Block World Reconstruction from Spherical Stereo Image Pairs

Hansung Kim and Adrian Hilton

Centre for Vision, Speech and Signal Processing
University of Surrey, Guildford, Surrey, GU2 7XH, UK
h.kim@surrey.ac.uk

Abstract

We propose a block-based scene reconstruction method using multiple stereo pairs of spherical images. We assume that the urban scene consists of axis-aligned planar structures (Manhattan world). Captured spherical stereo images are converted into six central-point perspective images by cubic projection and façade alignment. Depth information is recovered by stereo matching between images. Semantic regions are segmented based on colour, edge and normal information. Independent 3D rectangular planes are constructed by fitting planes aligned with the principal axes of the segmented 3D points. Finally cuboid-based scene structure is recovered from multiple viewpoints by merging and refining planes based on connectivity and visibility. The reconstructed model efficiently shows the structure of the scene with a small amount of data.

Keywords:
3D reconstruction, Scene modelling, Spherical imaging, Block world interpretation

1. Introduction

3D scene reconstruction from photographic images has been an important research topic for various domains. Applications include visual sets in film and game production, 3D map generation, virtual tourism and urban planning. There have been many studies into outdoor scene reconstruction from multi-view images [1, 2, 3]. Strecha et al. created a benchmarking site for the quantitative evaluation of algorithms against ground-truth by LIDAR

scanning [4]. However, the quality of pure image-based reconstruction largely depends on the capture environment.

Firstly, real environments include complex appearance causing errors in reconstruction from images. Textureless and non-Lambertian surfaces often result in errors in matching and reconstruction. Scenes reflected on glass or water induce false depth. Moving pedestrians and cars in the scene can be occluders in urban scene modelling.

Another problem is that normal cameras with a limited field-of-view (FOV) capture only a partial observation of the surrounding environment. Reconstruction of a complete model of the 3D environment requires additional views to capture the scene and occluded regions. Reconstruction of scene models from multiple images or video acquired with a standard camera has been the focus of considerable research. However, the limited FOV presents a challenging problem to ensure complete scene coverage for reconstruction. Agarwal st al.[5] reconstructed full 3D street models from 150,000 photos from the internet using grid computing. Pollefeys et al.[6] used 3,000 video frames to reconstruct one building and 170,000 frames for a small town. The relatively narrow FOV and low resolution of normal cameras require acquisition and processing of large image sets for scene model reconstruction.

Finally, conventional dense reconstruction methods such as LIDAR scans or image-based reconstruction result in millions of points with a high-level redundancy which do not efficiently represent the scene structure. The task of extracting a structured representation for subsequent visualisation is typically performed manually. When we applied our previous dense reconstruction algorithm [7] for datasets covering areas of 30m diameter surrounded by buildings, it produced more than 100 million faces with 60 million vertices. This occupies huge amount of system memory and may require out of core techniques [8] to visualise and render. Applications such as 3D structure representation and pre-visualisation require scene models in a structured form for efficient storage, transmission and rendering.

Piecewise-planar, plane-based and block-based scene modelling methods provide a good solution for the above problems. These approaches start from the assumption that man-made environments such as urban areas or building interiors are composed of piecewise planar surfaces. Furukawa et al.[9] and Gupta et al.[10] used the strong assumption of a piecewise-axis-aligned-planar world (Manhattan world).

We previously presented a dense environment model reconstruction [7] and a plane-based reconstruction [11] using a line-scan camera and manual

segmentation. In this paper, we propose an automatic block-based environment model reconstruction method based on the same input data. This produces a more complete scene model with a compact representation for storage and transmission. The geometry can be refined for higher resolution mesh models if dense depth information is available. The approach provides a compact scene model for hierarchical geometry representation of the detailed scene structure.

The main contributions of this paper are:

- We propose a 3D block-based scene reconstruction system. This is a simple and efficient way to represent the structure of a scene with high completeness for transmission and interactive visualization.

- Spherical stereo imaging enables full scene reconstruction with a small number of input images. This saves considerable time in scene capture and reconstruction.

- We propose a façade alignment algorithm to find regions in the scene for optimal alignment and cubic projection. Cubic projection decompose the spherical image into six central-point perspective images. The central-point perspective image is advantageous in feature matching and 3D plane reconstruction because it is distortion-free and has a vanishing point at the centre of the image aligned with the principal axes for a Manhattan world.

- We propose an automatic extraction of plane and cuboid structure from colour and depth images. Optimal block-based representation of the scene is recovered based on visibility, occupancy, point density and physical stability.

- We provide an optional user interaction to constrain primitive reconstruction to keep specific geometrical details or refine erroneous regions.

- High resolution texture mapping from the original images to the block based representation gives a quick rendering of the scene.

The rest of this paper is organised as follows: Section 2 introduces related previous works and Section 3 outlines overview of the proposed method. Section 4 presents capture method and cubic projection with façade alignment.

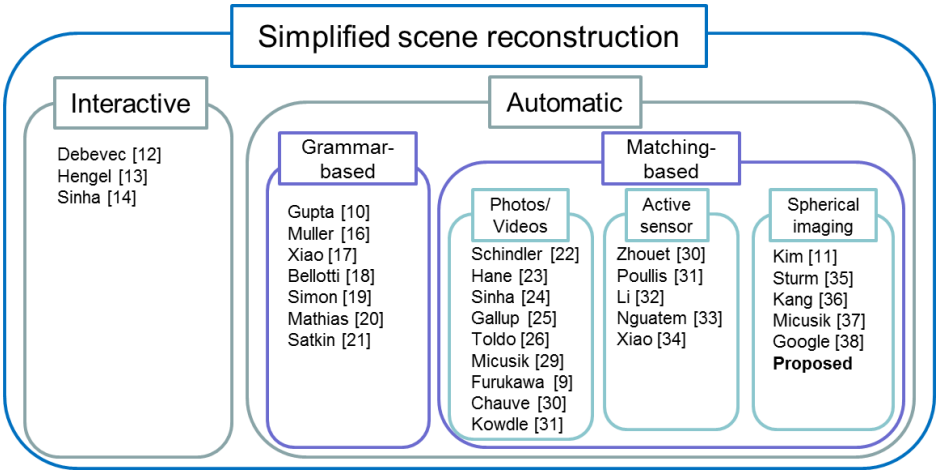


Figure 1: Categories of Simplified scene modelling methods

Depth reconstruction and region segmentation methods are proposed in Section 5. In Section 6, we introduce plane primitives reconstruction and structured block reconstruction methods. Experimental results and discussion are given in Section 7, and Section 8 makes conclusions of this work. Supplemental video is also available at: `http://www.cvssp.org/hkim/BlockWorld/BlockRecon-CVIU.mov` showing results of reconstruction for various scenes.

2. Related Work

Simplified scene modelling has been a long-standing area of research. Previous approaches can be separated into two categories: interactive and fully automatic methods. The automatic method are divided into grammar-based and matching-based approaches according to the registration strategy and the matching-based approach uses various input modalities as illustrated in Fig. 1

The FAÇADE system introduced by Debevec et al.[12] pioneered interactive environment modelling from images. In this approach, a simplified geometric model of the architecture is recovered interactively with manual correspondence using multiple view geometry. Novel views are rendered using view-dependent texture mapping, and additional geometric detail is recovered automatically through stereo correspondence. Their research was

commercialised as ImageModeler[1].

Hengel et al.[13] proposed an interactive 3D modelling method from video frames by tracing the shape of objects in the scene. They used structure from motion (SfM), feature point tracking and superpixel segmentation to get 3D information from 2D video frames. If users draw 2D primitives such as lines and circles on frames, then the system automatically builds 3D primitives from the user's input and reconstructed 3D information. This concept was extended by Sinha et al.[14] using feature-matching and SfM methods with line detection and vanishing point detection algorithms for interactive 3D architectural modelling from photo collections. SketchUp[2] provides a simple 3D reconstruction tool from multiple photos. This is similar to the Sinha's method but it does not use any matching method, just manual vanishing point alignment for photo registration to 3D coordinates. This tool is useful to build very simple scenes but has limitations in building complex scenes because it requires manual matchings for each primitive.

Automatic scene reconstruction can be divided into two categories: grammar-based and matching-based reconstruction. Grammar-based reconstruction uses semantic region detection and recognition to compose the world according to pre-defined rules. Gupta et al.[10] proposed block world reconstruction from a single outdoor image, inspired by the "Blocks World" work in the 1960's and Hoiem et al.'s "pop up 3D" [15]. They assume that the world is composed of blocks and match 2D image regions into 3D block view classes. They also estimate the density of each block using visual cues and use it to generate 3D parse graphs which describe geometric and mechanical relationships between objects within an image. Muller et al.[16] proposed a rule-base city modelling method using shape grammar rules from façade images, now commercialised as CityEngine[3]. Xiao et al.[17] proposed an automatic approach to generate street-side 3D photo-realistic models from images captured along streets at ground level with an assumption that building façades have two principal directions. They use a SfM method for the initial point cloud reconstruction and apply a multi-view semantic segmentation method for classifying regions into semantic models in the hand-labelled image database. Then, independent blocks are reconstructed using major

[1]ImageModeler, `http://usa.autodesk.com/adsk/servlet/pc/index?id=11390028&siteID=123112`

[2]SketchUp, `http://www.sketchup.com/`

[3]City Engine, `http://www.esri.com/software/cityengine/`

line structures and the final façade scene is modelled by inverse patch-based orthographic composition and structure analysis. This approach generates clean façade scenes but is highly computationally expensive, taking 23 hours on a cluster of 15 computers for semantic segmentation of 202 building blocks. Bellotti et al.[18] proposed an Architectonic Style Area (ASA) algorithm for procedural generation of buildings in an urban area, based on the concept of "architectonic likelihood". The algorithm accepts façade pictures from sample buildings and statistical description of the elements and styles as input, and composes façade models by statistically assembling sample images of architectonic components. Components are classified in an ontology based on the classic principles of architecture. The algorithm relies on rules that encode the semantics of the ontology. Simon et al.[19] proposed a grammar-based modelling method with basic shapes (roof, wall, window, balcony, floor, door, shop, etc.) and deviation tree for the procedural geometry. Mathias et al.[20] proposed a similar grammar-driven approach for reconstruction of buildings and landmarks, but they used an inverse procedural modelling strategy for SfM and image-based analysis. Satkin et. al [21] present a data-driven approach using repositories of 3D models to find the identities, poses and styles of objects in a scene. However, the grammar-based approach has a serious problem because semantic segmentation is not always stable, and this approach works only within the given rule and categories. Any object or building out of the given categories induces errors in reconstruction.

Multi-View Stereo (MVS) and SfM reconstruction is the most popular approach, not only in full geometry reconstruction, but also in piece-wise planar reconstruction. Schindler et al.[22] proposed a novel method for recovering the 3D-line structure of a scene from multiple widely separated views. 2D lines aligned to major axes are detected by EM-based vanishing point estimation. Those 2D lines are reconstructed as 3D lines to provide guide lines for 3D structure reconstruction. Hane et al. [23] proposed a piece-wise planar depth map fusion, which formulates an energy term in stereo matching using patch-based priors to reconstruct piece-wise planar scenes. Sinha et al.[24] suggested extracting vanishing directions and fitting point clouds into 3D planes reconstructed based on the vanishing directions. Gallup et al.[25] proposed a stereo method handling scenes containing both planar and non-planar regions by segmentation and planar region detection. The planar regions are represented by planes and the non-planar regions are modelled by the results of a standard multi-view stereo algorithm. One problem of this approach is the lack of completeness due to small independent planes.

Toldo et al.[26] proposed planar patch extraction based on photo consistency from point clouds using the J-linkage algorithm [27] and reconstructed the scene with a view clustering tree and hierarchical reconstruction. Some research has invoked the stronger Manhattan-world assumption [28] which states that the world is piecewise planar and aligned to orthogonal axes. Micusik et al.[29] proposed a super-pixel stereo on a Markov Random Field (MRF) and aligned surfaces to three dominant directions based on the gravity vector and vertical vanishing point. Furukawa et al.[9] also built indoor and outdoor scenes by axis aligned depth map integration relying on the Manhattan world assumption. The approach starts from point clouds generated by their Patch-based Multi-view Stereo (PMVS) algorithm [1] and finds an optimal minimum volume solution with plane hypotheses. Compensating or concealing the occlusion part of the scene is an important problem in 3D reconstruction. Chauve et al.[30] used additional ghost primitives to fill gaps between detected basic primitives by inducing cell complex. We use a similar plane extension technique in this paper to detect intersections of reconstructed partial planes. Kowdle et al.[31] proposed an active learning technique. They used an energy minimization framework for piecewise planar reconstruction but allowed simple user interaction to provide support for the uncertain regions.

Some approaches reconstruct geometry from point cloud datasets generated by an active sensor such as LIDAR without the help of image data. City modelling from aerial scans is one typical example. Zhouet al.[32] proposed a method to produce crack-free models composed of complex roofs and vertical walls from aerial LIDAR point clouds. Poullis et al.[33] also developed a fully automatic method for extracting high-fidelity geometric models directly from aerial LIDAR scans using 2D roof boundaries extraction based on GMM and camera pose estimation using Levenberg-Marquardt optimisation. Li et al.[34] introduced an idea for modelling algorithm from range data that exploits a priori knowledge that buildings can be modelled from cross-sectional contours using extrusion and tapering operations. Nguatem et al.[35] proposed an automatic cuboid fitting algorithm using a line sweep to reconstruct cuboid-based building model from point clouds. Xiao et al.[36] also developed a virtual walkthrough system with regularized texture-mapped 3D model using an inverse constructive solid geometry for large indoor scenes from ground-level photographs and 3D laser points.

The use of spherical imaging provides a simple approach to overcome the limited FOV of conventional cameras. Sturm[37] suggested a method for 3D

plane-based scene reconstruction from a single panoramic image. He used a priori constraints on the 3D structure such as: co-planarity of points, perpendicularity of planes and lines, and parallelism of planes and lines. Kang et al.[38] also proposed a similar 3D plane reconstruction method using the normal vector of plane and vanishing points from a single panoramic image. Point Grey developed an omnidirectional multi-camera system, the Ladybug[4], consisting of six XGA color CCDs to provide high resolution spherical images. Micusik et al.[39] used this camera for piecewise planar city modelling. They back-projected images to quadrangular planes and applied MRF superpixel stereo and depth sweeping algorithms for depth map reconstruction. The reconstructed depth maps were fused into surfaces aligned to three dominant directions. They assumed that the cameras are pre-calibrated and that reference images are also pre-segmented. Google also developed their own omnidirectional multi-camera system to reconstruct and render street models [40]. They simultaneously utilised range sensor to obtain a base-structure of the street scene and refined the model with optical flow estimation from captured images. In their approach, accurate registration of photometric and geometric information is important. Simultaneous sensing from different locations requires calibration and registration to align depth and image information.

Instead of omnidirectional or panoramic images, Feldman et al.[41] used the Cross Slits (X-Slits) projection with a rotating fisheye camera to generate a high quality spherical image and to reduce the dimension of the plenoptic function. In this research we use a similar line-scan camera to capture latitude-longitude image which has advantage in stereo matching and 3D reconstruction.

3. Overview of the Proposed System

In this research, we propose a simple and efficient method to reconstruct a simplified structured environment model from spherical image pairs. Figure 2 shows a block diagram for the whole process.

A linescan camera captures a full surrounding scene at multiple locations as vertical stereo pairs. The captured images are latitude-longitude images. They are projected into a unit cube with a novel façade alignment algorithm

[4]Pointgrey, `http://ww2.ptgrey.com/spherical-vision`

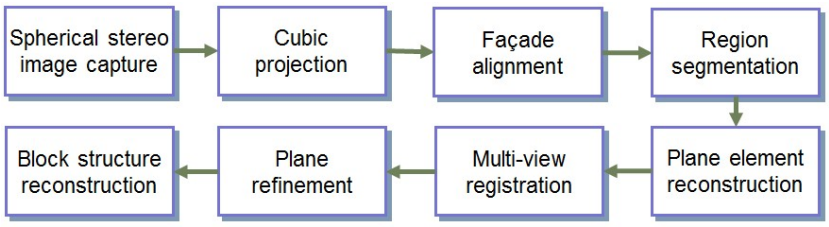


Figure 2: Block diagram of the system

based on the Hough transform. Each face image of the cubic projection is a distortion-free central-point perspective image whose three principal axes are aligned to vertical, horizontal, and the image centre directions, respectively.

To reconstruct depth information from stereo pairs, disparity estimation is performed. For automatic initial region segmentation, we propose a graph-based region segmentation extending Felzenszwalb and Huttenlocher's algorithm[42] to perform segmentation based on colour, normal direction and detected Hough lines. From the segmentation and disparity maps, independent 3D rectangular planes are constructed by plane fitting. The plane structure is refined by merging, expanding, cropping and eliminating planes validated against the reliability, visibility and occupancy.

Finally connected planes are extruded in the counter normal direction to construct block models. An optimal block structure is recovered based on the point density in each cuboid. The result represents the scene structure as a set of cuboids which can be used to render the scene with texture mapping.

4. Line-scan Capture and Cubic Projection

4.1. Spherical stereo acquisition

In this work, we use a commercial off-the-shelf line-scan camera[5] with a fisheye lens in order to capture the full environment as a high resolution spherical image. This camera samples rays on a hemisphere about the centre of projection and stitches together from the rotating slits together to form a new image. The camera rotates about axis through its optical centre. As a result the imaging geometry of the line-scan capture can be regarded as conventional perspective projection, and the result is a latitude-longitude image like a world map.

[5]Spheron, `https://www.spheron.com/products.html`

In order to recover depth information from the images, the scene is captured with the camera at two different heights. This vertical stereo line-scan camera capture has the following advantages:

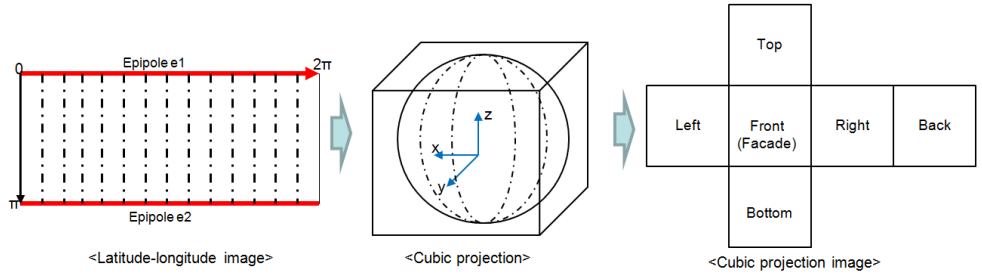
(1) Relatively simple calibration is required. Depth reconstruction only requires knowledge of the baseline distance between the stereo image pair and correction of radial distortion in the vertical direction. Radial distortion is rectified using a 1D lookup table to evenly map pixels on the vertical central line to the [0, π] range. Lens distortion parameters are fixed so that this mapping can be calculated for the lens in advance.

(2) Stereo matching can be simplified to a 1D search along the vertical scan line as discussed above, while normal spherical images require a complex search along conic curves or rectification of the images. In the latitude-longitude geometry, the great circles intersecting at the epipoles of the spherical geometry become parallel straight lines. Therefore, the conventional correlation-based matching on an 1D search range can be used to compute the disparity of spherical stereo images if they are vertically aligned. Error in the alignment can be corrected by rectification using the method proposed by Banno and Ikeuchi [43].

(3) High resolution images can be captured by a line-scan camera because the sensor array is 1D and the resolution about the axis depends on the step size. High resolution images provide more accurate depth estimation and high quality texture mapping.

4.2. Cubic projection and Façade alignment

The latitude-longitude images can be directly used for 3D reconstruction. However, we propose to convert the image into distortion-free perspective images via projection of the spherical image to a cube, referred to here as cubic projection. The cubic projection projects all pixels on the unit sphere to the unit cube in the range [-1, 1] in each axis. The converted image is decomposed into six perspective images as illustrated Fig. 3 (a) and (b). We set 0° of longitude as the x-axis in the cubic projection of Fig. 3 (b) and each side face image of the cubic projection has two vanishing points. These images have a single-vanishing point if we set the axes of the cubic projection to be aligned to the Manhattan world axes, which we refer to as façade alignment.



(a) Spherical to Cubic projection

(b) Projected image (Cathedral2)

(c) Facade aligned image (Cathedral2)

(d) Facade aligned projection image (Carpark1)

Figure 3: Cubic Projection

Façade alignment is the process of matching the main façades in the scene to be perpendicular to the principal axes of the cubic projection by rotating the spherical image around the vertical axis. Fig. 3 (c) shows the projection result when 126° of the longitude is set on the x-axis. Fig. 3 (d) is another example of the façade aligned cubic projection. We can observe that the horizontal and vertical lines in the scene were aligned to horizontal and vertical directions in each images, respectively, and that the lines aligned to the depth direction converges to the image centre. Therefore we can consider these images as formed by central-point perspective projection. Central-point perspective projection has significant advantage for axis-aligned plane reconstruction. Most of the current plane-based reconstruction algorithms

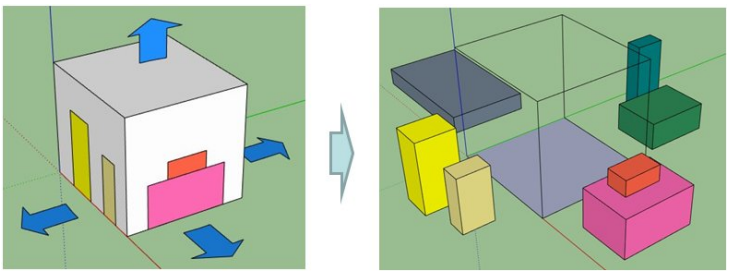


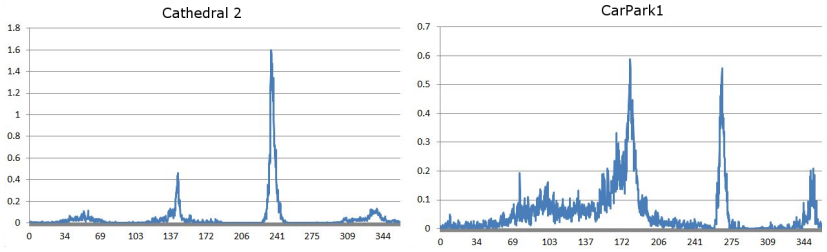
Figure 4: Block world reconstruction from façade-aligned cubic projection image

use vanishing point and principal directions detection in 3D space [9, 24, 29]. Cubic projection with the façade alignment can detect 3D principal directions in 2D images. Therefore, aligned 3D planes or 3D blocks can be built by extruding detected 2D planes in the depth direction as shown in Fig. 4.

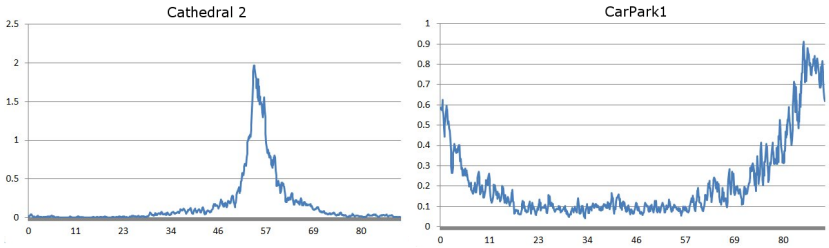
In order to find the most reliable angle shift t_{opt} to set the x-axis, we considers the number, sparseness, average length and average angle errors of image lines resulting from the probabilistic Hough transform [44]. The angle shift in longitude is equivalent to the horizontal pixel shift in the line-scan image. We detect the following three kinds of Hough lines as aligned among all detected lines H: Horizontal Hough lines H_h, Vertical Hough lines H_v and Perspective Hough lines (to the depth direction) H_p.

- $H_h = \{h_h | h_h \in H, |\theta(h_h)| < 1^\circ\}$
- $H_v = \{h_v | h_v \in H, |\theta(h_v) - 90^\circ| < 1^\circ\}$
- $H_p = \{h_p | h_p \in H, D(I_c, h_p) < r_c\}$

where θ is the angle of the line to the horizontal direction, and $D(I, L)$ is the distance between the image centre point I_c and the line L. The distance threshold r_c varies depending on the image resolution. In the above aligned Hough lines, H_h are most important to detect the façades of the scene and H_v are almost the same over any shift. We estimate the optimal rotational shift t_{opt} to find the façade direction of the scene by maximising the following energy term:



(a) Facade energy



(b) Facade energy (Sum)

Figure 5: Facade energy according to angle shift

$$t_{opt} = \underset{0° \leq t < 360°}{\text{argmax}} \; E_F(t) \tag{1}$$

$$E_F(t) = \lambda_R E_R(t) + \lambda_S E_S(t) + \lambda_L E_L(t) \tag{2}$$

$$E_R = \frac{|H_h \cup H_v \cup H_p|}{|H|}$$

$$E_S = \sigma_{H_h}$$

$$E_L = \frac{1}{|H_h|} \sum_{H_h} log(l(h_h)/\theta(h_h) + \varepsilon))$$

In Eq.(2), E_R represents the ratio of the number of aligned Hough lines to all Hough lines. E_S is the standard deviation of average y-position of H_h which relates to sparseness of the horizontal Hough lines. We give higher priority to sparse features in the scene because dense Hough lines can be detected from small areas which have complicate patterns and bias the optimisation. E_L represents the magnitude and accuracy of the detected H_h where $l(h)$ is the length of the line. The weighting factors λ_R, λ_S and λ_L can be adjusted according to the scene characteristics, but we fix λ_R=1.0,

λ_S=1.0, λ_L=0.3 throughout our experiments to show that those parameters are applicable to general scenes.

Figure 5 (a) shows $E_F(t)$ against all angle shifts from 0° to 360°. The Cathedral scene shows four distinctive peaks at around 90° intervals, while the CarPark scene has an ambiguous peak around 100° because the ground is slightly slanted and the high frequency texture of the tree and brick walls induce many outliers in the Hough transform. Using the assumption that façades in each side face are perpendicular to each other based on the Manhattan world assumption, we detect the optimal shift by maximising the following energy sum which results in Fig. 5 (b) with a more distinct peak point.

$$t_{opt} = \underset{0° \le t < 90°}{\operatorname{argmax}} \sum_{k=0}^{3} E_F(t + k * 90°) \tag{3}$$

The façade-aligned cubic projection image is a distortion-free central-point perspective image. It has several advantages over alternative projections. First, it is easy to extract axis-aligned planes from the image because it does not require any vanishing point detection. Second, it is easy to find matched features between multi-view images because they do not have distortion of the appearance while the spherical images have serious radial distortion according to the angle. Finally, multi-view registration is a simple 3 DOF problem (only translation) because the façades direction is already aligned for all views.

5. Depth Reconstruction and Region Segmentation

5.1. Depth reconstruction from spherical stereo

One of the most important problems in depth estimation is locating corresponding points in the images, a process referred to as disparity estimation. The estimated disparity fields can be converted into depth information by camera geometry. Depth reconstruction from images captured by conventional cameras require a calibration step to extract camera parameters. However, the spherical stereo pair and cubic projection pair used in this research do not require a complex calibration step because pixel positions in each image directly correspond to 3D spherical coordinates as described in Section 4.1.

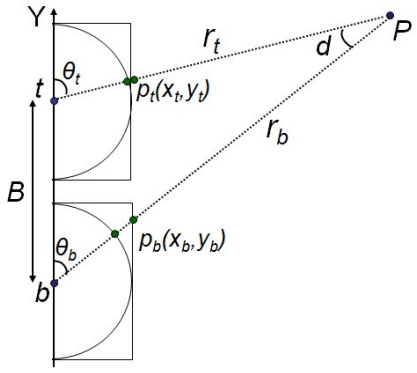
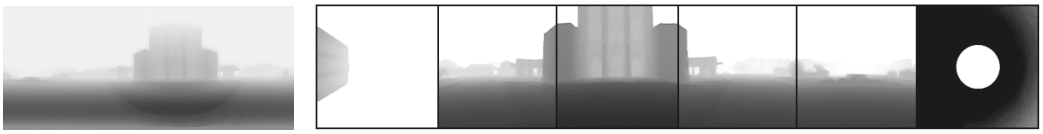


Figure 6: Spherical and cubic stereo geometry



(a) Disparity map (b) Depth map with cubic projection

Figure 7: Depth reconstruction result for Cath2 image set

The angle disparity d between two image pairs is defined as illustrated in Fig. 6. If we assume the angles of the projection of the point P onto the spherical or cubic projection image pair displaced along the y-axis are θ_t and θ_b respectively, then the angle disparity d of point $p_t(x_t, y_t)$ can be calculated as $d(p_t) = \theta_t - \theta_b$. The distance of the scene point P from the two cameras is calculated by triangulation as Eq. (4), where B is the baseline distance between the camera's center of projection and r_t and r_b represent the distance from P to the top and bottom cameras.

$$\begin{aligned} r_t &= B/\left(\frac{\sin\theta_t}{\tan(\theta_t + d)} - \cos\theta_t\right) \\ r_b &= B/\left(\cos\theta_b - \frac{\sin\theta_b}{\tan(\theta_b - d)}\right) \end{aligned} \tag{4}$$

Stereo matching can be carried our in either the spherical image pair or cubic projection image pairs. In the latitude-longitude image, the epipolar line for correspondence search is a vertical scan line. In cubic projection images, epipolar lines are vertical lines for side faces, and radial lines from

the centre for the top and bottom face images. Both image types have a trade-off according to the disparity estimation method. Disparity estimation on the latitude-longitude images is good for pixel-base approaches or global optimisation, but contains errors in area-based approaches like block matching because of the distortion of the image. On the other hand, cubic projection images show better results in area-based matching but correspondence should be independently estimated for each face image and requires boundary processing between face image. In any case, disparity estimation results can easily be converted between formats by the projection geometry in Fig. 6.

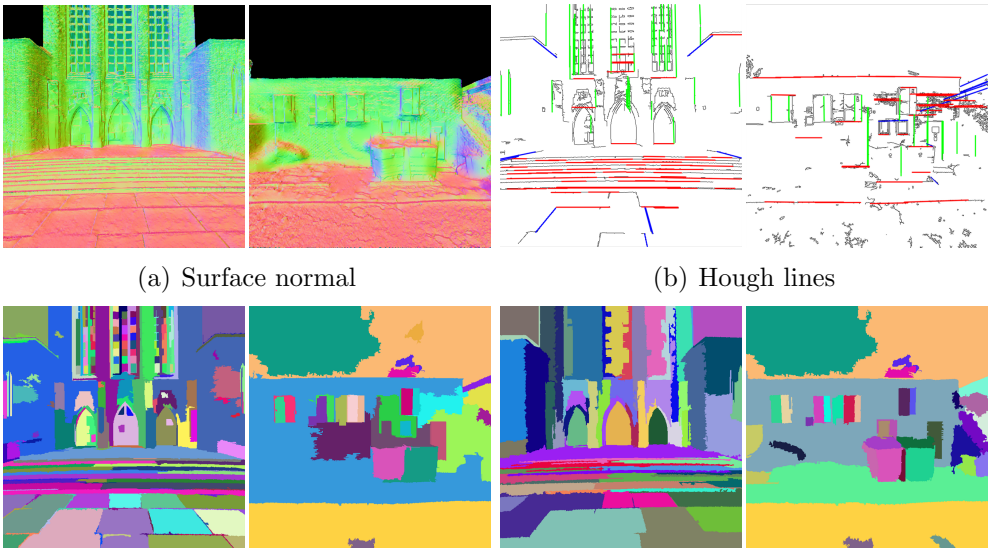
Any disparity estimation algorithm can be used for the proposed system as long as it does not produce too many outliers. We use latitude-longitude images and a PDE-based variational disparity estimation method previously proposed to generate accurate disparity fields with sharp depth discontinuities for surface reconstruction [7].

Figure 7 shows the result of the estimated angle disparity field and its depth map followed by cubic projection. Depth is mapped to grey scale according to their disparity or depth range.

5.2. Region segmentation for plane reconstruction

Felzenszwalb and Huttenlocher [42] proposed a simple and intuitive segmentation concept that: "The intensity differences across the boundary of two regions are perceptually important if they are large relative to the intensity difference inside at least one of the regions". We modify Felzenszwalb's segmentation method [42] to embrace 3D features such as surface normal direction and aligned Hough lines.

A graph $G = (V, E)$ is constructed for each face image domain, where $v_i \in V$ is the set of pixels and $(e_{ij}) \in E$ is the edge between neighbouring elements (v_i, v_j) with a weight $w(e_{ij})$. We set the affinity weights according to the colour difference W_c, face normal angle difference W_o and edge penalty W_e in Eq. (5).



(a) Surface normal (b) Hough lines

(c) Felzenszwalb (d) Proposed

Figure 8: Region segmentation results(Cathedral2 and Carpark1)

$$
\begin{aligned}
W &= W_c + \lambda_o W_o + W_e \qquad (5)\\
W_c &= dist(I(v_i) - I(v_j))\\
W_o &= |cos^{-1}(\overline{O(v_i)} \cdot \overline{O(v_j)})|\\
W_e &= \begin{cases} a & \text{if } v_i \text{ is on Aligned Hough lines,}\\ b & \text{else if } v_i \text{ is on Canny edge lines,}\\ 0 & \text{otherwise.} \end{cases}
\end{aligned}
$$

$I(v)$ is a colour value in (R, G, B) coordinates and $O(v)$ is a surface normal vector calculated from the depth map generated in section in 5.1. We experimentally set λ_o as 40 and edge penalties a and b as 400 and 200, respectively, in all of our experiments. We also set the region size preference parameter k in the Felzenszwalb's algorithm as 1200.

Figure 8 shows results of region segmentation for the main façade of the Cathedral and Carpark scenes. Figure 8 (a) shows the surface normal map projected into the (R, G, B) domain and Fig. 8 (b) shows Canny edge and Aligned Hough lines. Fig. 8(c) and (d) are results of Felzenszwalb's segmentation algorithm and the proposed algorithm respectively. We can

observe that the side walls and objects are clearly segmented by the proposed algorithm owing to the integration of surface normal and edge information.

However, the results are still over-segmented for plane reconstruction. Micusik et al.[29] used Felzenszwalb's segmentation for generating super-pixels and refined them iteratively using a 3D MRF. This method is computationally expensive to refine the super-pixel segmentation into meaningful coplanar regions and is unstable in many cases. Therefore we introduce a method to merge segmented regions in the plane reconstruction stage by considering their reliability and spatial relationship.

6. Block World Reconstruction

3D structured scene is reconstructed from the 2D images, region segments and disparity information. First, 3D rectangular plane elements are constructed by projecting segmented regions to 3D with the depth information. The resulting 3D planes are merged, eliminated and connected to generate the 3D plane structure of the scene. If multiple stereo reconstructions are available, they are registered and refined into one complete structure. Finally a block world model is generated from the plane structure by fitting cuboids.

We introduce a scene scale parameter S_c which represents the level of detail in the scene reconstruction. S_c defines the minimum size of objects to be reconstructed, and also to merge or eliminate less reliable planes. Small S_c can reconstruct details of the scene but large coherent area can be divided into small planes including erroneous pieces. Large S_c produces rough scene structure with less pieces but may lose scene details. We set S_c as 0.8m for outdoor scenes and smaller value of 0.2-0.4m for indoor scene according to the preference for scene details (S_c values for indoor scenes are given in Section 7). However, applying a single scale parameter to the whole scene can miss important details. We allow user interaction as an option to introduce hard constraint to specific regions to keep their properties in reconstruction.

6.1. Plane reconstruction

All 2D points $V_p \subset V$ in each segment can be projected into 3D space with the depth information to form a 3D point cloud. Rectangular planes are constructed from the segments and point clouds. Two different approaches can be used for plane detection in a point cloud segment: total least squares [45] and RANSAC-based [46]. The total least squares fitting is a form of linear regression and provides a solution to the problem of finding the best

Table 1: Plane classification

Class	Constraint
X plane	$n_x \sim N(0, \sigma^2_{n_x})$
Y plane	$n_y \sim N(0, \sigma^2_{n_y})$
Z plane	$n_z \sim N(0, \sigma^2_{n_z})$ & $t(z) \geq T_g$
Ground plane	$n_z \sim N(0, \sigma^2_{n_z})$ & $t(z) < T_g$
Arbitrary plane	otherwise (eliminated)

fitting 3D plane through a set of points, while the RANSAC-based plane detection iteratively selects a small subset of points at random to fit a model to that subset and remove outliers. The total least square method is fast and converges to a single solution, but the result can be biased by outliers. The RANSAC-based method can be more accurate if there are many outliers, but it is computationally expensive. In our case, the plane reconstruction is a large set of small problems, and the disparity estimation algorithm provides smooth and accurate depth fields over the surface except near region boundaries. Therefore, we exclude 10% of points close to region boundaries in the segmentation and apply the total least squares (orthogonal regression) fitting algorithm [45] and bounding box extraction. If the 3D point cloud is noisy, the RANSAC-based approach can be applied as an alternative.

Once all regions are fitted to planes, they are categorised into five classes (X, Y, Z, *Ground* and *Arbitrary*) according to the constraints with their normal vectors $\mathbf{n}$ and centre point $\mathbf{t}(x, y, z)$ as shown in Table 1. T_g, set as $S_c/2$, is a threshold to define *Ground* planes among Z-planes and $\sigma^2_{n_i}$ is the variation from the ideal normal vector for a particular plane orientation. We set $\sigma^2_{n_i}$ as 0.15 for the block world reconstructions that do not use arbitrary planes in our experiments. All reconstructed planes are saved as a vector list:

$$\mathbf{P} = \{p_i\} = \{[\mathbf{n}_i \, \mathbf{t}_i \, w_i \, h_i]\} \tag{6}$$

where w and h are the height and width of the plane.

Plane elements reconstructed from the region segments may include false planes and partial planes which can be merged into a larger plane. In order to refine those planes, we measure the following reliability factors for each plane: reconstruction confidence R_c, plane size R_s, distance from the camera R_d, distance between planes R_b and angle to the camera view direction R_θ.

$$R_c(p_i) = MSE(p_i) \tag{7}$$
$$R_s(p_i) = w_i \times h_i \tag{8}$$
$$R_d(p_i) = \|O_c - \mathbf{t}_i\| \tag{9}$$
$$R_b(p_i, p_j) = \|p_i - p_j\|_d \tag{10}$$
$$R_\theta(p_i) = |cos^{-1}(\overline{O_c\mathbf{t}_i} \cdot \mathbf{n}_i)| \tag{11}$$

$MSE(p_i)$ is the mean squared error calculated in the plane fitting, O_c is the location of the camera that the plane belongs to in the unified coordinate system and $\|\cdot\|_d$ is the minimum distance between two planes.

Merging Similar Planes: Two neighbouring planes p_i and p_j are merged into one plane and the bounding box is newly set to cover both regions if they satisfy the following conditions.

- Two planes are in the same category
- $R_b(p_i, p_j) < S_c$
- $\{R_s(p_i)_{new} < R_s(p_i)_{old}\} \cap \{R_s(p_j)_{new} < R_s(p_j)_{old}\}$

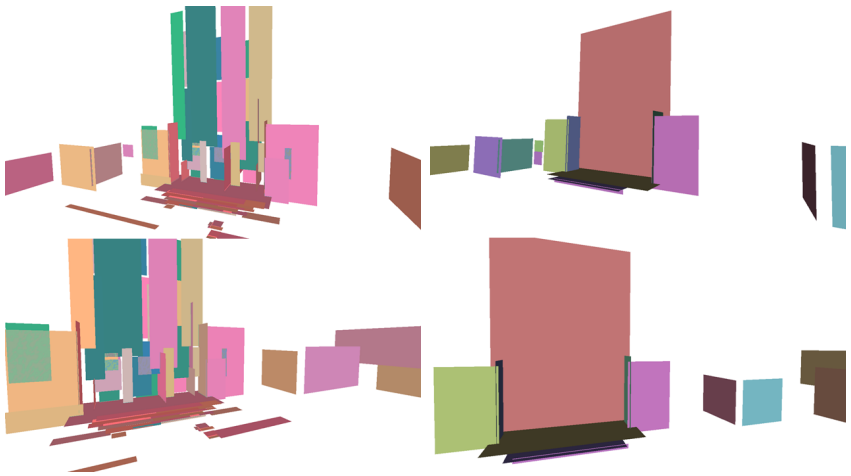
$R_s(p)_{old}$ is the original area of p and $R_s(p)_{new}$ is the new area to be extended to build the merged plane. The third condition keeps structures with a big hole such as bridge or door. The position of the new plane is set to the position of the old plane with lower $R_c(p)$.

Elimination of Unreliable Planes: According to the observations in [7], we assume that a plane is unreliable if it is too distant from the camera or its angle to the camera is too big. Therefore, the plane is eliminated if it satisfies the following conditions.

- $\{R_s(p_i) < S_c^2\} \cup \{R_d(p_i) > d_{max}\} \cup \{R_\theta(p_i) < \theta_{min}\}$

d_{max} is set to 20m for outdoor scenes and 5m for indoor scenes. θ_{min} is set to 15°. Figure 9 shows the original planes reconstructed from the segments and their refinement results by merging similar planes and eliminating unreliable ones.

Plane Intersection Refinement: All plane-to-plane intersections are checked if they have any intersection with each other in the extension range of $S_c/2$. If

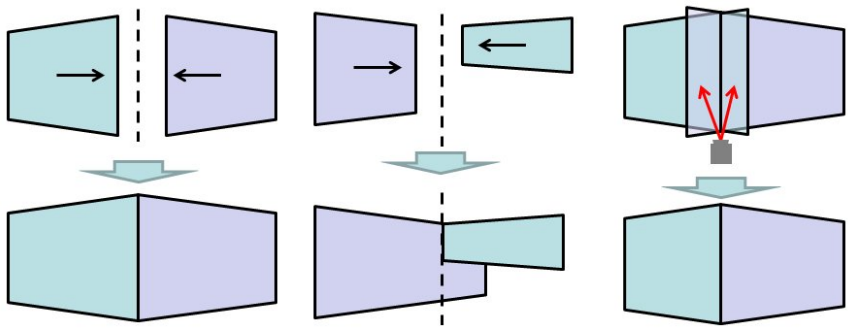


(a) Before plane refinement (b) After plane refinement

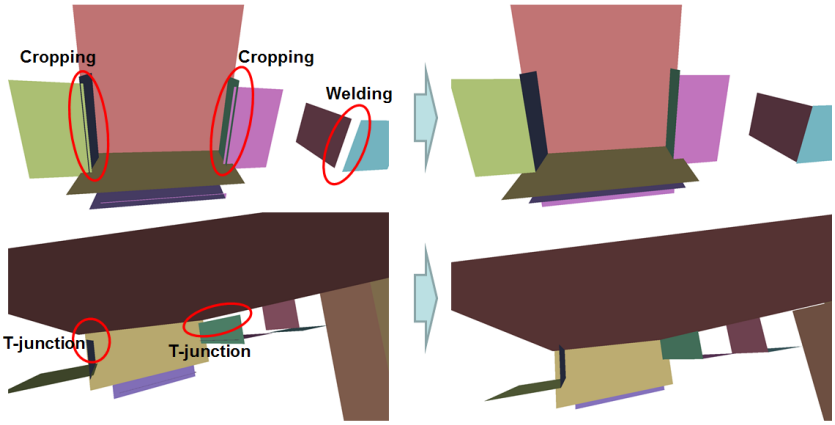
Figure 9: Plane refinement (Cathedral2, Randomly coloured planes)

any intersection is found, the length of intersection and visibility are checked to determine the type of intersection. If the intersection is larger than half of the bigger plane, two planes are welded at the intersection to generate a corner (Fig. 10 (a)). Otherwise, only the smaller plane stops growing at the intersection to generate a T-junction (Fig. 10 (b)). If two planes already have an intersection, residual parts are eliminated based on visibility constraints [47] (Fig. 10 (c)). If the plane does not meet any intersection during the extension in any direction, we keep the original boundary. Figure 10 (d) illustrates examples of the plane intersection refinement observed in the cathedral dataset.

Filling gaps from self-occlusion: The scene captured from a single fixed location inevitably has self-occlusions in the scene as illustrated in Fig. 11 (a). We adopt the minimum volume solution proposed by Furukawa et al.[48] based on the Manhattan World assumption. The plane occluded at the rear of the front plane is extended to the boundary of the orthogonal line-of-sight (LOS) from the surface normal direction as shown in Fig. 11 (b). The occluded region perpendicular to the orthogonal LOS is compensated by other viewpoints or the block reconstruction presented in the following subsections as demonstrated in Fig. 11 (c).



(a) Welding (b) T-junction (c) Cropping

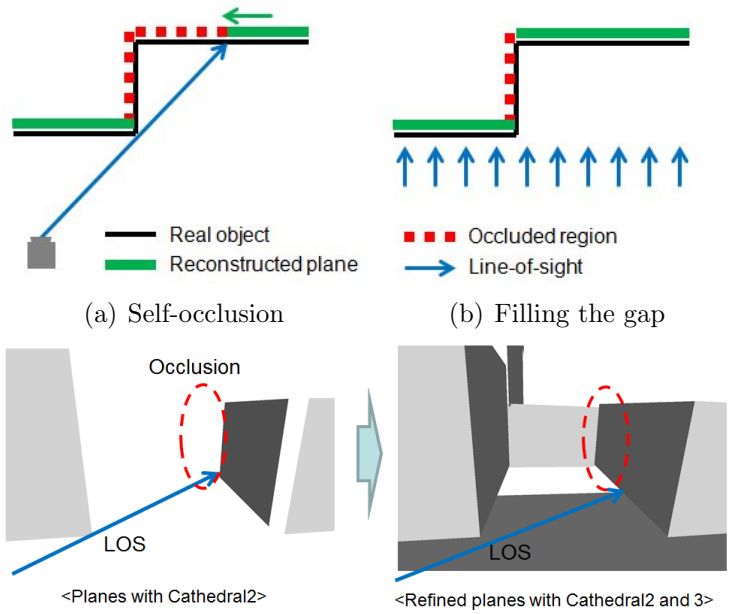


(d) Examples (Cathedral2, Randomly coloured planes)

Figure 10: Intersection refinement

6.2. Multiple-view plane structure reconstruction and texture mapping

The spherical camera is more advantageous in environment capture than normal camera with a limited FOV as mentioned in Section 1. However, spherical imaging does not capture the complete scene due to self-occlusion. Simple cases can be compensated by the minimum volume solution, but there is no way to get information for occluded regions behind any object from a single viewpoint. Another problem of the single-view spherical capture is the fact that the accuracy of the depth estimation is inversely-proportional to the distance and the angle of surface normal. These problems can be overcome by captures from multiple viewpoints. Merging reconstructions from multiple stereo pairs can be integrated into a common 3D scene structure. Kim and Hilton[7] proposed a mesh-fusion algorithm for dense meshes by mesh registration and reliable surface selection by considering surface visi-



(a) Self-occlusion (b) Filling the gap

(c) Example for the Cathedral scene

Figure 11: Occlusion filling

bility, orientation and distance. They calculate 3D rigid transforms between viewpoints using 2D feature matching.

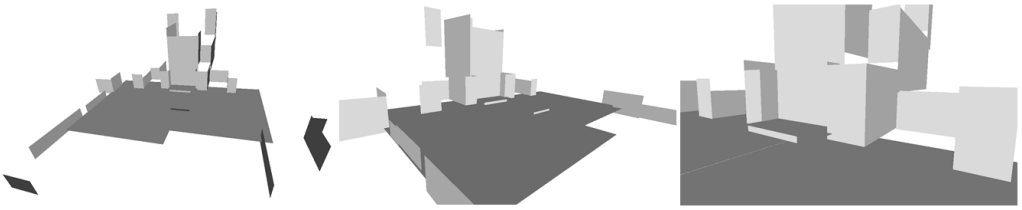
We propose a similar but much simpler and faster method based on the observations that: 1) the mesh registration is not optimised for rotation and translation (r, s), but only for translation s because the façade direction is already aligned for all viewpoints; 2) the surface reliability test is not applied for each vertex on the surface, but for the whole plane.

We use SURF feature matching [49] between captured images for different stereo pairs. The resulting 2D matches are projected into 3D space with the estimated depth field. However, these points are not reliable enough to be used in registration because of SURF matching and depth estimation errors. We use a RANSAC-based least square minimisation for the following error, where i and j are corresponding matching points in the model set m and reference set M, respectively.

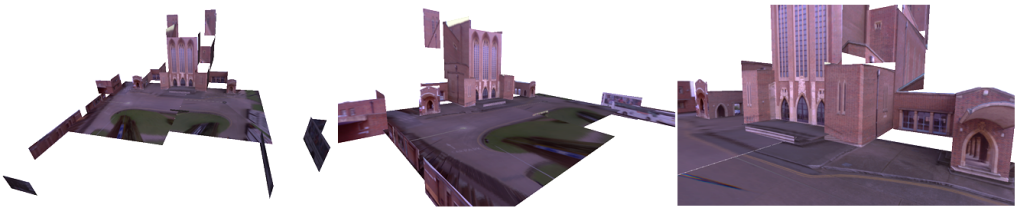
$$E_t(s) = \sum_{(i,j)} \|m_i - (M_j + s)\|^2 \tag{12}$$

Once all viewpoints are registered into a unified coordinate system, plane

(a) Multiple capture



(b) Reconstructed planes



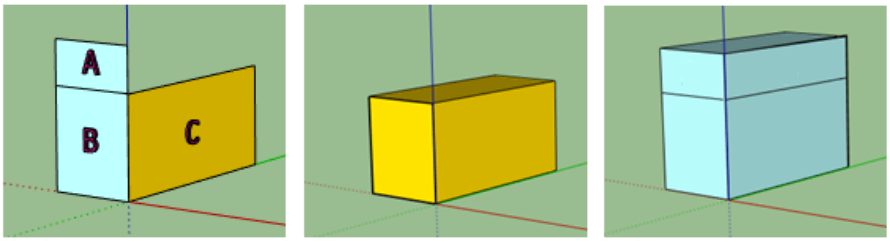
(c) Texture mapping

Figure 12: Multiple-view plane reconstruction

primitives are reconstructed independently for each viewpoint. All planes in the X, Y, and Z classes are then refined by the same method as the single view reconstruction.

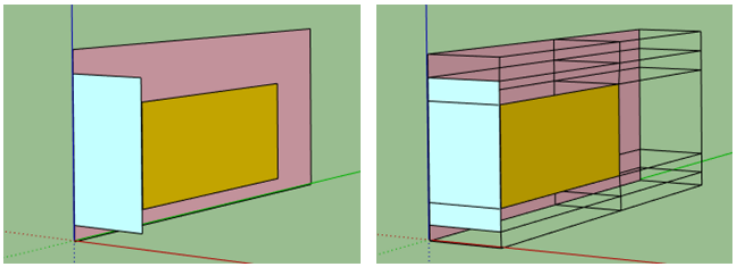
UV mapping [50] is used for texture representation. If texture mapping is required, all planes are subdivided into small regular triangles with their corresponding vertices in the texture image so that the mapped texture is not distorted. If the plane is merged from multiple viewpoints, the dominant viewpoint is decided by comparing their camera view direction R_θ in Eq. (11) and the texture is obtained from the dominant view image. Multiple blending is not used because the blending result can be blurred or result in ghosting artefacts due to the simplified geometry.

Figure 12 shows the result of plane reconstruction from multiple pairs of spherical stereo. The Cathedral scene was captured at three different locations and each reconstructions is merged into the central viewpoint. All major objects in the scene are reconstructed.



(a) Primitives (b) Min volume (c) Max volume

Figure 13: Cuboid reconstruction from two planes



(a) Primitives (b) Possible volumes

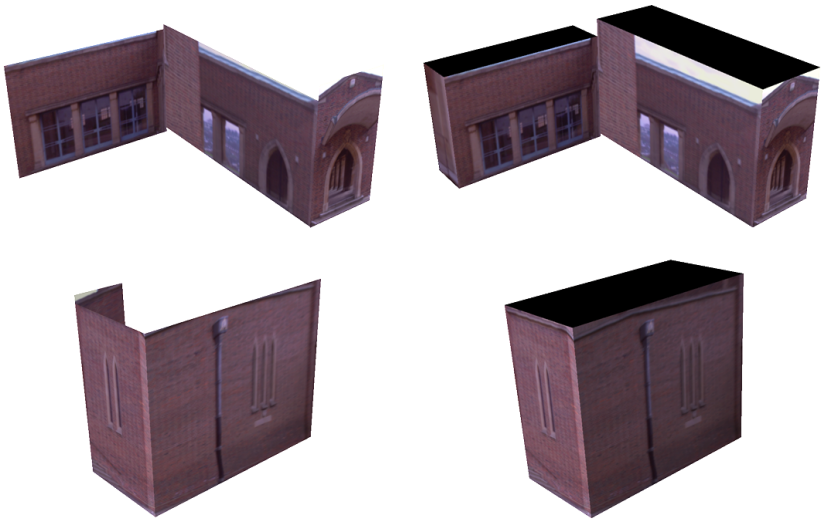
Figure 14: Cuboid reconstruction from three planes

6.3. Block-world reconstruction

Plane-based reconstruction describes simplified scene structure. However, block-based visualisation can provide better perception of the scene with surface normal orientations motivated by Gupta et al.[10]. This provides a model with higher completeness and an efficient representation of the scene because each block has only six degrees of freedom (3D location and dimensions).

Here we propose a cuboid fitting method starting from plane primitives reconstructed in Section 6.2. As mentioned in Section 4.2, cuboid reconstruction can be considered as an outward extrusion process (counter surface normal direction) of each cubic projection face.

If a plane primitive is connected to other perpendicular plane primitives whose extrusion directions overlap and they have different boundary lengths in the weld junction, the volume of the cuboid is decided by the original 3D point density in the primitive regions. We define an discrete objective function $D_o(P)$ as the density of 3D points belonging to the region as in Eq. (13).



(a) Plane primitives (b) Reconstructed cuboids

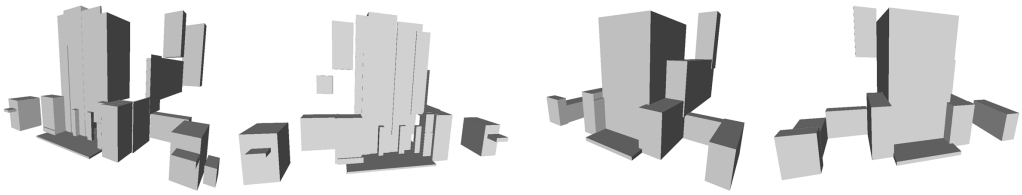
Figure 15: Examples of cuboid reconstruction (Cathedral, partial view)

$$D_o(P) = \frac{Number\ of\ 3D\ points\ in P}{Area\ of\ region\ P} \tag{13}$$

We start from the minimum volume for connected planes and check all possible volumes made up from planes upto the maximum volume. The objective function $D_o(P)$ is calculated for each volume and the volume with the maximum value is the optimal cuboid.

Let us consider a simple example as illustrated in Fig. 13. From the two plane primitives, we can consider two cases of cuboid reconstruction: Minimum volume in Fig. 13 (b) based on the regions B and C, and Maximum volume in Fig. 13 (b) based on the regions A, B and C. We compare $D_o(B \cup C)$ and $D_o(A \cup B \cup C)$ to choose the volume with the higher density. In the more complex case with three planes in Fig. 14, there are 18 different cases between minimum and maximum volumes. We calculate $D_o(P)$ for all possible volumes in the same way, and choose the case with the highest $D_o(P)$.

If the planes are isolated, they are extruded to an initial depth d_{init}. In case of multi-view reconstruction, planes are extruded in the counter normal direction. In the extrusion process, cuboids can intersect each other. If any intersection is detected, the original plane primitives and larger objects take



(a) S_C=0.4 (b) S_C=0.8

Figure 16: Cathdral main building reconstructed with different scene scale parameters

priority over limiting the extrusion of the smaller object.

Finally the block structures are refined based on their physical stability [10]. There may be floating blocks which do not meet the ground plane due to occlusion or disparity errors in the automatic reconstruction. These blocks are physically unstable, violating the law of gravity. If any block is not supported by other stable blocks and is close to the ground plane, the block is extended to the ground to retain the physical stability.

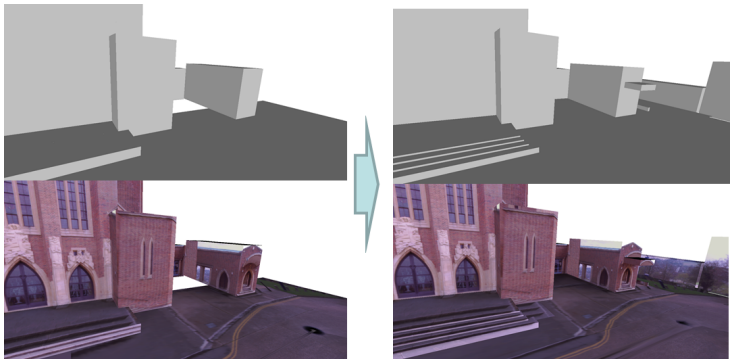
Figure 15 shows examples of the cuboid reconstruction from plane primitives in the Cathedral scene.

6.4. Optional user interaction to constrain primitive reconstruction

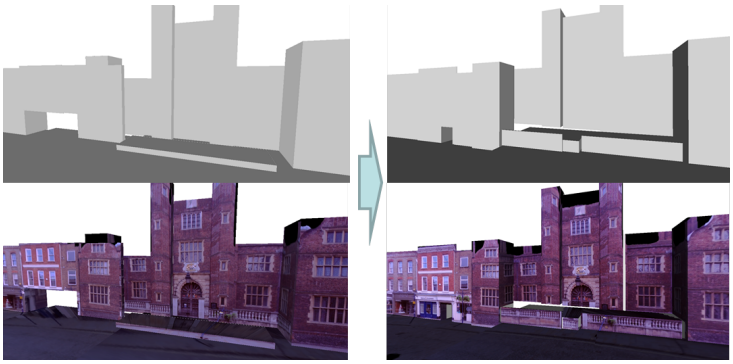
The proposed pipeline is a fully automatic method from the image input to the block world reconstruction. However, there are two possible problems in applying this automatic pipeline to various environments. First, automatic plane primitives reconstruction can fail to build meaningful coplanar regions due to the errors in disparity estimation or region segmentation. Second, applying a single scene scale parameter S_C can miss important details in the scene geometry.

Geometrical details can be preserved by adjusting the scene scale parameter S_C in reconstruction, but this may result in an over segmented scene with cluttered geometry. Figure 16 shows the Cathedral main building reconstructed for different scale parameters S_C. The smaller S_C in Fig. 16 (a) includes geometric details such as the sculptures and window regions in the main façade, and the eaves of the side wings. However, it can not fully resolve the steps and there are holes in the façade.

In order to overcome these problems, we implemented a simple user interface as an option to constrain the segmentation step. The user can merge or split regions by scribbling and assign X,Y and Z-plane class to specific regions as a hard constraint so that these regions are not affected by the



(a) Cathedral



(b) Highstreet

Figure 17: Hard-constrained primitive reconstruction introduced by user interaction. In both the Cathedral and Highstreet scene user-interaction required < 1 minute to introduce constraints on the reconstruction (Left: Fully-automatic, Right: With hard constraints)

region refinement step. It takes less than a minute for each image to define semantic constraints which are kept as independent clusters in the scene reconstruction.

Figure 17 illustrates results of introducing hard constraints on primitive reconstruction with user interaction. We observe that the eaves and steps are reconstructed regardless of the large scene scale parameter in the Cathedral scene. The cluttered background is also simplified by adding constraints. In the Hightstreet scene, the cluttered fence region and erroneous shop window regions are reconstructed by similar user interaction.

The inclusion of simple user-interaction to constrain the reconstruction is left as an option according to the application requirements for full or semi-automatic scene modelling. All experimental results in the following sections

are produced by the fully automatic process without user interaction.

7. Experimental Results

All scenes presented in this section were captured with a Spheron commercial line scan camera introduced in Section 4.1. We attached a Nikon 16mm f/2.8 AF fisheye lens to the system and captured vertical stereo pairs with a baseline of 60cm for outdoor scenes and 20cm for indoor scenes, respectively. The resolution of spherical images is 3143×1414.

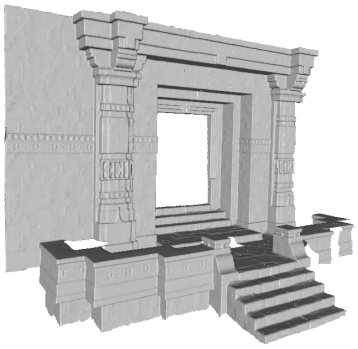
7.1. Evaluation against LIDAR ground-truth

The goal of our proposed approach is to reconstruct an approximate representation of the scene structures. Evaluation of geometric accuracy against ground-truth scene geometry therefore only provides a partial measure. In this section, we evaluated reconstruction results from test scenes against ground-truth models from LIDAR scans to show how close the proposed approach can represent the scenes. We compared accuracy and completeness of representation with a dense reconstruction method represented in Kim and Hilton[7].

Figure 18 shows the ground-truth from multiple LIDAR scans and the reconstructed models from three viewpoints using the proposed algorithm. The "Gate" scene has a width of 9m and a height of 6m. Stereo pairs are captured with a baseline of 60cm and the scene scale parameter S_c is set to 0.2m. The reconstructed plane primitives represent the approximate structure of the scene. Figure 19 also shows the ground-truth model from seven LIDAR scans and reconstructions of the main building for the "Cathedral" outdoor scene in Fig. 12. The main building has a width of 30m and a height of 20m.

Accuracy (how close the reconstruction is to the ground-truth) and completeness (how much of the ground-truth is modelled by the reconstruction) are measured based on the evaluation methodology proposed in Seitz et al.[51]. Both ground-truth and reconstructions are incomplete, therefore we considered only subset regions of the target model in measuring error distance. The reconstruction and ground-truth are registered in the same coordinate frame . Then Hausdorff distance from each vertex in the source model to the closest point in the target model is calculated. Accuracy is measured by the RMS error from the reconstruction to the ground-truth, and completeness is measure by the ratio of vertices in the ground-truth

(a) Captured images



(b) LIDAR ground-truth

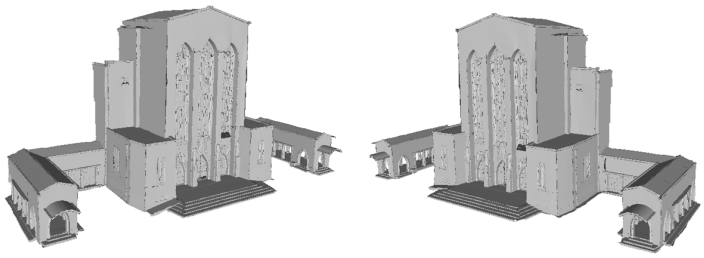
(c) Plane reconstruction

(d) Block reconstruction

Figure 18: Gate scene

whose closest points to the reconstruction exist within an allowable distance d_c. The plane and block reconstructions have vertices only at the corners of each plane. In order to measure the accuracy and completeness, all planes are regularly sampled on a 2cm×2cm grid. The block reconstruction includes redundant planes to complete cuboid structure. Therefore we use plane reconstruction results for the accuracy test and block reconstruction results for the completeness test.

Overlapped models and accuracy maps of the plane primitives against the ground-truth are illustrated in Fig. 20. Table 2 shows comparison of accuracy and completeness with dense reconstruction results from the same data sets using Kim and Hilton[7]. In measuring completeness, we set the allowable distance d_c as 0.25m for the Gate scene and 1.0m for the Cathedral scene



(a) LIDAR ground-truth

(b) Plane reconstruction

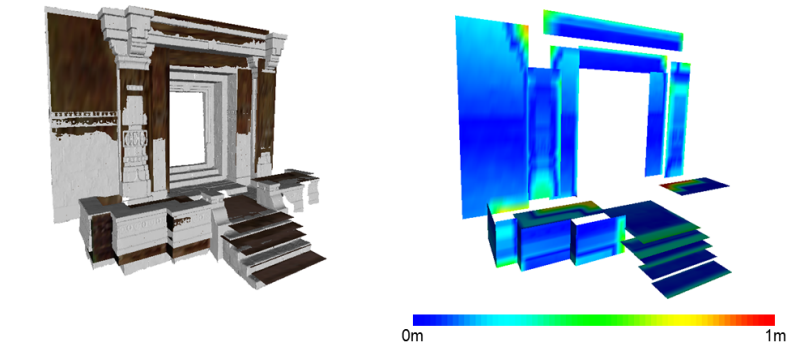
(c) Block reconstruction

Figure 19: Cathedral scene (main building only)

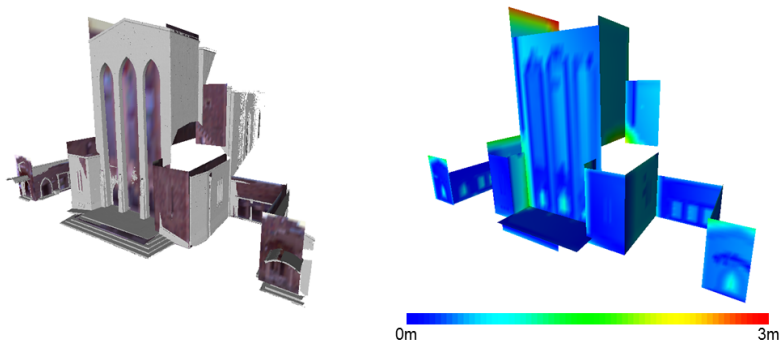
considering the scene scales and the range of errors. The dense reconstruction shows better results in the accuracy test, but the proposed block reconstruction also shows competitive accuracy especially with the Gate scene whose scene scale factor is set small. The proposed method also shows higher completeness with the Cathedral scene because occluded regions are covered in block reconstruction. Although the proposed algorithm cannot reconstruct geometric details in the scene, the reconstructed plane primitives are reliable and provide an efficient approximation of the scene structure.

7.2. Scene reconstruction results

We evaluated the proposed algorithm on four outdoor and one indoor scenes. The capture points and spherical images are shown in the first and second columns of Fig. 21. The Cathedral scene has a complex structure with many self-occlusions, there is sufficient overlap between views to reconstruct the complete cathedral façade. The Carpark scene was captured in a relatively small but complex area of 20m×25m including occlusions by



(a) Gate



(b) Cathedral

Figure 20: Accuracy maps of plane reconstruction results (Left: Overlaid reconstruction with ground-truth, Right: Accuracy map)

cars. There are relatively few overlapping regions between view 1 and 3. The Highstreet scene covers a street of 80m with four image pairs and includes many small and non-planar objects such as benches and trees. The Plaza scene covers a large and relatively complex area of 60m×80m with five image pairs. The Reception is an indoor scene captured in three locations. It covers an area of 20m×7m and the main area is connected to other corridors and rooms. The scene scale parameter S_c for the reception scene is set as 0.4m.

In the captured spherical images, most horizontal straight lines are distorted and it is hard to understand the structure of the scenes from the images. The columns 3-5 in Fig. 21 show automatically façade aligned cubic projection images for the spherical captures. We can observe that all horizontal and vertical lines in the scenes are aligned to x and y axes in the image planes and the vanishing point is located at the centre of each image.

Table 2: Accuracy and completeness evaluation of the dense reconstruction [7] and proposed reconstruction methods against ground-truth

Dataset	Accuracy (RMSE)		Completeness (%)	
	Gate	Cathedral	Gate	Cathedral
Dense	0.11	0.32	90.18	74.43
Proposed	0.17	0.57	82.28	88.37

Following automatic façade alignment and cubic projection the direction of the principal axis for Manhattan world plane reconstruction is easily found from the image axis.

Figure 22 and 23 show the reconstructed block-based structure of the scenes and their texture mapped results. The Cathedral set in Fig. 22 (a) clearly shows main structure of the scene though details such as narrow steps and awnings in the main building are not reconstructed. The Carpark scene in Fig. 22 (b) is more complicated but is efficiently represented with cuboids. This shows some errors in structure around cars and the wrong texture in some regions because of occlusions between objects and walls. The Highstreet scene in Fig. 22 (c) consists of a long street with many windows on buildings. Some buildings or parts of buildings are missing because they have large windows where reconstructed depth information is unreliable due to the reflection and transparency of the windows. Small windows can be reconstructed with the proposed algorithm because the plane location for the surrounding region is estimated in the refinement process. The Plaza scene in Fig. 23 (a) also includes large reflective regions and produces a few erroneous planes dominated by the scenes reflected on the glass, but they are removed in the refinement process. In the Plaza scene, viewpoints 3 and 4 do not have sufficient overlap in the images. Manual feature matching is performed for multiple view registration. The Reception scene in Fig. 23 (b) demonstrates how the proposed system works for an indoor environment. Textureless walls in the scene may cause serious distortion and errors in dense reconstruction. However, the majority of the walls are reconstructed in correct positions by the proposed system. Free-viewpoint video rendering of the scenes is available from: `http://www.cvssp.org/hkim/BlockWorld/BlockRecon-CVIU.mov`.

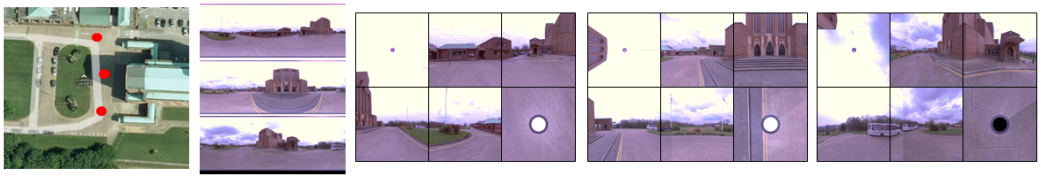
From the examples above, we can see that the proposed method generates a coarse approximation of the scene structure. Texture mapping produces natural rendering results.

Table 3 shows an analysis of runtime for the Cathedral dataset. We assume that we already have scanned spherical stereo image pairs and their

Table 3: Running time analysis (Cathedral dataset)

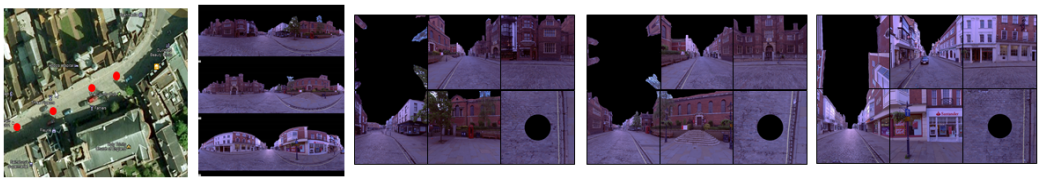
Step	Time (sec)
Data loading	0.29
Façade alignment	19.56
Region segmentation	3.91
Plane primitives reconstruction	4.12
Multi-view plane refinement	6.51
Cuboid reconstruction	2.58
Total	36.97

disparity maps. The proposed algorithms were run on a Intel Core i7 3.40GHz Windows machine with 32GB RAM. In the table, steps from data loading to plane primitives reconstruction were performed for individual pairs in parallel. It takes approximately 40 seconds per datasets.

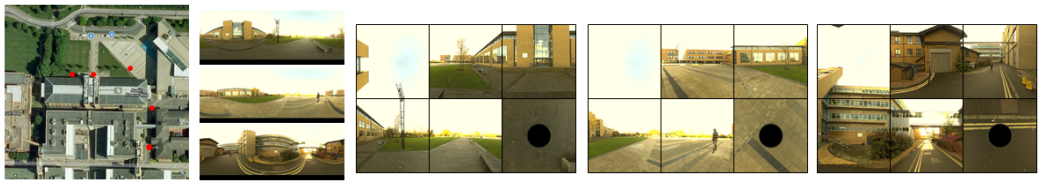


(a) Cathedral

(b) Carpark



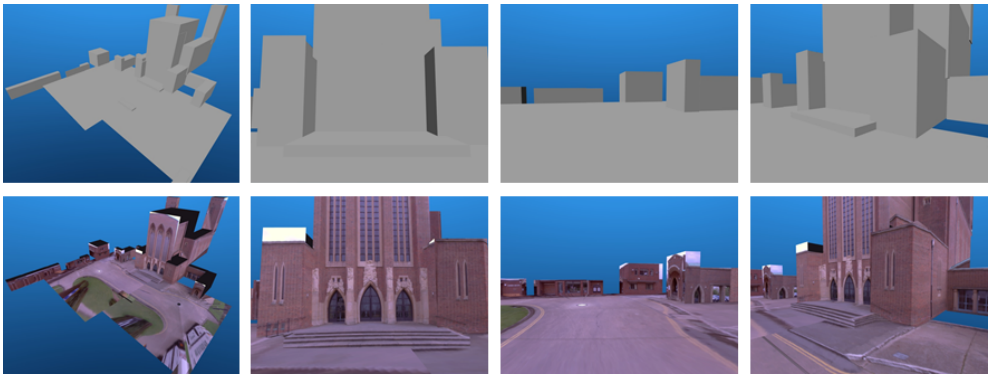
(c) Highstreet



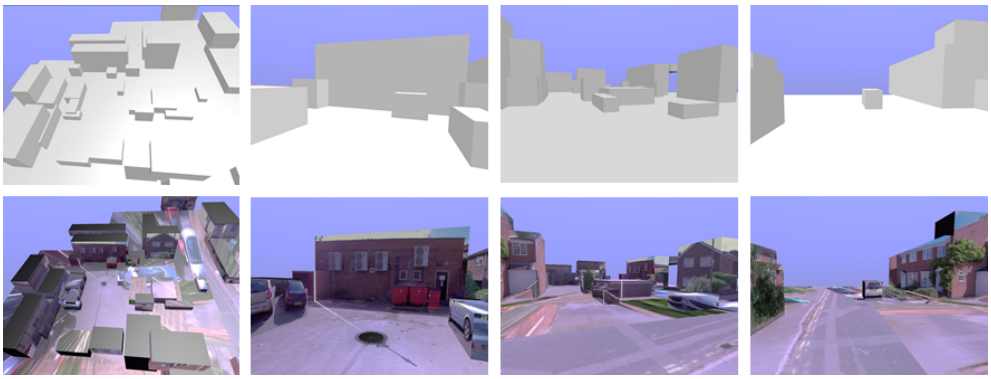
(d) Plaza

(e) Reception

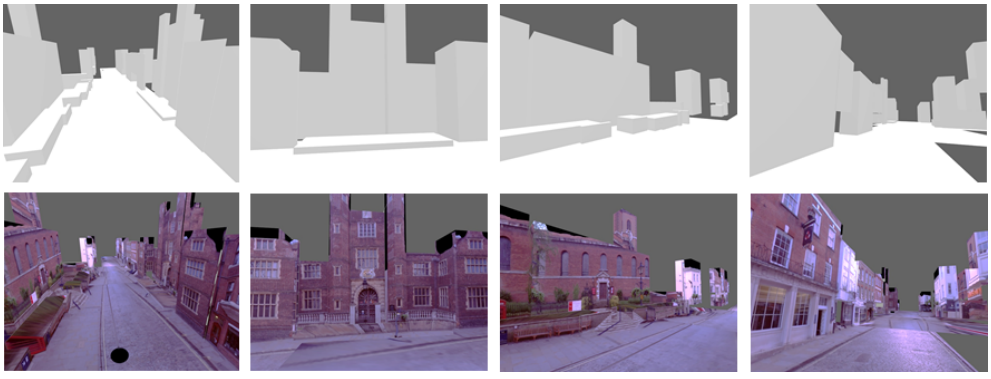
Figure 21: Test datasets (First column: Capture points on maps (from http://maps.google.com), Second column: Spherical capture from three selected points (top image of captured stereo pairs), Three right columns: Façade aligned cubic projection of the selected images



(a) Cathedral

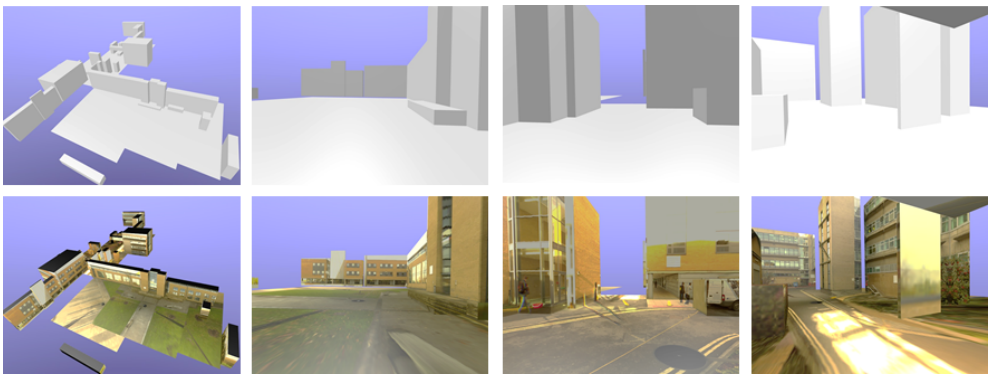


(b) Carpark

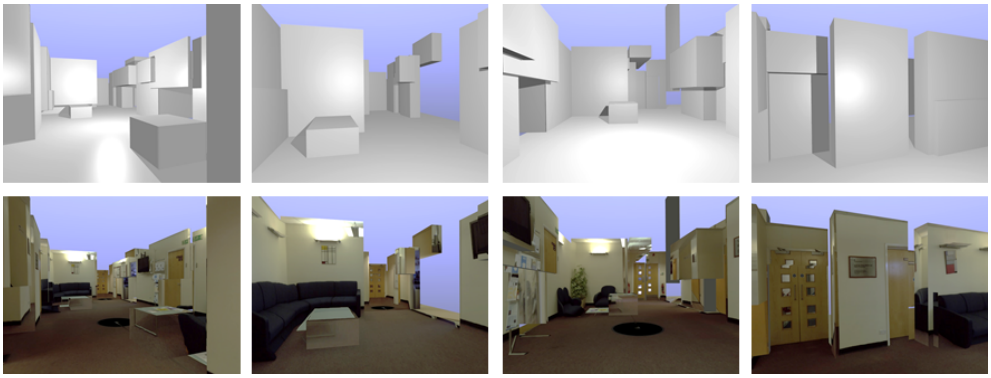


(c) Highstreet

Figure 22: Reconstructed structure and texture mapping result 1



(a) Plaza



(b) Reception

Figure 23: Reconstructed structure and texture mapping result 2

Table 4: Comparison with dense reconstruction method [7]

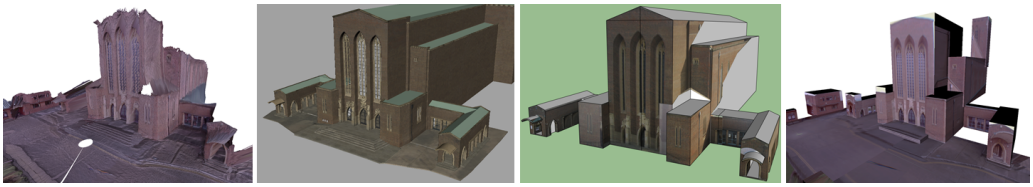
Dataset	Dense reconstruction		Proposed method			
	# of triangles	Data file size	# of planes	Plane data file size	# of cuboids	Cuboid data file size
Cathedral	792,512	81.9MB	34	4.52KB	25	2.65KB
Carpark	480,644	52.7MB	43	5.69KB	35	3.55KB
Highstreet	987,220	108.2MB	45	5.95KB	38	3.82KB
Plaza	1,254,356	132.5MB	48	6.34KB	34	3.46KB
Reception	325,780	40.5MB	50	6.60KB	43	4.27KB

7.3. Comparison with other methods

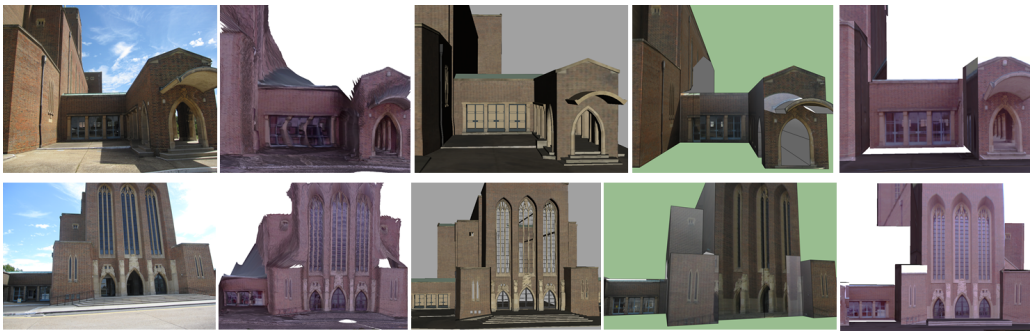
We compared the amount of data produced for the scene reconstruction with dense geometry reconstruction using the approach of Kim and Hilton[7]. We used the same input images and disparity estimation methods for both reconstructions. The dense reconstruction results were saved in the obj format with vertex positions, vertex normals, UV texture and triangle information. The plane primitive information was saved as an ASCII file with the format in Eq. (6) with headers. The cube information was also saved as an ASCII file with the position and length in each direction. Texture index numbers are also included to identify the correct texture for rendering. In Table 4, we see that the size of data required to represent the scenes is reduced by three to four orders of magnitude. The block world provides a compact representation of the scene as a set of 3D cuboid proxies for rendering. Detailed geometry is not represented, but texture mapping enables rendering of the appearance of detailed geometry suitable for scene visualisation.

We also compared visualisation quality, processing time and data file size including texture information for the Cathedral scene with other methods in Fig. 24 and Table 5. The dense reconstruction method [7] produced a huge amount of data. It recovered fine details of the scene but shows geometrical errors in the occluded or ambiguous regions such as the ceiling and windows. The LIDAR model was created from 7 LIDAR scans and dozens of reference stills using MAYA software[6] by a professional CG designer. It is a clean model with high accuracy but it took one full day even by the specialist to build the mesh model and generate textures from the raw sources. The SketchUp result was modelled from nine photographs using the Google

[6]Autodest MAYA, `http://www.autodesk.co.uk/products/maya/`



(a) Final model (From Left: Dense recon., CG from LIDAR, SketchUp and Proposed)



(b) Rendering results (From Left: Photograph, Dense recon., CG from LIDAR, SketchUp and Proposed)

Figure 24: Comparison with other methods (Cathedral)

Table 5: Processing time and volume comparison with other methods (Cathedral)

Method	Dense recon.[7]	CG from LIDAR	SketchUp	Proposed
Processing Time	12 mins	1 day	3 hours	37 secs
File size (inc. texture)	84 MB	12.9 MB	12.5 MB	1.33 MB

SketchUp tool. It took about 3 hours to align vanishing points and geometrical primitives to the original photographs. Blending of multiple photographs for texture mapping resulted in incorrect or blurred textures in some regions. The proposed method is the fastest in building geometry and shows relatively clear structure and texture with the minimum amount of data.

8. Conclusions

In this paper, we propose a block-based simplified 3D scene reconstruction method from spherical stereo image pairs. Vertical spherical stereo pairs are captured at multiple locations in the scene and converted into cubic projection images which are aligned to principal axes. A façade alignment algorithm is proposed which automatically generates central point perspective images

aligned with the principal building faces. This is advantageous in 3D structure reconstruction as it is free from spherical distortion and has a vanishing point at the centre of the image aligned with the principal axes. From the captured images and estimated disparity maps, planar regions are segmented and reconstructed. Reconstructed planes from multiple capture locations are merged and refined to obtain a more complete scene reconstruction. Finally, optimal cuboid structures are reconstructed based on the density of plane primitives.

Results show that the proposed algorithm produces a simplified structured representation of the scene requiring several orders of magnitude less storage compared with dense scene reconstruction. The resulting scene representation provides a compact 3D proxy for visualisation of the scene. Potential future extensions of this research include: 1) Simplified scene reconstruction with arbitrary planes not aligned to the principal axes and various type of 3D structure primitives; 2) Bundle adjustment for large scale loop closure and precise registration; 3) Texture blending and occlusion mapping from multiple viewpoints.

Acknowledgement

This research was supported by the European Commission, FP7 IMPART project (grant agreement No 316564).

References

[1] Y. Furukawa, J. Ponce, Accurate, dense, and robust multiview stereopsis, IEEE Trans. Pattern Analysis and Machine Intelligence 32 (8) (2010) 1362–1376.

[2] H. Vu, R. Keriven, P. Labatut, J. Pons, Towards high-resolution large-scale multi-view stereo, in: Proceedings of CVPR, 2009, pp. 1430–1437.

[3] N. Salman, M. Yvinec, Surface reconstruction from multi-view stereo, in: Proceedings of ACCV, 2009.

[4] C. Strecha, W. Hansen, L. Gool, P. Fua, U. Thoennessen, On benchmarking camera calibration and multi-view stereo for high resolution imagery, in: Proceedings of CVPR, 2008, pp. 1–8.

[5] S. Agarwal, N. Snavely, I. Simon, S. Seitz, R. Szeliski, Building rome in a day, in: Proceedings of ICCV, 2009, pp. 72–79.

[6] M. Pollefeys, D. Nistér, J. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S. Kim, P. Merrell, C. Salmi, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewénius, R. Yang, G. Welch, H. Towles, Detailed real-time urban 3d reconstruction from video, International Journal of Computer Vision 78 (2) (2008) 143–167.

[7] H. Kim, A. Hilton, 3d scene reconstruction from multiple spherical stereo pairs, International Journal of Computer Vision 104 (1) (2013) 94–116.

[8] G. Sellers, J. Obert, P. Cozzi, K. Ring, E. Persson, J. de Vahl, J. M. P. van Waveren, Rendering massive virtual worlds, in: Proceedings of SIGGRAPH 2013 Courses, 2013, pp. 23:1–23:88.

[9] Y. Furukawa, B. Curless, S. Seitz, R. Szeliski, Manhattan-world stereo, in: Proceedings of CVPR, 2009.

[10] A. Gupta, A. A. Efros, M. Hebert, Blocks world revisited: Image understanding using qualitative geometry and mechanics, in: Proceedings of ECCV, 2010.

[11] H. Kim, A. Hilton, Planar urban scene reconstruction from spherical images using facade alignment, in: Proceedings of IVMSP, 2013.

[12] P. Debevec, C. Taylor, J. Malik, Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach, in: Proceedings of SIGGRAPH, 1996, pp. 11–20.

[13] A. V. Hengel, A. Dick, T. Thormählen, B. Ward, P. H. S. Torr, Videotrace: Rapid interactive scene modelling from video, in: Proceedings of SIGGRAPH, 2007.

[14] S. N. Sinha, D. Steedly, R. Szeliski, M. Agrawala, M. Pollefeys, Interactive 3d architectural modeling from unordered photo collections, in: Proceedings of SIGGRAPH ASIA, 2008.

[15] D. Hoiem, A. Efros, M. Hebert, Recovering surface layout from an image, International Journal of Computer Vision 75 (1).

[16] P. Müller, G. Zeng, P. Wonka, L. Gool, Image-based procedural modeling of facades, in: Proceedings of SIGGRAPH, 2007.

[17] J. Xiao, T. Fang, P. Zhao, M. Lhuillier, L. Quan, Image-based street-side city modeling, in: Proceedings of SIGGRAPH ASIA, 2009.

[18] F. Bellotti, R. Berta, R. Cardona, A. D. Gloria, An architectural approach to efficient 3d urban modeling, Computers & Graphics 35 (5) (2011) 1001–1012.

[19] L. Simon, O. Teboul, P. Koutsourakis, N. Paragios, Random exploration of the procedural space for single-view 3d modeling of buildings, International Journal of Computer Vision 93 (2011) 253–271.

[20] M. Mathias, A. Martinovic, J. Weissenberg, L. J. V. Gool, Procedural 3d building reconstruction using shape grammars and detectors, in: Proceedings of 3DIMPVT, 2011, pp. 304–311.

[21] S. Satkin, M. Rashid, J. Lin, M. Hebert, 3dnn: 3d nearest neighbor, International Journal of Computer Vision 111 (1) (2015) 69–97.

[22] G. Schindler, P. Krishnamurthy, F. Dellaert, Line-based structure from motion for urban environments, in: Proceedings of 3DPVT, 2006, pp. 846–853.

[23] C. Hane, C. Zach, B. Zeisl, M. Pollefeys, A patch prior for dense 3d reconstruction in man-made environments., in: Proceedings of 3DIMPVT, 2012, pp. 563–570.

[24] S. Sinha, D. Steedly, R. Szeliski, Piecewise planar stereo for image-based rendering, in: Proceedings of ICCV, 2009.

[25] D. Gallup, J.-M. Frahm, M. Pollefeys, Piecewise planar and non-planar stereo for urban scene reconstruction, in: Proceedings of CVPR, 2010, pp. 1418–1425.

[26] R. Toldo, A. Fusiello, Photo-consistent planar patches from unstructured cloud of points, in: Proceedings of ECCV, 2010.

[27] R. Toldo, A. Fusiello, Robust multiple structures estimation with j-linkage, in: Proceedings of ECCV, 2008.

[28] J. Coughlan, A. Yuille, Manhattan world: orientation and outlier detection by bayesian inference, Neural Computation 15 (5) (2003) 1063–1088.

[29] B. Micusik, J. Kosecka, Multi-view superpixel stereo in urban environments, International Journal of Computer Vision 89 (1) (2010) 106–119.

[30] A.-L. Chauve, P. Labatut, J.-P. Pons, Robust piecewise-planar 3d reconstruction and completion from large-scale unstructured point data, in: Proceedings of CVPR, 2010, pp. 1261–1268.

[31] A. Kowdle, Y.-J. Chang, A. C. Gallagher, T. Chen, Active learning for piecewise planar 3d reconstruction, in: Proceedings of CVPR, 2011, pp. 929–936.

[32] Q.-Y. Zhou, U. Neumann, 2.5d dual contouring: A robust approach to creating building models from aerial lidar point clouds, in: Proceedings of ECCV, 2010, pp. 115–128.

[33] C. Poullis, S. You, 3d reconstruction of urban areas, in: 3DIMPVT, 2011, pp. 33–40.

[34] W. Li, G. Wolberg, S. Zokai, Lightweight 3d modeling of urban buildings from range data, in: Proceedings of 3DIMPVT, 2011, pp. 124–131.

[35] W. Nguatem, M. Drauschke, H. Mayer, Finding cuboid-based building models in point clouds, in: Proceedings of ISPRS, 2012, pp. 149–154.

[36] J. Xiao, Y. Furukawa, Reconstructing the worlds museums, International Journal of Computer Vision 110 (3) (2014) 243–258.

[37] P. Sturm, A method for 3d reconstruction of piecewise planar objects from single panoramic images, in: Proceedings of IEEE Workshop on Omnidirectional Vision, 2000, pp. 119–126.

[38] H.-D. Kang, K.-H. Jo, 3d reconstruction of planar objects using the properties of plane and vanishing points from a single panoramic image, in: Proceedings of World Multi-Conference on Systemics, Cybernetics and Informatic, 2006, pp. 342–346.

[39] B. Micusik, J. Kosecka, Piecewise planar city 3d modeling from street view panoramic sequences, in: Proceedings of CVPR, 2009, pp. 2906–2912.

[40] D. Anguelov, C. Dulong, D. Filip, C. Frueh, S. Lafon, R. Lyon, A. Ogale, L. Vincent, J. Weaver, Google street view: Capturing the world at street level, IEEE Computer 43 (6) (2010) 32–38.

[41] D. Feldman, D. Weinshall, Realtime ibr with omnidirectional crossed-slits projection, in: Proceedings of ICCV, 2005, pp. 839–845.

[42] P. F. Felzenszwalb, D. P. Huttenlocher, Efficient graph-based image segmentation, International Journal of Computer Vision 59 (2).

[43] A. Banno, K. Ikeuchi, Omnidirectional texturing based on robust 3d registration through euclidean reconstruction from two spherical images, Computer Vision and Image Understanding 114 (4) (2010) 491–499.

[44] J. Matas, C. Galambos, J. Kittler, Robust detection of lines using the progressive probabilistic hough transform, Computer Vision and Image Understanding (2000) 119–137.

[45] D. Eberly, Least Squares Fitting of Data, `http://www.geometrictools.com/Documentation/LeastSquaresFitting.pdf`, [Online; accessed 22-Aug-2013] (2008).

[46] R. Schnabel, R. Wahl, R. Klein, Efficient ransac for point-cloud shape detection, Computer Graphics Forum 26 (2) (2007) 214–226.

[47] A. Hilton, Scene modelling from sparse 3d data, Image and Vision Computing 23 (10) (2005) 900–920.

[48] Y. Furukawa, B. Curless, S. M. Seitz, R. Szeliski, Reconstructing building interiors from images, in: Proceedings of ICCV, 2009.

[49] H. Bay, A. Ess, T. Tuytelaars, L. Gool, Surf: Speeded up robust features, Computer Vision and Image Understanding 110 (2008) 346–359.

[50] T. Mullen, Mastering Blender, 1st Edition, Wiley Publishing, Inc., 2009.

[51] S. Seitz, B. Curless, J. Diebel, D. Scharstein, R. Szeliski, A comparison and evaluation of multi-view stereo reconstruction algorithms, in: Proceedings of CVPR, 2006, pp. 519–528.