al.(2008)Chen, Kang, Xu, Dorsey, and Shum, Lipp et al.(2014)Lipp, Wonka, and Müller, Xu et al.(2014)Xu, Chang, Sheffer, Bousseau, Mccrae, and Singh, Iarussi et al.(2015)Iarussi, Bommes, and Bousseau]). These family of approaches are suited to modelling novel buildings from an architectural standpoint, and belong to a large class of CAD modelling software. The prior use of point-based reconstruction provides a cue to speed up the extraction of surfaces on existing buildings. It should be viewed as a potential aid for current CAD modelling tools, removing the need for intensive user interaction where a 3D model already exists.

The system presented in this paper follows the standard modelling pipeline described at the beginning of this section. A point cloud is reconstructed first, before any user interaction, to extract points for fitting geometric primitives. This places the work in to a mainstream research area, allowing it to be compared to fully autonomous point cloud techniques, and also interactive multiple view surface reconstruction algorithms.

The literature arranged in Table 1 is a representative sample of the current state of the art in interactive surface recovery. It compares the system presented in this paper to the most recent and most similar alternative systems in the area. It can be seen that while [Sinha et al.(2008)Sinha, Steedly, Szeliski, Agrawala, and Pollefeys] and VideoTrace [van den Hengel et al.(2007)van den Hengel, Dick,

	Fit type	Input
Sinha et al.	Planes	Polygon tool
VideoTrace	B-Splines	Spline tool
Morphit	B-Splines	Free-form
This system	Quadric forms	Free-form

Table 1: A comparison of some relevant, and current, interactive techniques for surface reconstruction.
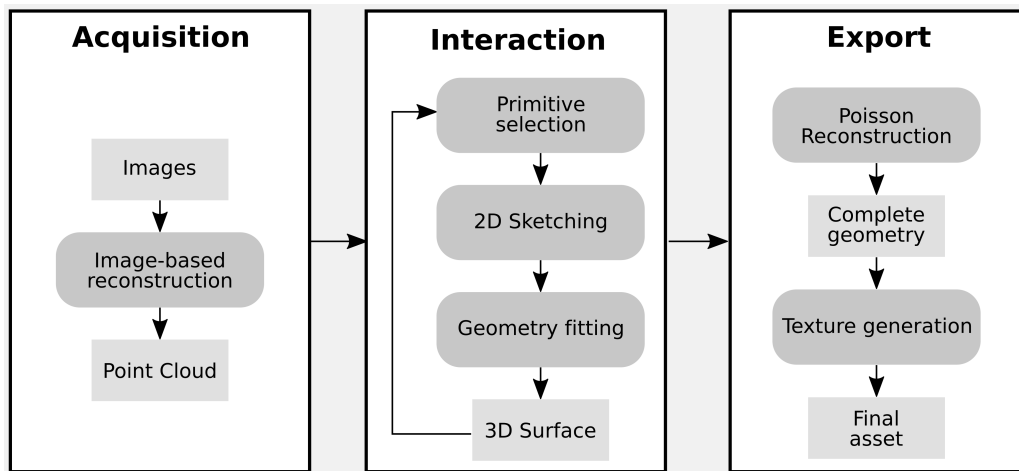

Figure 1: A schematic diagram of the system.

Thormählen, Ward, and Torr] are able to fit primitive parametric models to the data, they cannot deal with free-form input. The method in Morphit [Yin et al.(2014)Yin, Huang, Zhang, Gong, Cohen-Or, and Chen] allows a user to improve implicit surface fitting using free-form input, which works well on complicated surfaces but cannot predict when there are large holes in the surface. While there does not appear to be any previous work that uses rough, or loose, annotations for primitive geometry recovery, there are other applications that have similar interfaces for related editing tasks such as colorization [Levin et al.(2004)Levin, Lischinski, and Weiss] and segmentation [Subr et al.(2013)Subr, Paris, Soler, and Kautz].

The limitations of contemporary work is given a more detailed review in the results section, with some specific examples and a more in depth discussion. It is concluded that the method presented in this paper is a contribution to the area of free-form modelling with geometric primitives.

3   System Overview

The system assumes the existence of a point cloud with a collection of calibrated images. The user is presented with two panes, one which displays an interface with the image and allows a frame number to be selected, and the other containing the point cloud and tools to view the cloud from an arbitrary pose and scale. Sketches can be drawn on the image pane with colour selection representing curves which correspond across each of the views.

A schematic diagram for the system is presented in Figure 1. The point clouds are generated through image based reconstruction, using the well known and freely available software Visu-

	Point selection		**Surface fitting**

Quadric

LVM

Figure 2: An example of some sketch input to the system and the corresponding output from the primitive fitting algorithms. The first row contains an example of fittinAn example of someg an ellipsoid to a column of a building, and the second row is an example of a perpendicular extrusion using a pair of latent variable models.

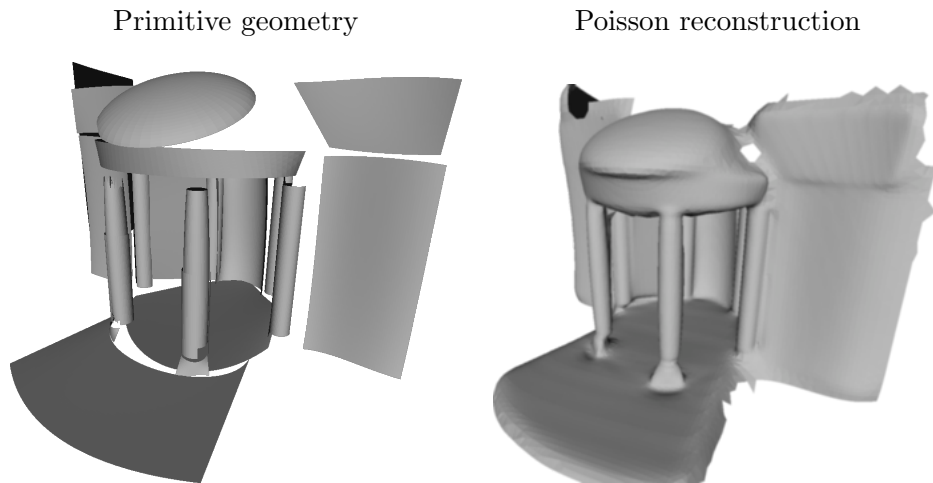**Primitive geometry**   **Poisson reconstruction**

Figure 3: After iteratively selecting points in the original cloud, system produces a collection of primitives which form the surface of the required object. The left image shows an example of the extracted geometry, and the right image shows the mesh after it has been closed using Poisson reconstruction.

alSFM [Changchang(2011a), Furukawa(2010)]. The rest of this paper describes the interactive modelling, which makes up the principle contribution. The meshes are finalised with a Poisson reconstruction [Kazhdan et al.(2006)Kazhdan, Bolitho, and Hoppe] and then textured.

The system is a free form, colour coded, tool for highlighting corresponding trajectories. The trajectories enable a sub-selection of the point cloud to be extracted for more advanced fitting. The types of surface that can be fit are first classified into 'whole surface fitting' and 'combining curves'. The former reduces the point cloud to a single point sub-cloud, from which a surface derived from a
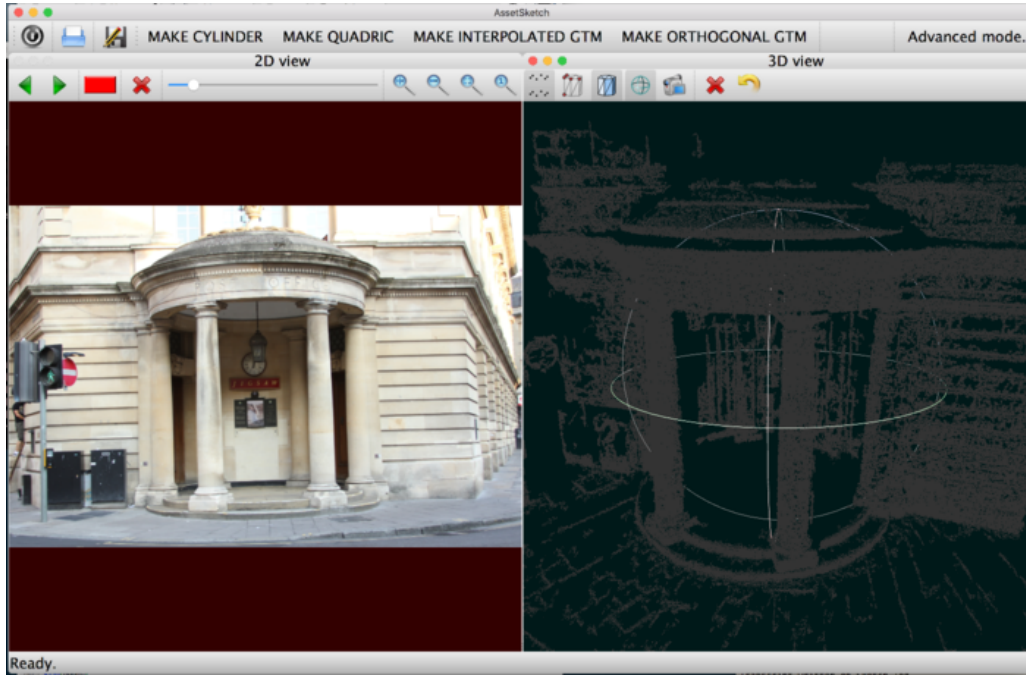
Figure 4: The graphical user interface.

quadric is fitted, either an Ellipsoid or Cylinder. The latter takes two point sub-clouds representing two curves in three dimensions and combines them through perpendicular extrusion or interpolation. The categories are labelled below,

1. **Whole surface fitting**. A single point cloud is derived from the user sketches, from which the algorithm fits, either,

   (a) an **Ellipsoid**,

   (b) or a **Cylinder**.

   The method for fitting a quadric is described in Section 5, with a discussion on how to obtain the primitive surfaces from the parameters.

2. **Combining curves**. A pair of point clouds are derived from the user sketches, representing a pair of curves in 3D. The curves are then combined to form a surface through;

   (a) **Perpendicular extrusion**, one of the curves is copied along the other to form the vertices of a surface;

   (b) or, **Interpolation**, where the curves form the ends of the surface and intermediate vertices are interpolated.

   The latent variable model for the curves is described in Section 6, from which the surfaces defined above are created.

The graphical user interface is displayed in Figure 4. The user is presented with a panel for sketching on camera views on the left, and with the point cloud on the right. Some examples of fitting primitives to the point cloud using the interface are given in Figure 2. The top row shows an ellipsoid being fit to the data from two curves on two views, and the bottom row shows a perpendicular

extrusion. The process is refined over a number of iterations to produce an output such as on the left of Figure 3. The surface is saved as a collection of meshes which are then processed by a Poisson surface reconstruction algorithm, in order to fill the gaps between the surfaces; displayed on the right hand side of Figure 3.

The method of free-form point selection is first described in Section 4, before the various fitting algorithms. The curves are modelled using a latent variable model, this is described in Section 6 along with the methods for reconstructing surfaces from curves. The ellipsoid and cylinder fitting algorithms are described in Section 5.

4 Free-Form Point Selection

The problem of selecting points is difficult if both the input and the point cloud is contains noise. It is natural to appeal to Bayesian statistics, since it is a well established framework for these kinds of problem. Although there are many different types of noise, it is assumed that the user-input and point cloud exhibit additive Gaussian noise, and the corresponding points that were selected in the cloud are extracted by thresholding the probability distribution.

The interface takes a pen width and colour, and allows a user to draw a profile across the image. By marking the image the user is supplying the likelihood that a pixel belongs to a group of 3D points that are of interest in the point cloud, either because the pixels represent the boundary between surfaces or fall within the interior of the surface. It is shown how to combine these distributions in order to derive a probability that each point in the cloud belongs to a sketch. A threshold can then be applied to the density in order to extract points of interest.

Let the collection of all images be denoted $F = \{f_1, \ldots, f_N\}$. The subset of images that have been marked are denoted U, so that $i \in U$ refers to image f_i. Typically only a few images are required for marking and so the size of U is small compared to F, particularly in the case of a video.

Points along the user sketch are denoted $\boldsymbol{\mu}_{ij}$; this is the jth point in image f_i. Each point has an associated covariance $\Sigma_{ij} = \sigma_{ij}^2 I$; the σ_{ij} represents the uncertainty in the estimated edge location and are chosen to be the brush width in the user interface. If we now consider an arbitrary pixel $\boldsymbol{y} \in \mathbb{N}^2$, in an image. the likelihood that this pixel belongs to the mark drawn by the user is modelled as a Gaussian mixture,

$$p(\boldsymbol{y}|f_i) = \frac{1}{K_i} \sum_{j=1}^{K_i} \mathcal{N}(\boldsymbol{y}|\boldsymbol{\mu}_{ij}, \Sigma_{ij}) \tag{1}$$

Now suppose that we have an arbitrary 3D point $\boldsymbol{z} \in \mathbb{R}^3$, and a collection of cameras $\Pi = \{\pi_1, \ldots, \pi_{|U|}\}$,

$$\pi_i : \mathbb{R}^3 \to \mathbb{R}^2 \tag{2}$$

$$\boldsymbol{z} \mapsto \frac{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} P_i \begin{bmatrix} \boldsymbol{z} \\ 1 \end{bmatrix}}{(P_i \begin{bmatrix} \boldsymbol{z} \\ 1 \end{bmatrix})_4}. \tag{3}$$

where $P_i \in \mathbb{R}^{3 \times 4}$ is the ith camera matrix.

We wish to find the overall probability of the point in the cloud $\boldsymbol{z}_k$, where $0 < k < N$. In order to do it we assume that the sketches are conditionally independent given the projection functions and

the 3D point, in which case the distribution can be written as follows,

$$p(\boldsymbol{z}_k|\Pi) = \frac{1}{\mathcal{C}} \prod_{i=1}^{|U|} p(\pi_i(\boldsymbol{z}_k)|f_i) \qquad (4)$$

The normalising factor $\mathcal{C}$ is then the sum over each point in the cloud,

$$\mathcal{C} = \sum_{k=1}^{N} \left(\prod_{i=1}^{|U|} p(\pi_i(\boldsymbol{z}_k)|f_i) \right) \qquad (5)$$

We use this probability to identify 3D points among the million or so points in the cloud. Every point in the cloud is assigned a probability, many of them very low. We simply keep all of the points in the cloud which have probability larger than the mean probability.

It would be possible to include further interaction to identify the surface normals. This is not necessary for this method since the positions of the cameras are known. The interior of the surface is set so that all of the normals are pointing towards a candidate camera.

5   Fitting a Quadric

The first category of point extraction is termed 'whole surface fitting', in which the user selects a region which contains points that belong to a whole surface. Either an ellipsoid or a cylinder is then fitted. Both of these surface types can be neatly represented as a quadric. A full and detailed explanation of the method can be found in [Beale et al.(2016)Beale, Yang, Cosker, and Campbell]. The salient points are described in this section, along with the method that is used for trimming the final surfaces.

Quadrics are implicit functions, specifically, solutions of the equation,

$$\boldsymbol{z}^T A \boldsymbol{z} + \boldsymbol{b}^T \boldsymbol{z} + c = 0 \qquad (6)$$

with $A \in \mathbb{R}^{3\times3}$ matrix which is symmetric positive definite, $\boldsymbol{b} \in \mathbb{R}^3$ a vector and $c \in \mathbb{R}$ a scalar.

In three dimensions A and $\boldsymbol{b}$ can be written as follows,

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \qquad (7)$$

We let,

$$\boldsymbol{\psi} = [a_{11}, a_{22}, a_{33}, a_{12}, a_{13}, a_{23}, b_1, b_2, b_3, c]^T \qquad (8)$$

Suppose that the matrix of data points are denoted $Z = [\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N]$ with the kth column denoted $\boldsymbol{z}_k \in \mathbb{R}^3$, and let the jth element of $\boldsymbol{z}_k$ be z_{jk}. We create a vector of monomials for each data point as follows,

$$\begin{aligned} \boldsymbol{x}_i &= [\quad z_{1i}^2, z_{2i}^2, z_{3i}^2, 2z_{1i}z_{2i}, 2z_{1i}z_{3i}, 2z_{2i}z_{3i}, \\ &\quad z_{1i}, z_{2i}, z_{3i}, 1]^T \end{aligned} \qquad (9)$$

If we define a monomial data matrix $X = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]$, the maximum a-priori solution for the parameters is,

$$\boldsymbol{\psi} = \left(\sigma^2 X^T X + \begin{bmatrix} J & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \right)^{-1} \begin{bmatrix} \mathbf{1} \\ \mathbf{0} \end{bmatrix} \qquad (10)$$

(a) Point cloud   *(b) Points selected by user sketches*   *(c) Recovered surface*
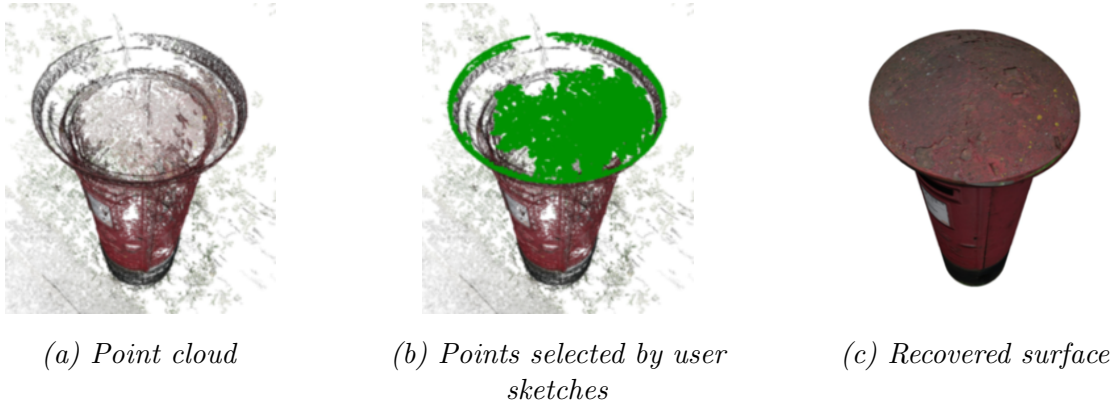
Figure 5: An example of fitting a quadric to a point cloud. (a) The point cloud. (b) Points selected by user sketches are shown in green. (c) The final quadric surface fit to the point cloud. The surface extends to the extremal points selected by the users sketches or curves.

Where σ is a prior parameter which controls how close to a sphere the quadric is. The smaller it is, the more likely that the quadric will take the shape of a sphere. This method is useful since it prevents the quadric from forming a hyperboloid which is made of two disconnected surfaces. Usually when fitting primitive geometry we are looking for a single closed surface, and the best fit to be a contiguous collection of points.

In practise, the surfaces must be trimmed so that they do not go too far beyond the point cloud. This is done by considering the eigen-decomposition of the matrix A. Considering the expansion,

$$(\boldsymbol{x} - \boldsymbol{\mu})^T A (\boldsymbol{x} - \boldsymbol{\mu}) = \tau \tag{11}$$

$$\iff \boldsymbol{x}^T A \boldsymbol{x} - 2\boldsymbol{\mu}^T A \boldsymbol{x} + \boldsymbol{\mu}^T A \boldsymbol{\mu} \tag{12}$$

and taking $\boldsymbol{b} = -2A\boldsymbol{\mu}$ and $c = \boldsymbol{\mu}^T A \boldsymbol{\mu}$, it can be seen that Equations 11 and 6 are equivalent. Removing the mean and projecting the points onto the principal vectors provides a way to determine how much of the quadric is covered by the points. To trim the final surface we simply remove vertices and faces which are outside the maximum and minimum projected values.

Figure 5 shows an example of fitting a quadric surface to a set of 3D points selected from the point cloud.

6   Fitting a Latent Variable Model

The second categories of point extraction is termed 'combining curves'. In this mode the user selects a pair of curves in the interface which are then combined either through interpolation of perpendicular extrusion in order to create the surface. This section describes the method used for fitting a parametric function to the curves and then combining them in to a single surface.

One of the principal difficulties in fitting a parametric model to a point cloud is that the points are noisy and unordered. We suppose that the points relate to the output of a continuous function, of a single variable. In other words given a value t we would like to be able to compute an arbitrary point on the curve. If we know how the value of t relates to the output, the problem is one of standard regression, if we do not, the t values are unknown or hidden, and are termed 'Latent Variables'. A number of different methods for extracting latent variables and fitting parameters are documented in [Lawrence(2004)]. Following the work of [Bishop et al.(1998)Bishop, Svensén, and Williams], we
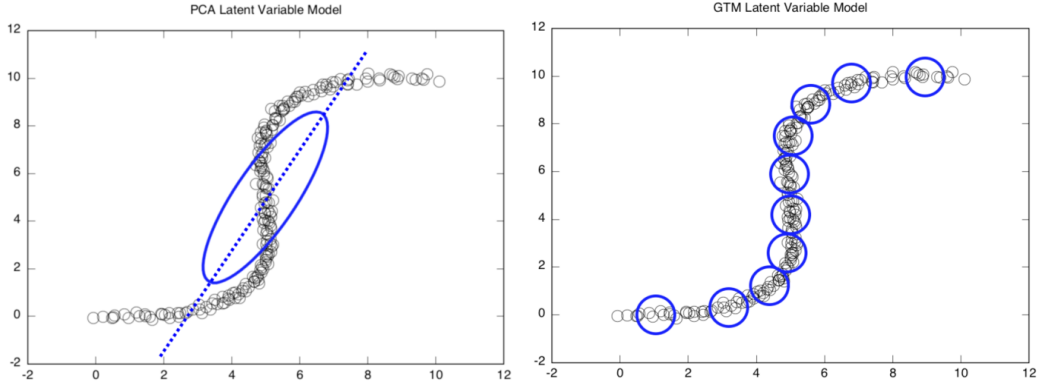
Figure 6: An example of multivariate polynomial regression using both Principle Component Analysis (left) and a Generative Topographic Map (right). The black points are the noisy point cloud in 2D, the blue ellipses represent Gaussians.

make use of the Generative Topographic Map (GTM), since it is flexible to a range of different types of function, requiring only a finite set of basis vectors.

There are several different types of curve, such as B-Splines, Bezier curves, or linear combinations of sinusoids or exponential kernels. A multidimensional polynomial is chosen since it is simple and portable. Letting $\psi_d \in \mathbb{R}^L$ denote the basis vectors for $0 < d < D$, and the matrix $\Phi = [\phi_0, \ldots, \phi_D]$. For a polynomial basis this is written as follows,

$$
\Phi = \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \\ 0 & 1 & 2 & \ldots & D \\ \vdots & \vdots & \vdots & & \vdots \\ 0^L & 1^L & 2^L & \ldots & D^L \end{bmatrix} \tag{13}
$$

$$ \tag{14} $$

Given a weight matrix $W \in \mathbb{R}^{3 \times L}$, the product $W\Psi$ is then a collection of polynomials evaluated at each of the points $0, \ldots, D$; with coefficients in W. The value of D can be chosen by the user, but is generally fixed at about 50. The values of $W\Psi$ are mapped in to the 3D point cloud space, defining the equidistant centres of a Gaussian mixture along the polynomial. The model is best represented by Figure 6 (right) where the black points are unordered 2D data distributed along the curve, and the curve itself is shown in blue.

The constrained Gaussian mixture model, for the collection of points $Z = [Z_1, \ldots, Z_N]$ is given by,

$$
p(Z|\Phi, W) = \prod_{k=1}^{N} \left(\sum_{j=1}^{D} \omega_k \mathcal{N}(Z|W\phi_j, \sigma^2 I) \right) \tag{15}
$$

The mixture is solved for the parameters W and σ by finding the derivatives of the log-likelihood of Equation 15. The solution is a fixed point iteration, which can be formulated as an Expectation-Maximisation algorithm as follows. We first compute the expected probability of a point given the parameters and the data,

$$
R_{ij} = \frac{\omega_i \mathcal{N}(Z_j|W\phi_i, \sigma^2 I)}{\sum_{m=1}^{D} \omega_m \mathcal{N}(Z_j|W\phi_m, \sigma^2 I)} \tag{16}
$$

10

The maximisation step finds the maximum likelihood values for the parameters given the expected probabilities,

$$W_{new}^T = (\Phi G \Phi^T)^{-1} \Phi R Z^T \tag{17}$$

$$\sigma^2 = \sum_{i=1}^{D} \sum_{j=1}^{K} R_{ij} ||W_{new}\boldsymbol{\phi} - Z_j||^2 \tag{18}$$

where G is a diagonal matrix with elements $G_{ii} = \sum_n R_{in}$.

Since this procedure is iterative it requires an initialisation of the variables W and σ. This is done using principle component analysis. Consider a line defined by the function $f(t) = \boldsymbol{n}t + \boldsymbol{a}$, and a Gaussian which has been fit to the data defined by by parameters $\boldsymbol{\mu}$ and Σ. We would like the line to pass through the mean of the data and along the largest principle component of Σ. This is shown graphically on the left of Figure 6, the unordered data is represented by the black circles and we would like to fit the line represented by the blue dashes. The parametrisation of f makes makes this easy: set $\boldsymbol{a} = \boldsymbol{\mu}$ and let $\boldsymbol{n}$ be the vector corresponding to the largest eigenvalue of Σ. In the notation of the above the weight matrix is then taken to be $W = [\boldsymbol{a}, \boldsymbol{n}, \boldsymbol{0}, \ldots, \boldsymbol{0}]$, and the variance is initialised as the smallest eigenvalue of the covariance matrix.

Generally speaking, the ends of the curve do not need to be computed since the algorithm converges to fit the points precisely. It is possible that the shape of the curve causes the Gaussians to be centres beyond the points. In this case the GTM is trimmed by removing components at either end which have a value of $\frac{1}{K}\sum_{k=1}^{K} R_{ij}$ that falls below a threshold, or in other words, Gaussian components are removed from the ends which do not have any points associated with them. Doing this keeps the resulting curve tight to the original data, even after convergence.

The final surface is built from the curves. Either the surface is an interpolation of two drawn edge curves, or one curve is extrapolated in the direction of a perpendicular second curve. In practise, drawing a profile and extrapolating it along an edge gives very good results; a similar approach was taken in 3-sweep [Chen et al.(2008)Chen, Kang, Xu, Dorsey, and Shum].

Letting $\boldsymbol{q}_i, \boldsymbol{p}_i \in \mathbb{R}^3$ denote two extracted curves, and $\boldsymbol{s}_{kj}$ be a set of vertices on the desired surface, for $0 < i, j, k \leq D$. We first reorder the curves so that they are oriented in the same direction; if $||\boldsymbol{q}_0 - \boldsymbol{p}_D|| < ||\boldsymbol{q}_0 - \boldsymbol{p}_0||$ then the curve $(\boldsymbol{p}_j)_{j=1}^{D}$ is reversed. The interpolation is then performed using the following equation,

$$\boldsymbol{s}_{kj} = \left(\frac{j}{D}\right)\boldsymbol{q}_k + \left(1 - \frac{j}{D}\right)\boldsymbol{p}_k \tag{19}$$

If the curves are perpendicular then the curves are reordered in the same way as above. One side of the surface is initialised with a trajectory and then copied and translated it according to the velocities of the second. This method avoids the requirement of computing curve intersections, which may be difficult if the data is noisy. The surface vertices are as follows,

$$\boldsymbol{s}_{kj} = \begin{cases} \boldsymbol{q}_k & : j = 0 \\ \boldsymbol{q}_k + \boldsymbol{p}_j - \boldsymbol{p}_{j-1} & : j > 0 \end{cases} \tag{20}$$

An example of fitting a perpendicular GTM surface is shown in Figure 7.

7   Finalising the surface

The system described in previous sections provides a way for a user to fit geometric primitives to a point cloud. It remains to fill in the holes in the surfaces.

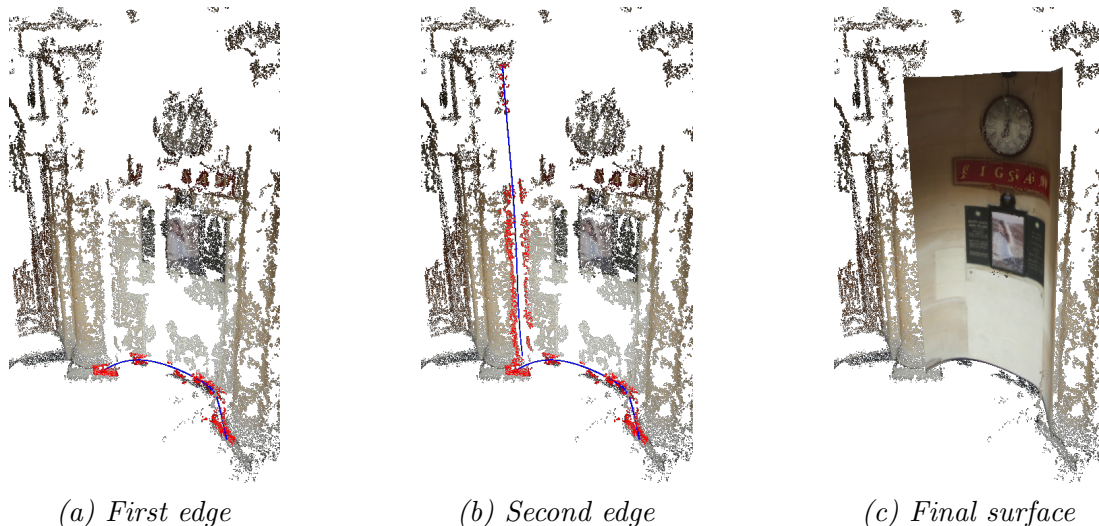*(a) First edge*   *(b) Second edge*   *(c) Final surface*

Figure 7: An example of fitting a perpendicular GTM surface from a pair of curves. A single collection of sketches define each of the edges, and the first edge is extruded along the second edge to create the final surface.

To close the holes between the primitive surfaces the Poisson reconstruction algorithm is used, as described in [Kazhdan et al.(2006)Kazhdan, Bolitho, and Hoppe]. An open source implementation is available in the software package MeshLab [Cignoni et al.(2008)Cignoni, Corsini, and Ranzuglia]. Since the output from the software is a collection of meshes with normals no pre-processing is required, and a closed surface is fit to the geometry. The mesh is completed by removing all of the vertices and faces which have long edges. This is possible since the Poisson reconstruction tends to produce large triangles in regions where it is predicting.

8 Results

The results in this section present comparisons against state of the art alternatives for reconstructing surfaces from point clouds. It is difficult to assess the quality of the output without first establishing a high quality ground truth. The problem of overfitting in statistical regression motivates an argument that, even with a good quality ground truth, the quality of a result is difficult to measure, and ultimately lies with the perceiver. Another difficulty in assessing the results is in the human interface. The competing methods require user interaction in order to fit the surfaces. Rather than make arguments based on the effort or complexity of the user input, a single working example is chosen with rational arguments about the limitations of the methods. It is argued that each of the competing softwares are fit for the purpose that they were designed, but fail on the dataset that we have, making the method presented in this paper a reasonable replacement.

The interactive methods that compared against are, VideoTrace [van den Hengel et al.(2007)van den Hengel, Dick, Thormählen, Ward, and Torr] and Morfit [Yin et al.(2014)Yin, Huang, Zhang, Gong, Cohen-Or, and Chen], both of which are contemporary, and present an alternative approach to surface reconstruction. The comparison is made on a point cloud created from photographs of a building with columns and a non-planar facade. A large portion of the entrance is occluded and there are holes in the point cloud as a result of it. These problems make it very difficult to automatically reconstruct the surface, and a user must provide some idea of where the surfaces should be. This is the example that is used to make the initial comparisons in Sections 8.1 and 8.2.

The principal argument for using the method presented in this paper is that a large percentage of man-made objects are composed of a collection primitive surfaces. The final set of results show the performance on a dataset containing 6 examples of buildings and outdoor furniture, the comparisons provide a visual assessment of the improvement beyond the standard method of normal estimation and Poisson reconstruction.

8.1 Comparison against Morfit

The Morfit [Yin et al.(2014)Yin, Huang, Zhang, Gong, Cohen-Or, and Chen] software package generates a surface based on the morphological skeleton of the object. The software allows a user to edit the skeleton, and then sketch extra vertices and normals on a plane which is orthogonal to an arbitrary point on the skeleton, before fitting surface. So long as the skeleton is a good enough prediction the surface will allow a user to specify the surface shape in regions where there are large holes.

Point cloud   Skeleton
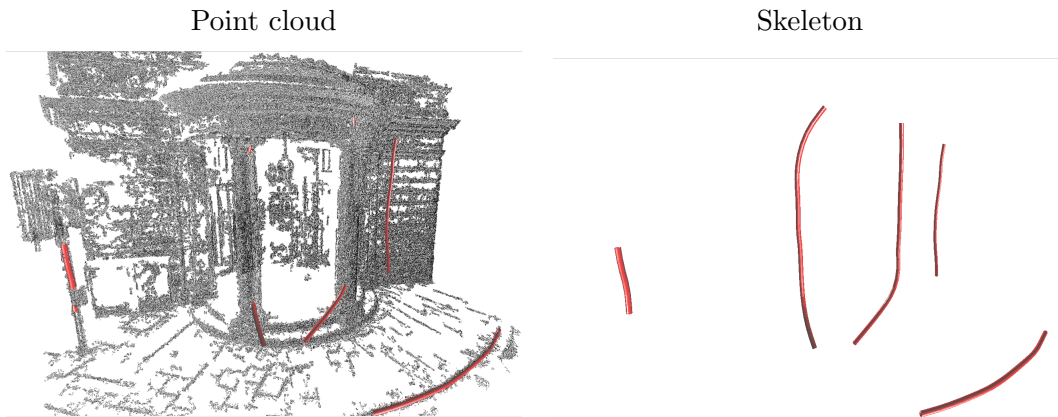

Figure 8: An example of a skeleton fitted to point cloud of a building with a non-planar and partially occluded facade.

The principle limitation of Morfit is that it requires a morphological skeleton in order to compute the surface. Consider the point cloud shown in Figure 8. The building has 6 columns arranged in a cylinder, the occluded entrance is cylindrical and each of the walls are planar. The right hand side of Figure 8 shows the automatically extracted skeleton which correctly identified the two front columns, but was unable to find a skeleton for other features such as the back wall. In fact, a skeleton does exist for this building. It is part of a larger closed structure which is only partially visible. Using the Morfit interface to fix the skeleton is extremely time consuming, and running the sweeper on the resulting branches yields poor results.

The dataset that Morfit was tested on is primarily made up of models which are small and closed, such as the model Gekko featured throughout the paper. The algorithm works well on this data, since the skeleton is much easier to compute and the reconstructed points are evenly distributed along the object.

8.2 Comparison against VideoTrace

VideoTrace is an interactive software package for reconstructing 3D surfaces from a collection of photographs of an object from multiple views. It first reconstructs a sparse point cloud and then allows a user to draw a polygon or non-uniform rational b-spline in three dimensions using the interface. The

VideoTrace

| View 1 | View 2 | Reconstruction |

This system

| View 1 | View 2 | Reconstruction |
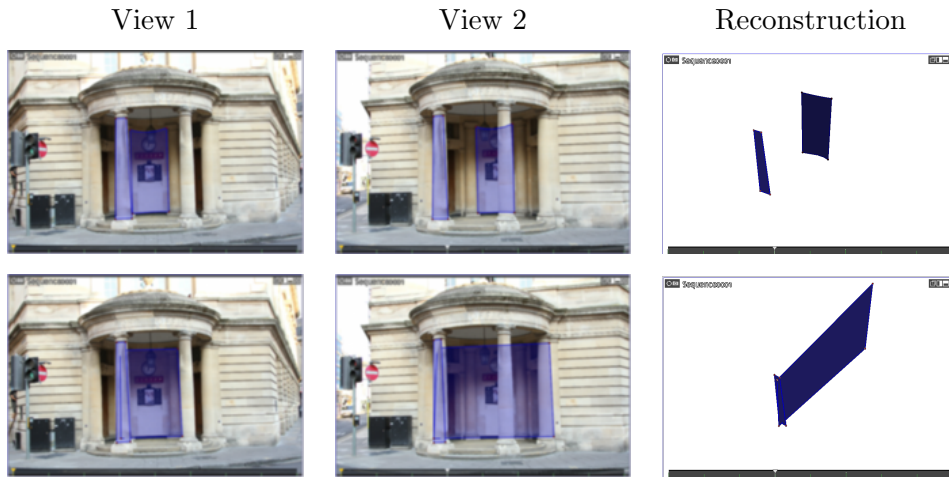
Figure 9: Example output from video trace. The first row shows a successful surface reconstruction of an occluded face, and the second row shows an unsuccessful reconstruction of the same face. The final row is the output from the system presented in this paper.

curves can be joined to create a surfaces, which can then either be extruded or mirrored two create a more complicated closed surface.

One of the principal limitations of this system is the way in which the user identifies the 3D surface. Assuming that the camera geometry and point cloud have been accurately reconstructed, the system takes a polygon from a single view and uses the point cloud to infer where the polygon lies in 3D space. The user can then refine the fit by moving control points from different views. This presents a problem if there is a surface which is occluded, such as the post office canopy in Figure 9. The column has been highlighted for comparison. On the top row the occluded surface was identified by creating a polygon on the left hand image, the computed geometry was correct in the second view, and also in the 3D reconstruction on the right hand side. The second row shows this method to fail when the user wants to extract the larger segment on the occluded face. Again, the polygon was identified in the left hand column but is significantly erroneous in both the second view and the 3D reconstruction on the right hand side. The method presented in this paper correctly identifies the surface, even though we have sketched over an occlusion, and is displayed on the bottom row.

8.3 Qualitative results

We provide some final comparisons on a dataset of objects, all of which are man-made objects in an urban environment. The dataset contains building which have a surface structure which is not restricted to planes alone, making a large number of traditional methods inapplicable.

The results can be visually compared in Figure 10. It can be seen that the Poisson reconstruction fails where there are large holes in the surface such as the floor of the post-office, or on the roof of the phone booths. The final textured meshes are shown in Figure 11.

The software is demonstrated in the supplementary material. A candidate user is recorded fitting primitives using only free-form input. The videos demonstrate the ease of use for both latent variable models and quadric surfaces, on a touch screen display. The use of Bayesian statistics makes the system robust to noise in the input trajectories, resulting in geometry which is a good fit to the data.

9   Conclusion

The system that has been presented in this paper enables a user to quickly fit parametric surfaces to point clouds. It is demonstrated that we outperform alternative methods on a dataset of medium to large scale objects such as buildings and street furniture, which are man-made and comprise surfaces which are piecewise primitive. Alternative interactive methods either do not make the correct assumptions about the data or do not correctly deal with occluded surfaces, making them laborious for desktop use by an artist.

The system performs well on data which exhibits a large amount of noise, and has holes in the point cloud; a limiting artefact for modern autonomous reconstruction techniques such as Poisson reconstruction.

While the system does perform well under the assumptions, it is not expected to work well if they are broken. Highly complicated natural objects such as flowing water, vegetation and so on, require an algorithm which does not constrain the smoothness of the surface to the same degree. To some extent, simplicity is a matter of scale, and so at a close range one might expect better results; but it is an extension that we do not consider in this work.

We conclude that the system provides a useful interface for reconstructing surfaces from urban scenes, and produces high quality and visually pleasing results.

References

[Andrews and Séquin(2014)] James Andrews and Carlo H Séquin. Type-constrained direct fitting of quadric surfaces. *Computer-Aided Design and Applications*, 11(1):107–119, 2014.

[Beale et al.(2016)Beale, Yang, Cosker, and Campbell] Daniel Beale, Yongliang Yang, Darren Cosker, and Neill Campbell. Fitting quadrics with a bayesian prior. *Computational Visual Media*, 2016.

[Bishop et al.(1998)Bishop, Svensén, and Williams] Christopher M Bishop, Markus Svensén, and Christopher KI Williams. Gtm: The generative topographic mapping. *Neural computation*, 10(1):215–234, 1998.

[Bo et al.(2012)Bo, Ling, and Wang] Pengbo Bo, Ruotian Ling, and Wenping Wang. A revisit to fitting parametric surfaces to point clouds. *Computers & Graphics*, 36(5):534–540, 2012.

	Point cloud	Poisson reconstruction	Primitives	Closure

The cross

The post office

Mantova Basilica

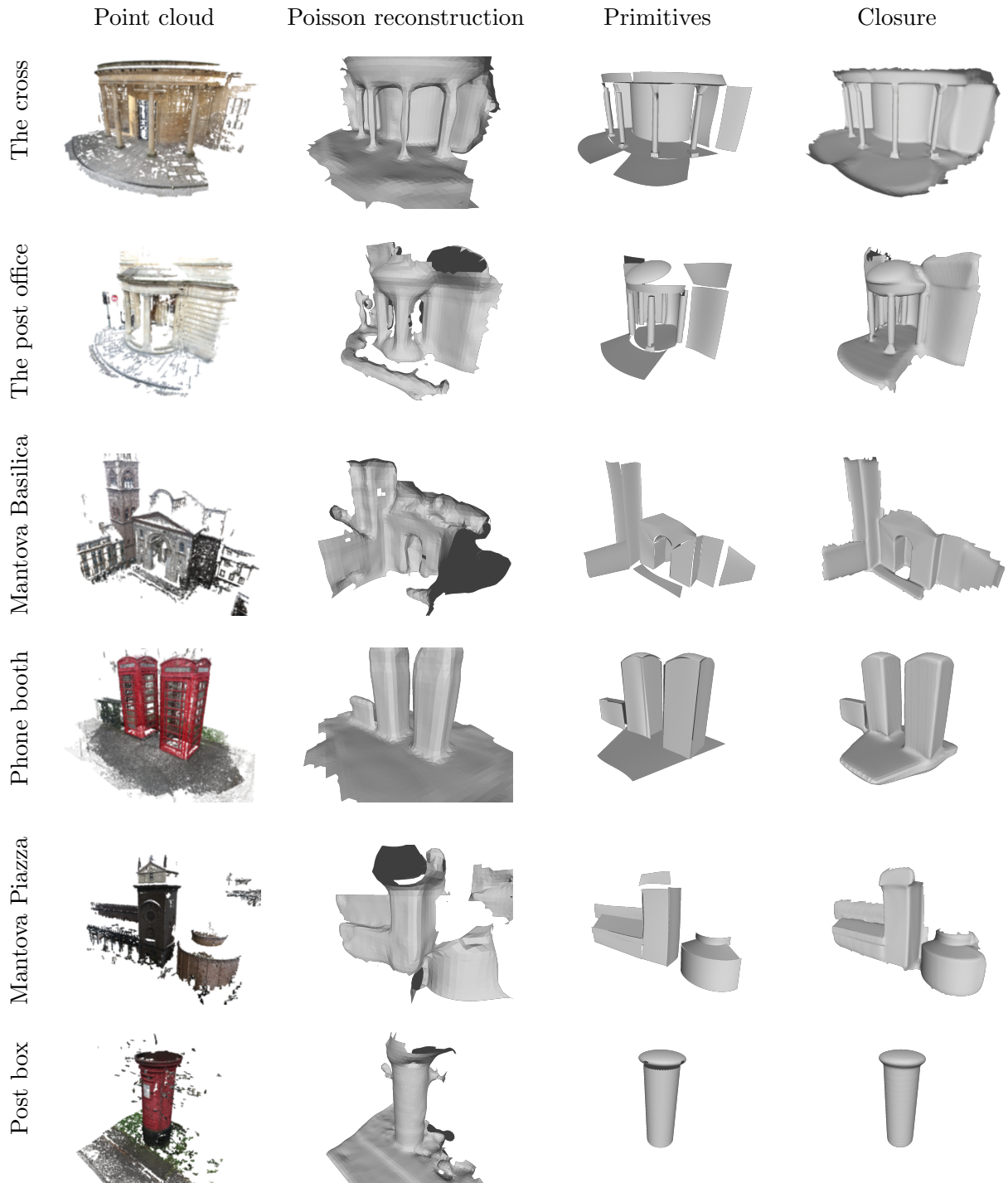Phone booth

Mantova Piazza

Post box

Figure 10: Some results from the method presented in this paper after fitting the primitives and applying surface closure. The far left shows the original point cloud, the second column shows the results of normal estimation and Poisson reconstruction, and the right hand side shows the extracted primitives, and the result of applying a Poisson reconstruction to the output.

[Bolitho et al.(2009)Bolitho, Kazhdan, Burns, and Hoppe] Matthew Bolitho, Michael Kazhdan, Ran-

Figure 11: The final textured results.

dal Burns, and Hugues Hoppe. Parallel poisson surface reconstruction. In *Advances in Visual Computing*, pages 678–689. Springer, 2009.

[Campbell et al.(2010)Campbell, Vogiatzis, Hernández, and Cipolla] Neill DF Campbell, George Vogiatzis, Carlos Hernández, and Roberto Cipolla. Automatic 3d object segmentation in multiple views using volumetric graph-cuts. *Image and Vision Computing*, 28(1):14–25, 2010.

[Cao et al.(2014)Cao, Ju, Fu, and Hu] Yan-Pei Cao, Tao Ju, Zhao Fu, and Shi-Min Hu. Interactive image-guided modeling of extruded shapes. In *Computer Graphics Forum*, volume 33, pages 101–110. Wiley Online Library, 2014.

[Changchang(2011a)] W. Changchang. Visualsfm: A visual structure from motion system. *http://ccwu.me/vsfm/*, 2011a.

[Changchang(2011b)] W. Changchang. Multicore bundle adjustment. *Computer Vision and Pattern Recognition (CVPR)*, 2011b.

[Chen et al.(2013)Chen, Zhu, Shamir, Hu, and Cohen-Or] Tao Chen, Zhe Zhu, Ariel Shamir, Shi-Min Hu, and Daniel Cohen-Or. 3-sweep: extracting editable objects from a single photo. *ACM Transactions on Graphics (TOG)*, 32(6):195, 2013.

[Chen et al.(2008)Chen, Kang, Xu, Dorsey, and Shum] Xuejin Chen, Sing Bing Kang, Ying-Qing Xu, Julie Dorsey, and Heung-Yeung Shum. Sketching reality: Realistic interpretation of architectural designs. *ACM Transactions on Graphics (TOG)*, 27(2):11, 2008.

[Cignoni et al.(2008)Cignoni, Corsini, and Ranzuglia] Paolo Cignoni, Massimiliano Corsini, and Guido Ranzuglia. Meshlab: an open-source 3d mesh processing system. *Ercim news*, 73(45-46):6, 2008.

[Debevec et al.(1996)Debevec, Taylor, and Malik] Paul E Debevec, Camillo J Taylor, and Jitendra Malik. 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*, pages 11–20. ACM, 1996.

[Dragiev et al.(2011)Dragiev, Toussaint, and Gienger] Stanimir Dragiev, Marc Toussaint, and Michael Gienger. Gaussian process implicit surfaces for shape estimation and grasping. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 2845–2850. IEEE, 2011.

[Furukawa(2010)] J. Furukawa, Y. Ponce. Accurate, dense and robust multi-view stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 32(8):1362–1378, 2010.

[Furukawa et al.(2009)Furukawa, Curless, Seitz, and Szeliski] Yasutaka Furukawa, Brian Curless, Steven M Seitz, and Richard Szeliski. Reconstructing building interiors from images. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 80–87. IEEE, 2009.

[Gopi et al.(2000)Gopi, Krishnan, and Silva] M Gopi, Shankar Krishnan, and Cláudio T Silva. Surface reconstruction based on lower dimensional localized delaunay triangulation. In *Computer Graphics Forum*, volume 19, pages 467–478. Wiley Online Library, 2000.

[Habbecke and Kobbelt(2012)] Martin Habbecke and Leif Kobbelt. Linear analysis of nonlinear constraints for interactive geometric modeling. In *Computer Graphics Forum*, volume 31, pages 641–650. Wiley Online Library, 2012.

[Hoppe et al.(1992)Hoppe, DeRose, Duchamp, McDonald, and Stuetzle] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. *Surface reconstruction from unorganized points*, volume 26. ACM, 1992.

[Iarussi et al.(2015)Iarussi, Bommes, and Bousseau] Emmanuel Iarussi, David Bommes, and Adrien Bousseau. Bendfields: Regularized curvature fields from rough concept sketches. *ACM Transactions on Graphics (TOG)*, 34(3):24, 2015.

[Jiang et al.(2009)Jiang, Tan, and Cheong] Nianjuan Jiang, Ping Tan, and Loong-Fah Cheong. Symmetric architecture modeling with a single image. *ACM Transactions on Graphics (TOG)*, 28(5): 113, 2009.

[Kazhdan and Hoppe(2013)] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (TOG)*, 32(3):29, 2013.

[Kazhdan et al.(2006)Kazhdan, Bolitho, and Hoppe] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. *Proceedings of the fourth Eurographics symposium on Geometry processing*, 2006.

[Kolmogorov(2004)] Vladimir Kolmogorov. *Graph based algorithms for scene reconstruction from two or more views.* PhD thesis, Citeseer, 2004.

[Krishnamurthy and Levoy(1996)] Venkat Krishnamurthy and Marc Levoy. Fitting smooth surfaces to dense polygon meshes. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 313–324. ACM, 1996.

[Labatut et al.(2009)Labatut, Pons, and Keriven] Patrick Labatut, J-P Pons, and Renaud Keriven. Hierarchical shape-based surface reconstruction for dense multi-view stereo. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 1598–1605. IEEE, 2009.

[Lau et al.(2010)Lau, Saul, Mitani, and Igarashi] Manfred Lau, Greg Saul, Jun Mitani, and Takeo Igarashi. Modeling-in-context: user design of complementary objects with a single photo. In *Proceedings of the Seventh Sketch-Based Interfaces and Modeling Symposium*, pages 17–24. Eurographics Association, 2010.

[Lawrence(2004)] Neil D Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. *Advances in neural information processing systems*, 16:329–336, 2004.

[Levin et al.(2004)Levin, Lischinski, and Weiss] Anat Levin, Dani Lischinski, and Yair Weiss. Colorization using optimization. *ACM Trans. Graph.*, 23(3):689–694, August 2004.

[Li et al.(2011)Li, Wu, Chrysathou, Sharf, Cohen-Or, and Mitra] Yangyan Li, Xiaokun Wu, Yiorgos Chrysathou, Andrei Sharf, Daniel Cohen-Or, and Niloy J Mitra. Globfit: Consistently fitting primitives by discovering global relations. In *ACM Transactions on Graphics (TOG)*, volume 30, page 52. ACM, 2011.

[Lipp et al.(2014)Lipp, Wonka, and Müller] Markus Lipp, Peter Wonka, and Pascal Müller. Push-pull++. *ACM Transactions on Graphics (TOG)*, 33(4):130, 2014.

[Mitra and Nguyen(2003)] Niloy J Mitra and An Nguyen. Estimating surface normals in noisy point cloud data. In *Proceedings of the nineteenth annual symposium on Computational geometry*, pages 322–328. ACM, 2003.

[Monszpart et al.(2015)Monszpart, Mellado, Brostow, and Mitra] Aron Monszpart, Nicolas Mellado, Gabriel J Brostow, and Niloy J Mitra. Rapter: rebuilding man-made scenes with regular arrangements of planes. In *ACM SIGGRAPH*, volume 2015, 2015.

[Musialski et al.(2013)Musialski, Wonka, Aliaga, Wimmer, Gool, and Purgathofer] Przemyslaw Musialski, Peter Wonka, Daniel G Aliaga, Michael Wimmer, L Gool, and Werner Purgathofer. A survey of urban reconstruction. In *Computer Graphics Forum*, volume 32, pages 146–177. Wiley Online Library, 2013.

[Nan et al.(2010)Nan, Sharf, Zhang, Cohen-Or, and Chen] Liangliang Nan, Andrei Sharf, Hao Zhang, Daniel Cohen-Or, and Baoquan Chen. Smartboxes for interactive urban reconstruction. *ACM Transactions on Graphics (TOG)*, 29(4):93, 2010.

[Portelli et al.(2010)Portelli, Ganovelli, Tarini, Cignoni, Dellepiane, and Scopigno] Davide Portelli, Fabio Ganovelli, Marco Tarini, Paolo Cignoni, Matteo Dellepiane, and Roberto Scopigno. A framework for user-assisted sketch-based fitting of geometric primitives. 2010.

[Schnabel et al.(2007)Schnabel, Wahl, and Klein] Ruwen Schnabel, Roland Wahl, and Reinhard Klein. Efficient ransac for point-cloud shape detection. In *Computer graphics forum*, volume 26, pages 214–226. Wiley Online Library, 2007.

[Sinha et al.(2008)Sinha, Steedly, Szeliski, Agrawala, and Pollefeys] Sudipta N Sinha, Drew Steedly, Richard Szeliski, Maneesh Agrawala, and Marc Pollefeys. Interactive 3d architectural modeling from unordered photo collections. In *ACM Transactions on Graphics (TOG)*, volume 27, page 159. ACM, 2008.

[Starck et al.(2006)Starck, Miller, and Hilton] Jonathan Starck, Gregor Miller, and Adrian Hilton. Volumetric stereo with silhouette and feature constraints. In *British machine vision conference*, pages 1189–1198, 2006.

[Subr et al.(2013)Subr, Paris, Soler, and Kautz] Kartic Subr, Sylvain Paris, Cyril Soler, and Jan Kautz. Accurate binary image selection from inaccurate user input. *Computer Graphics Forum*, 32(2pt1):41–50, 2013.

[van den Hengel et al.(2007)van den Hengel, Dick, Thormählen, Ward, and Torr] Anton van den Hengel, Anthony Dick, Thorsten Thormählen, Ben Ward, and Philip HS Torr. Videotrace: rapid interactive scene modelling from video. volume 26, page 86. ACM, 2007. ISBN 0730-0301.

[Wang et al.(2006)Wang, Pottmann, and Liu] Wenping Wang, Helmut Pottmann, and Yang Liu. Fitting b-spline curves to point clouds by curvature-based squared distance minimization. *ACM Transactions on Graphics (ToG)*, 25(2):214–238, 2006.

[Williams and Fitzgibbon(2007)] Oliver Williams and Andrew Fitzgibbon. Gaussian process implicit surfaces. *Gaussian Proc. in Practice*, 2007.

[Xu et al.(2014)Xu, Chang, Sheffer, Bousseau, Mccrae, and Singh] Baoxuan Xu, William Chang, Alla Sheffer, Adrien Bousseau, James Mccrae, and Karan Singh. True2form: 3d curve networks from 2d sketches via selective regularization. *ACM Transactions on Graphics*, 33(4), 2014.

[Yin et al.(2014)Yin, Huang, Zhang, Gong, Cohen-Or, and Chen] Kangxue Yin, Hui Huang, Hao Zhang, Minglun Gong, Daniel Cohen-Or, and Baoquan Chen. Morfit: Interactive surface reconstruction from incomplete point clouds with curve-driven topology and geometry control. *ACM Transactions on Graphics*, 33(6), 2014.