



An image-based approach to the reconstruction of ancient architectures by extracting and arranging 3D spatial components*

Divya Udayan J¹, HyungSeok KIM^{†1}, Jee-In KIM²

(¹Internet and Multimedia Engineering, Konkuk University, Seoul 143-701, Korea)

(²Department of Advanced Technology Fusion, Konkuk University, Seoul 143-701, Korea)

E-mail: divuda@konkuk.ac.kr; hyuskim@konkuk.ac.kr; jnkm@konkuk.ac.kr

Received Apr. 20, 2014; Revision accepted Aug. 24, 2014; Crosschecked Dec. 10, 2014

Abstract: The objective of this research is the rapid reconstruction of ancient buildings of historical importance using a single image. The key idea of our approach is to reduce the infinite solutions that might otherwise arise when recovering a 3D geometry from 2D photographs. The main outcome of our research shows that the proposed methodology can be used to reconstruct ancient monuments for use as proxies for digital effects in applications such as tourism, games, and entertainment, which do not require very accurate modeling. In this article, we consider the reconstruction of ancient Mughal architecture including the Taj Mahal. We propose a modeling pipeline that makes an easy reconstruction possible using a single photograph taken from a single view, without the need to create complex point clouds from multiple images or the use of laser scanners. First, an initial model is automatically reconstructed using locally fitted planar primitives along with their boundary polygons and the adjacency relation among parts of the polygons. This approach is faster and more accurate than creating a model from scratch because the initial reconstruction phase provides a set of structural information together with the adjacency relation, which makes it possible to estimate the approximate depth of the entire structural monument. Next, we use manual extrapolation and editing techniques with modeling software to assemble and adjust different 3D components of the model. Thus, this research opens up the opportunity for the present generation to experience remote sites of architectural and cultural importance through virtual worlds and real-time mobile applications. Variations of a recreated 3D monument to represent an amalgam of various cultures are targeted for future work.


Key words: Digital reconstruction, 3D virtual world, 3D spatial components, Vision and scene understanding
doi:10.1631/FITEE.1400141 **Document code:** A **CLC number:** TP391.7

1 Introduction

Cultural heritage (Remondino, 2011) is the legacy of physical artifacts inherited from past gen-

[†] Corresponding author

* Project partially supported by the Ministry of Culture, Sports and Tourism and Korea Creative Content Agency in the Culture Technology Research & Development Program 2014 (50%), and the Next Generation Information Computing Development Program through the National Research Foundation of Korea funded by the Ministry of Science, ICT and Future Planning (No. 2012M3C4A7032185) (50%)

 ORCID: HyungSeok KIM, <http://orcid.org/0000-0003-4816-2992>

© Zhejiang University and Springer-Verlag Berlin Heidelberg 2015

erations that should be passed down to future generations. There is a social and cultural responsibility to preserve and maintain ancient works. Small objects such as artwork and other cultural masterpieces are collected in museums and art galleries. However, large monuments face threats from natural and human factors. Thus, a need to use modern technologies for the interpretation, conservation, and preservation of our cultural heritage has arisen. There are two main categories of digital reconstruction depending on the target application areas. The first category deals with the reconstruction of 3D

models for archaeological studies and historical references. This requires a very accurate 3D reconstruction of the available models. The second category is architecture related with applications in virtual tourism, games, and entertainment. In such applications, the virtual visualization is more important than the accuracy of the 3D model. Our approach can be used in the field of architecture. In such applications, obtaining good multiple images of buildings and their camera parameters is often difficult and time-consuming for a casual user, as the different view photographs will have different lighting and shading effects. Therefore, for fast reconstruction of buildings we use a single image in our approach as we need to consider only minimum fluctuations in light intensity. Our approach can be used even by casual users for applications such as games, tourism, personal web pages, and entertainment. Users can use additional images to refine the 3D models of buildings if necessary, using previous works.

In this article, we consider the 3D reconstruction of ancient monuments from a single image using a case study based on Mughal architecture. Three-dimensional reconstruction from a single image is possible only if additional object information is available. The main type of prior information needed is the parallelism of straight object edges. Straight and parallel object edges are frequently present in man-made structures and buildings in particular. Therefore, reconstructions of 3D buildings are more feasible using single images than using multiple images provided with prior structural information.

Our reconstruction approach includes data acquisition from a single photograph taken from a single view of the monument, which reduces the cost and labor of using 3D scanners to create point clouds of an entire building. Single view reconstruction can also be used to reconstruct buildings that are non-existent today and are available only from single photographs. The key idea of our approach is to reduce the infinite number of solutions that might otherwise arise when recovering a 3D geometry from 2D photographs by estimating the abstract geometric shape features of the components, followed by capturing detailed structural information and component filling. First, an initial model is automatically constructed using locally fitted planar primitives along with their boundary polygons and the adjacency relation among parts of the polygons. This helps to

create a rough estimation of the geometric components and their positions in the monument. The structural information together with a hierarchical analysis makes it possible to detect automatically the optimal symmetry lines and estimate the depth of the entire structural monument. In this research, we used the Taj Mahal as our primary candidate monument for a structural study because it is one of the most complex ancient monuments still in existence, and then we extended our approach to other ancient buildings.

1.1 Study of structural information from architectural style

The Taj Mahal is a beautiful monument owing to its geometric shapes and symmetry (Fig. 1). According to historians, the master plan of the Taj Mahal was ordered through orthogonal grids. It is possible to infer some structural details from the architecture of the Taj Mahal by closely examining the main characteristics of Mughal architecture. The main characteristics of Mughal architecture are as follows (Encyclopedia, 2014): (1) Iwans, i.e., vaulted spaces, surrounded by three walls and an opening.

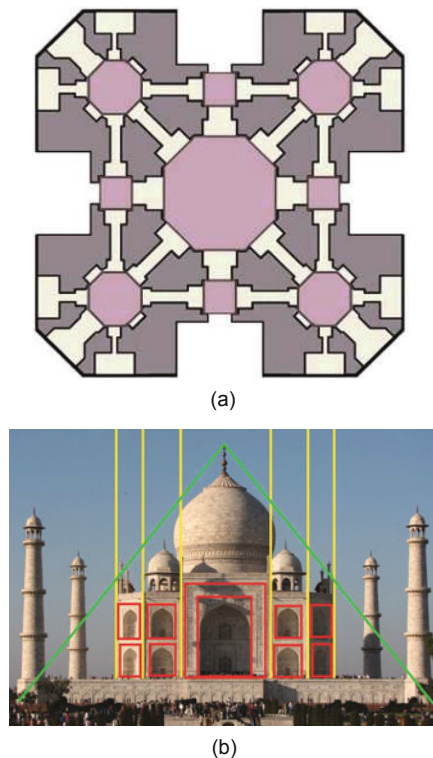


Fig. 1 The Taj Mahal monument: (a) nine-fold plan; (b) rectangular framing grid

This architectural feature is built to resemble a gateway, and is used extensively for both religious and secular buildings. (2) Arches built not only for aesthetic reasons, but also to place Masyrabias windows and lessen the reach of sunlight into the building. (3) Bulbous domes usually placed on top of cylindrical drums, before tapering to a point and decorated with a finial. In Mughal architecture, multiple smaller domes can also be seen decorating the rooftops of buildings. (4) Muqarnas, i.e., stalactite-like decorations, commonly used under arches, especially under the vaults of the Iwans. (5) Calligraphy used for decorative accents around the gate of an Iwan, under cornices, and around the arches surrounding the building. (6) Mashrabias, i.e., pierced screens used as windows, placed extensively throughout the building.

The structural inferences that are possible from a study of the above characteristics are the symmetry, geometric components, and component repetition.

1. Symmetry: Mughal architecture has symmetry as a distinguishing feature; i.e., the buildings have the same number of minarets, arches, and pillars on each side. Even the pools and gardens are designed to provide a mirror-like symmetry. The Taj Mahal was built with perfect symmetry. The mausoleum is raised over an enriched version of the nine-fold plan shown in Fig. 1a, which was used by the Mughals for tombs and garden pavilions. This nine-fold plan begins with a square or rectangle shaped structure, the corners of which are typically squared off to form an irregular octagon. This basic shape is further divided into nine discrete but interconnected rooms: a central domed chamber, surrounded by eight rooms, four in the corners and one in each of the spaces between the corner rooms. The mausoleum plan is expressed with perfect cross-axial symmetry, such that the building is focused on the central tomb chamber. This bilateral symmetry is symbolic of rulers aiming at absolute power as an expression of the ruling force, which brings about balance and harmony.

2. Geometric components: The geometric components also follow a symmetrical design. The main structure is cubical. The central circle at the base arch tapers upwards to create a half-onion shaped dome. The windows have arched recesses that come to a point and can be observed on both stories. The

doorways are rectangular. The hall in which the tombs are placed is octagonal. Multiple lines of symmetry can be seen. Another notable feature is that the Taj Mahal was formed through a combination of planes.

3. Repetition: The basic geometric primitive in the design of the Taj Mahal is the rectangle. This core element repeats itself throughout the facade: the main portal (four rectangles), the smaller opening of the main entrance (one rectangle), the openings at the chamfered corner (one-half rectangle), the space between the top of the portal and the base of the dome (two rectangles), and the giant dome itself (four rectangles). This rectangular framing grid is the most outstanding feature of Mughal design (Fig. 1b). The dome structure also repeats, with a central bulbous dome and other smaller domes with four minarets located in four directions. We derived our approach for 3D reconstruction based on these structural inferences. First, we determine the geometric components of the monument in question from a single image. Second, we identify and analyze the symmetrical structure and recreate it. Third, we fill in the geometric components using the property of repetition.

2 Related works

Digital representation and reconstruction from images has been used in the field of computer vision for years. Three-dimensional building models are needed for many applications, including enhanced augmented reality applications where the models are registered with real world scenes (Pylvanainen *et al.*, 2012; Yang *et al.*, 2013) to facilitate display, interaction, and effective rendering.

Three-dimensional reconstruction of buildings based on processing visual data (images and video frames) is a fast growing field of research. Even though the possibilities offered by digital technologies are large, the high costs that are involved in their usage, the lack of standard procedures, and the unavailability of experts often limit the widespread digitization of cultural heritage sites (Frahm *et al.*, 2010; Styliadis and Sechidis, 2011; Manferdini, 2012; Garcia-Gago *et al.*, 2014). As a consequence, researchers of image-based technologies and, in particular, structure from motion (SfM) (Wang and Olano, 2011; Ceylan *et al.*, 2014), aim at managing and

visualizing the 3D data through the web. SfM is based solely on image correspondences and has a lot of applications, as summarized by Wei *et al.* (2013). This approach has extended the use of digital technologies and procedures in this field.

We can classify image-based representation techniques into several categories, two of which are of interest to us. One is a system based on image depth representation based on building layouts. Works related to this category include Shade *et al.* (1998) and Oh *et al.* (2001). This representation is simple, easy to render, and permits direct user control. However, these techniques require either view warping to assign depth or disparity per pixel (Chen and Williams, 1993; Laveau and Faugeras, 1994; McMillan and Bishop, 1995; Shade *et al.*, 1998), or a painting metaphor (Kang, 1998) to recover depth. This leaves the depth assignment task to the user's expertise. The facade system (Debevec *et al.*, 1996) models architectural scenes from one or more images using a collection of simple primitives, and through user assistance. Another framework based on building layouts to characterize analytically the local shape space was proposed by Bao *et al.* (2013). A recent study on image structural analysis (Zhang *et al.*, 2013) was based on a hierarchical and layered analysis of irregular facades for a high-level understanding of facade structures. Interactive facades improved multi-view stereo reconstruction and novel image editing possibilities (AlHalawani *et al.*, 2013). Another structure recovery approach from single images was proposed by Shen *et al.* (2012) enabling the generation of new semantically meaningful structures by assembling existing labeled parts with respect to the acquired data. A good constraint for recovering image depth is symmetry and regularity (Mitra and Pauly, 2008; Mitra *et al.*, 2013). Ceylan *et al.* (2012) proposed a framework for image-based 3D reconstruction of buildings based on symmetry priors. In our work, we focus on approaches to finding, extracting, encoding, and exploiting geometric symmetries and high-level structural information from a single image.

The second category is concerned with systems that use a set of images as input to build a more traditional geometric representation (Faugeras *et al.*, 1995; Debevec *et al.*, 1996; Poulin *et al.*, 1998; Liebowitz *et al.*, 1999; Zhang *et al.*, 2002). This type of traditional approach uses a particular cue

such as the shape from shading (Horn, 1990), texture (Super and Bovik, 1995), or vanishing points (Guilou *et al.*, 2000). The methods based on texture-dependent cues are difficult to apply on surfaces that do not have a uniform textural appearance. Nagai *et al.* (2007) used hidden Markov models for inferring depth from single images for objects such as hands and faces. Criminisi *et al.* (2000) proposed an interactive method for computing 3D geometry, where the user can specify the region of interest, 3D coordinates of some points, and the reference height of an object. Saxena *et al.* (2008a) showed that depth can be predicted from monocular image features. The algorithm was later used for improving the performance of stereovision, especially in robotic applications. Although all these cues allow a strong assumption about the shape of a structure, they tend to impose more constraints on geometry, as they support only a restricted class of images, making it difficult to recover the geometry of complex architectures such as those used in ancient Mughal designs. Also, these methods provide no or less user editing capability.

Our goal is to combine these two approaches and build a flexible image-based reconstruction of geometric primitives and meshes given a single image, providing fewer constraints on geometry and more user editing capability. Some of our depth-acquisition techniques are based on a generalization of the approach proposed by Horry *et al.* (1997), in which a central perspective and user-defined billboards are needed to estimate depth from a 2D image. Our method is not restricted to the geometry of a scene, but involves a faster depth estimation of the geometric components using structural knowledge of the architectural style, and thereby the creation of a component library related to that style.

3 Digital reconstruction methodology

Section 1.1 describes the structural information gathered from our study. We use this structural information to distinguish between the different components, and later aggregate them to reconstruct a 3D building. Section 3.1 describes an overview of the proposed digital reconstruction approach.

3.1 Overview

The overall reconstruction methodology involves a series of modules which can be broadly

fitted under three steps. The process begins with a pre-processing step which involves the resizing of the 2D image and the generation of a multi-scale representation of the image to be used later in the depth inference phase. Fig. 2 shows the pipeline of the overall process.

In the structural information capturing phase (Fig. 2a), components are extracted and their positions identified using structural analysis and architectural style information. In classical architecture, buildings are composed of architectural elements. These elements are logically organized hierarchically in space to produce the full structure. It is possible to infer prior information from architectural components like parallelism and the perpendicularity of straight object edges. Also, some structural inferences, like columnar elements, consist of a capital; a column itself consists of a long vertical tapered cylinder, and a pedestal or base can be considered as additional structural information.

The geometry estimation phase includes feature extraction, depth inference, and mesh generation (Fig. 2b). The image is divided into non-overlapping patches $P = \{P_1, P_2, \dots, P_N\}$, as the depth information from a single image is derived from the designed feature set. The feature set is designed using certain monocular cues like size, shade, distortion, and vanishing points. One of the approaches considered in our work is the shape-from-texture technique which uses cues from image plane variations in the texture properties such as density, size, and orientation. For example, the orientation of the surface is determined by the texture gradient. Another approach considered in our work for recovering 3D geometry infor-

mation is shape from inverse perspective projection, in conjunction with geometric constraints on the 3D geometric entity. The geometric constraints include Euclidean distance constraints and orientation constraints. Note that the camera parameters are not needed explicitly as they are calculated implicitly during the reconstruction process. For 3D reconstruction from a single image, the camera parameters can be estimated if sufficient information about parallelism and perpendicularity constraints is available. The actual depth between the components is inferred from the estimated likelihood of the neighboring patches as a result of the feature extraction module. A connected mesh structure is generated by estimating the plane upon which each patch lies (its 3D position and orientation) and approximating its outline using coarse polygons. The final mesh is constructed by positioning the edges and faces relative to each other by using common points. The orientations of the object edges and faces are found by using parallelism and perpendicularity constraints.

The component filling phase maps the components to their identified positions (Fig. 2c). We use a pre-built component library related to each architectural style. The components are mapped to their actual locations in the mesh generator, and further manual adjustments are done to create the final 3D model. The component library can be updated with the edited components and can be reused later for modeling similar style buildings.

3.2 Component extraction

The first step in our approach is to extract the geometric components of a monument from a

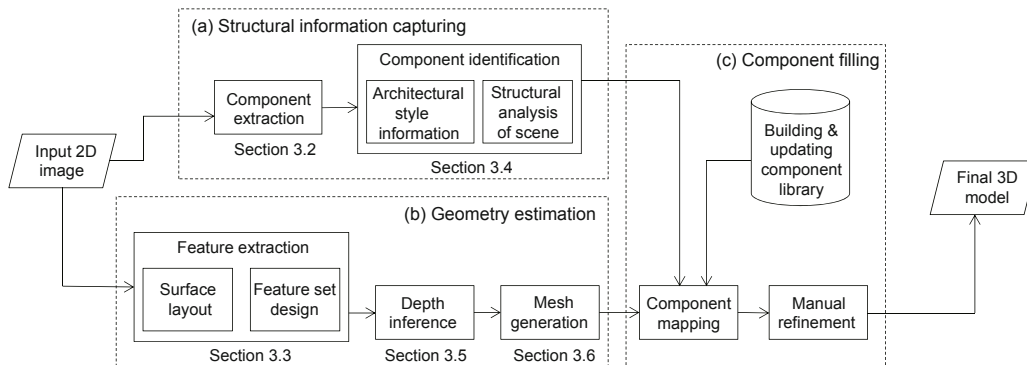


Fig. 2 Pipeline of the overall process: (a) structural information capturing; (b) geometry estimation; (c) component filling

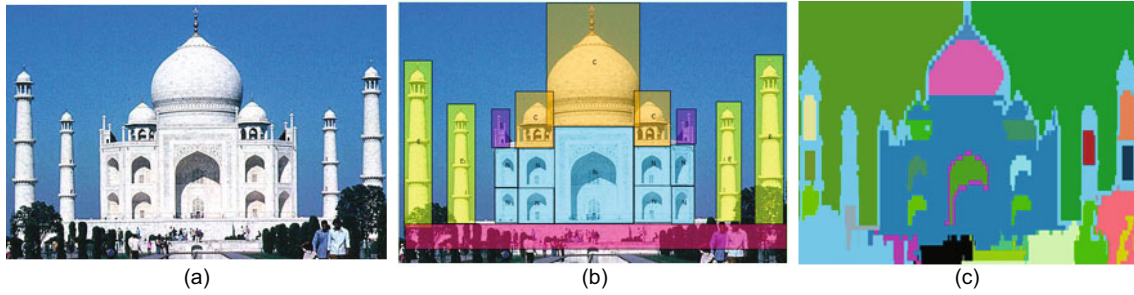


Fig. 3 Component extraction results of the input image: (a) digital input image; (b) identified components and their repetition shown in a unique color; (c) feature-based surface layout results

single image (Fig. 3). The system computes an intensity score using a normalized correlation coefficient (NCC) to identify a repeated region in an image as such regions are resilient to intensity variations due to lighting and shadow fluctuations (Ma *et al.*, 2010). There are many other feature detection and matching algorithms, like speeded-up robust features (SURFs) (Lowe, 2004) and scale-invariant feature transform (SIFT) (Bay *et al.*, 2006). These algorithms are good at feature tracking and finding correspondence matches, but not for component identification from a single image where components are frequently repeating. Feature points are too small and unevenly distributed. To make these algorithms suitable for our approach, optimization needs to be done using algorithms like RANSAC (Dung *et al.*, 2013).

To remediate this problem, the system identifies the components in the image using two approaches for a better understanding. The first approach involves template matching using NCC, and the second uses features or cues from the 2D image, which is detailed later in Section 3.3.

To begin with, we use a semi-automated approach to identify the structural components and number of repetitions of each geometric component inside a 2D image $I(x, y)$, where x and y are the coordinates of each pixel in the image. The user draws a rectangle over any one component represented by a template $T(x, y)$. The template $T(x, y)$ is shifted into nine different positions. The intensities are multiplied and summed at each position, producing a correlation coefficient matrix.

Let $I(x, y)$ be the intensity value of the $M_x \times M_y$ image I at pixel (x, y) , $x \in \{0, 1, \dots, M_x - 1\}$, $y \in \{0, 1, \dots, M_y - 1\}$. Similarly, let $T(x, y)$ be the intensity of the $N_x \times N_y$ template T at pixel (x, y)

where $N_x \leq M_x$ and $N_y \leq M_y$. NCC is evaluated at every point (u, v) for I and T , which is shifted over from the original image $I(x, y)$ by u steps in the x direction and v steps in the y direction. The formula for the (u, v) th entry of the correlation matrix is as follows:

$$R_{u,v} = \frac{\sum_{x,y} (I(x, y) - \bar{I}_{u,v})(T(x - u, y - v) - \bar{T})}{\sqrt{\sum_{x,y} (I(x, y) - \bar{I}_{u,v})^2 \sum_{x,y} (T(x - u, y - v) - \bar{T})^2}}, \quad (1)$$

where $\bar{I}_{u,v}$ denotes the mean value of $I(x, y)$ within the area of template T shifted by (u, v) steps, and is given by the following formula:

$$\bar{I}_{u,v} = \frac{1}{N_x N_y} \sum_{x=u}^{u+N_x-1} \sum_{y=v}^{v+N_y-1} I(x, y), \quad (2)$$

and \bar{T} denotes the mean value of template T , defined in the same way.

The NCC has the following advantages which make it well suited for component identification: (1) The NCC is brightness invariant; i.e., in the case of changes in external illumination, the NCC will not change. (2) NCCs are robust to blurring. Even if the NCC changes with the blurring of the template, the position of its maxima will not change. This is important if images are taken at high view angles as they may suffer from serious blurring. (3) The NCC is usually fast to calculate. (4) The calculated NCCs will be evenly distributed across the whole image.

However, there are also some clear limitations of using the NCC. The NCC can determine a numerical value only between 0 and 1, where 0 means ‘no match’ and 1 means ‘identical’, To obtain sub-pixel accuracy, a second-order polynomial around the position of the NCC maximum is established. The

template $T(x, y)$ is shifted into nine different positions to determine the second-order polynomial, applying the least-squares method.

Fig. 3b shows the results of the identified components and their repetitions from the input 2D image of the Taj Mahal. Similar components are labeled using a single unique color to distinguish each structural part.

The second approach for component identification uses features or cues from the given 2D image.

3.3 Feature extraction

The 2D image is divided into non-overlapping patches $P = \{P_1, P_2, \dots, P_N\}$. Depth information from a single image is represented by a composite feature vector derived from the designed feature set. The goal of this module is to find the features within each patch and its neighboring patches. The feature extraction phase consists of surface layout and feature set design steps.

3.3.1 Surface layout

To classify regions in the image and extract the feature set, we use a robust graph-based segmentation method (Felzenszwalb and Huttenlochet, 2004). Our goal is to determine the surface layout from an image. For this, we divide the image into smaller regions that provide spatial support to the color and texture features. In a graph-based approach, the whole image is represented by a graph, $G = (V, E)$, where each node $v_i \in V$ represents the pixels in the image, and each edge $(v_i, v_j) \in E$ represents that the pair of neighboring pixels v_i and v_j is connected. An edge weight $w_{ij}(v_i, v_j)$ is associated with each edge based on a certain property. In our case, the edge weights depend on the difference in intensity of the edges:

$$w_{ij} = |I(v_i) - I(v_j)|, \quad (3)$$

where $I(v_i)$ is the intensity of v_i , and $I(v_j)$ is the intensity of v_j . Each vertex is considered the smallest sub-graph at the beginning of the segmentation.

After constructing a graph, we merge the sub-graphs with similar intensity levels to form larger sub-graphs (i.e., non-overlapping sub-regions). We use a pre-defined predicate to measure the evidence for a boundary between two sub-graphs. In this method, we also use a predicate to determine whether two neighboring sub-regions (i.e., connected

sub-graphs) should be merged. Given a graph $G = (V, E)$, the resulting predicate $P(C_1, C_2)$ is used to compare the inter-sub-graph differences with the within-sub-graph differences:

$$P(C_1, C_2) = \begin{cases} \text{TRUE}, & \text{Diff}(C_1, C_2) \leq \text{MInt}(C_1, C_2), \\ \text{FALSE}, & \text{otherwise}, \end{cases} \quad (4)$$

where $C_1, C_2 \subseteq G$, $\text{Diff}(C_1, C_2)$ is the difference between the two sub-graphs, and $\text{MInt}(C_1, C_2)$ is the minimum internal difference within each sub-graph. If the difference between the two sub-graphs is smaller than their minimum internal difference, the sub-graphs should be merged.

Initially, each vertex in the constructed graph is considered an isolated sub-graph, and each edge is treated as invalid. The edges are then sorted according to their weights in non-decreasing order and then traversed. If two vertices connected by an edge being traversed belong to two different sub-graphs, their boundary can be discarded, resulting in the edge being set as valid, and the two sub-graphs are merged to form a larger sub-graph. Having traversed all edges, a collection of trees, each of which is a minimum spanning tree (MST), can be obtained. To obtain an MST, we use the Kruskal method (Gormen *et al.*, 1990). As each MST corresponds to a sub-region of the image, the pixels located in a sub-region are significantly similar in their feature space, and the pixels from a different sub-region will be significantly different. The surface layout results from a 2D image of the Taj Mahal are shown (Fig. 3c).

3.3.2 Feature set design

The design of a feature set depends on the sensitivity to various aspects of the shape, albedo, shading, and viewpoint. There are three basic principles for selecting the features (Lowe, 2004). First, we should ensure that all relevant information is captured. Second, to improve the accuracy of both the training and test sets, the number of features should be minimized. Third, the designed features should be independently predictive, leading to simpler decision boundaries for an improved generalization. The design of the feature set in our approach is based on these three principles to encode cues about the properties of each region, such as (1) its size, shape, and position, (2) the material properties from the color and texture, (3) the surface orientation from

the texture, (4) histograms of the orientations, and (5) intersections of the straight line segments. To extract the features, we use a human-like inference methodology.

Humans use prior knowledge of a similar environment and monocular depth cues such as variations in texture, object size, haze, and defocus to infer depth from a single image. For example, the texture may vary for different users at different distances. A texture gradient provides information on the flow direction of the pixels, and is a valuable source of depth cues. Haze is another cue that we considered. Haze is caused by atmospheric light scattering and can provide texture information to determine depth. Although there are many possible feature extraction methods, the same basic features apply to all, including the color, gradients, histograms of the gradients, edges, position, and region shape. Fig. 4 shows the gradient features and edges of the input image. The image under consideration is a color or RGB image in which the texture is distributed by red, green, and blue color channels. In addition, the greatest variation in color is due to changes in luminance or brightness. Thus, to integrate the texture information into a single intensity channel,

we convert the RGB image into the YCbCr format (Fig. 5), in which we obtain the intensity information in the Y intensity channel and the two color channels. There are also other formats available such as HSV and CIELAB. HSV decomposes an image into hue, saturation, and value, but the angular measurement of hue (e.g., when the maximum and minimum values are both red) can be disturbing and affect the accuracy of the extracted features. CIELAB was designed to be perceptually uniform so that small changes in color with an equal Euclidean distance will be perceived by the vision algorithms as having similar degrees of change. For these reasons, in our approach we prefer to use the YCbCr color space.

The intensity information is extracted and convoluted using nine Laws filter masks (Davies, 2005) (Fig. 6a). The extraction of low-frequency information, such as haze and blurriness, is obtained by convolving the color channels Cb and Cr using two local averaging filters. The texture gradient is obtained by convolving the intensity channel with six oriented edge filters (Nevatia and Babu, 1980) (Fig. 6b). The outputs of the filters are shown in Fig. 7. The output of the convolution of image $I(x, y)$ with filters F_n ($n = 1, 2, \dots, 17$) provides a weighted absolute

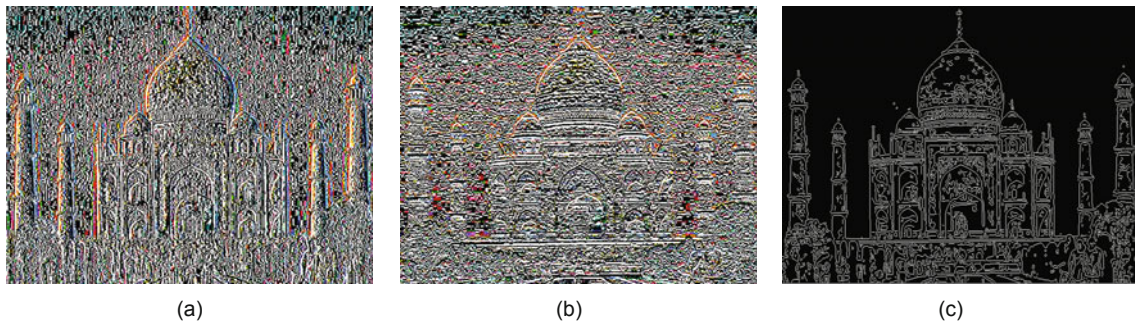


Fig. 4 Gradient features for X coordinate (a) and Y coordinate (b), and the edges (c) for the input image

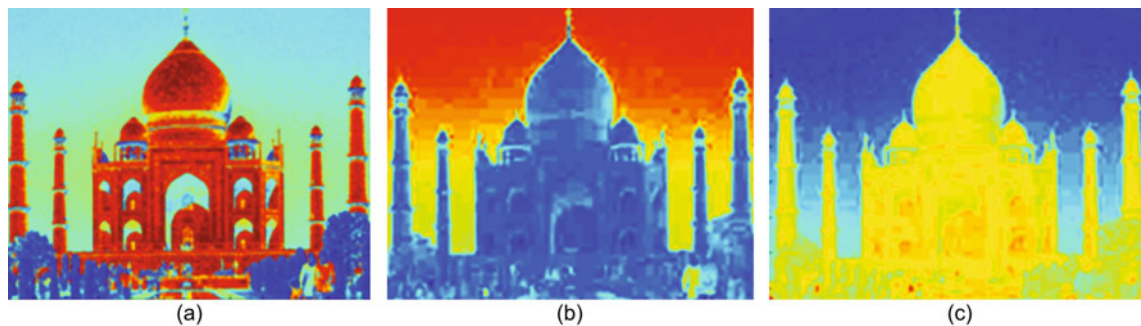


Fig. 5 Image conversion from RGB into YCbCr

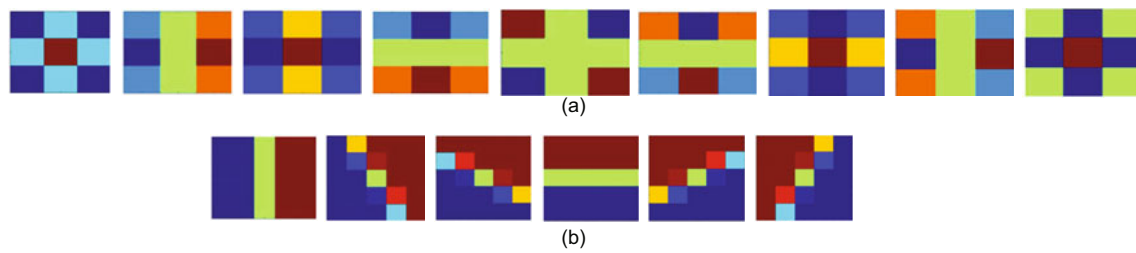


Fig. 6 Filters used for feature extraction: (a) Laws filter masks (nine 3×3 filters); (b) texture gradient filters

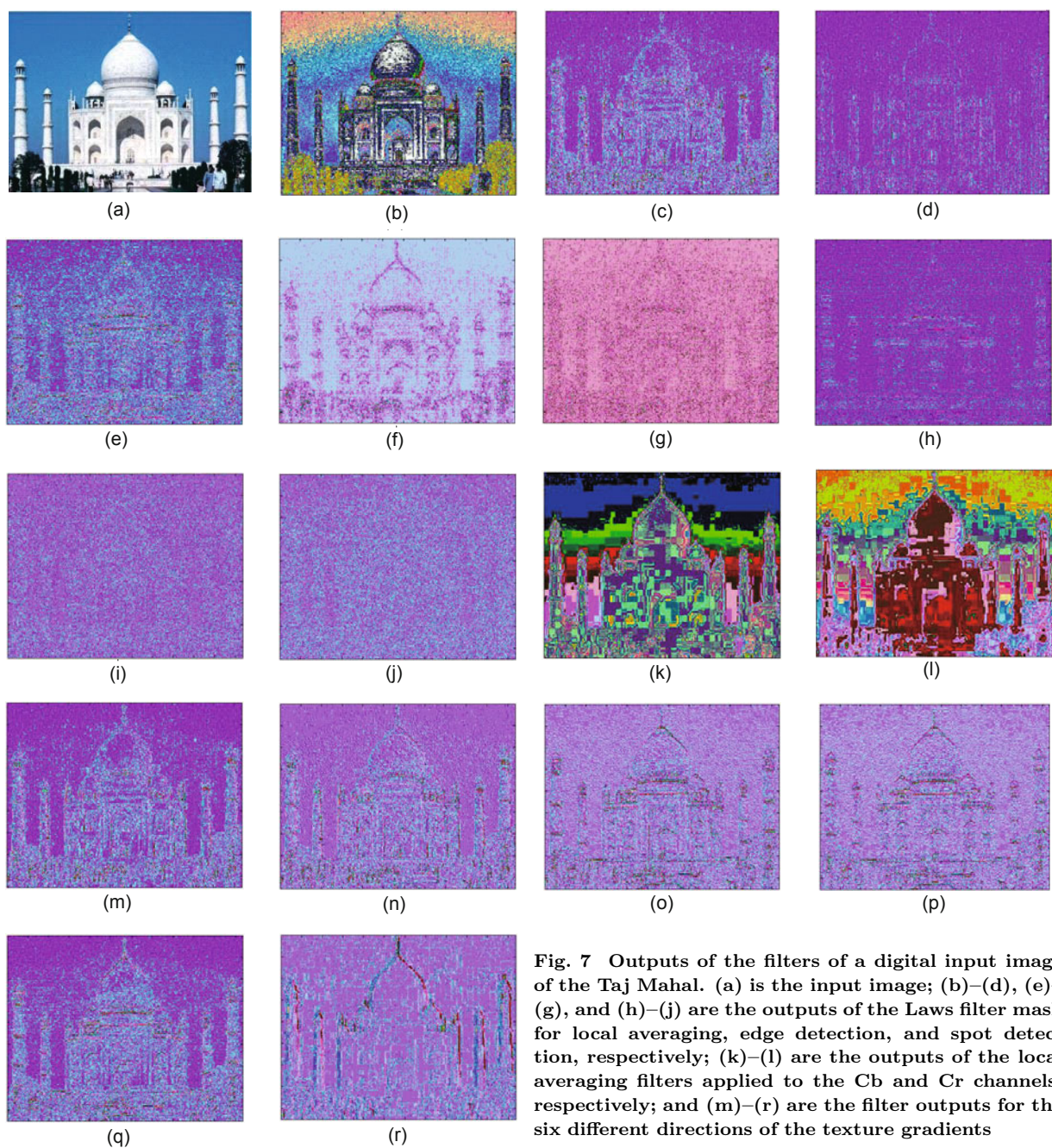


Fig. 7 Outputs of the filters of a digital input image of the Taj Mahal. (a) is the input image; (b)–(d), (e)–(g), and (h)–(j) are the outputs of the Laws filter mask for local averaging, edge detection, and spot detection, respectively; (k)–(l) are the outputs of the local averaging filters applied to the Cb and Cr channels, respectively; and (m)–(r) are the filter outputs for the six different directions of the texture gradients

energy sum and a squared energy sum, respectively, and results in a 34-dimensional feature vector, an element (two-tuple) of which is

$$E(n) = \sum_{(x_p, y_p) \in P} |I(x, y)F_n(x, y)|^k, \quad (5)$$

where $k = 1, 2$, and x_p and y_p are the pixel coordinates of patch $P = \{P_1, P_2, \dots, P_N\}$.

3.4 Component identification

A 2D photograph of a scene is not simply an image. It also contains a large amount of geometric information about the captured scene (Hoiem *et al.*, 2005). Our method results in a spidery mesh interface that contains the approximate depth parameters of the scene.

We first perform an automatic structural analysis of the scene. An analysis of the structural arrangement of the components provides an idea of the geometric position of each component. Thus, the number of infinite solutions that might otherwise result from a 3D recovery of a 2D scene can be reduced. Depending on the structural arrangement, the system automatically determines (1) the position of the geometric components, (2) the optimum symmetry line, and (3) the component repetitions. Müller *et al.* (2007) proposed an approach to procedurally model the components based on image facades. Using an idea from this approach, we focus on a single frontal view and symmetry for creating a component library for the different parts of the monument in question.

Fig. 8 shows the automatic structural analysis results of the input image of the Taj Mahal. Depending on each architectural style, the components are created in a component library. During the structural analysis, we can easily map each colored block that represents the component to its corresponding component in the 3D component library. Components can be reused later for similar architectural style buildings. Additional experiments were conducted on two other Mughal architectural designs: Humayun’s Tomb and the Badshahi Mosque. Table 1 shows the time required for a structural analysis and the number of components identified for each monument.

Table 1 Results of the structural analysis of different monuments

Monument	Time required for structural analysis (s)	Number of identified components
Taj Mahal	72	18
Humayun’s Tomb	45	27
Badshahi Mosque	55	22

3.5 Depth inference

Architecture modeling and rendering from photographs were suggested early by Debevec *et al.* (1996). In our process used for creating a component library, the actual depth relative to each component is inferred from the estimated likelihood of the neighboring patches of the feature extraction module. The key idea is to find the plane upon which each identified patch lies (its 3D location and 3D orientation).

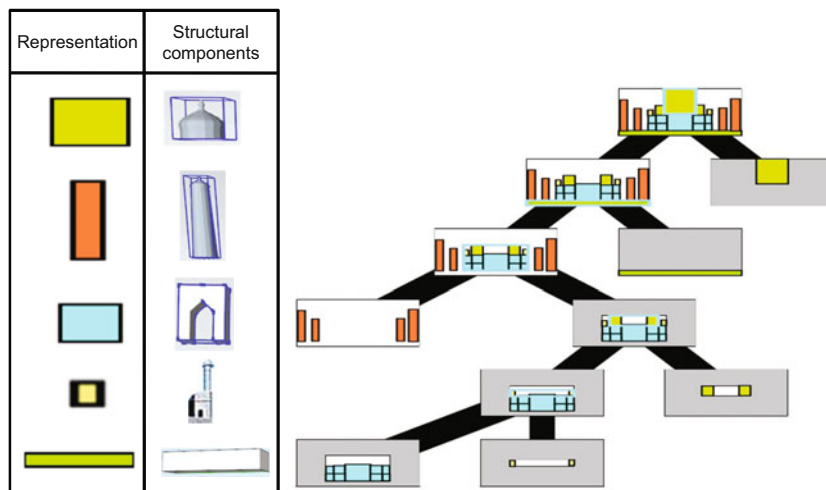


Fig. 8 Automatic structural hierarchical analysis of the components of the Taj Mahal and its mapping using a 3D component library

We used a database of trained images taken from the Stanford Make3D trained dataset (Saxena *et al.*, 2008b) for our depth template learning phase. This dataset was trained using a Markov random model (Geman and Geman, 1984). We predict the 3D plane parameter for each patch using a linear predictor. To capture the depth information at the neighboring pixels, the predictor uses a concatenation of the features within each patch and features within four of its neighbors. Thus, the local depth estimation method estimates the absolute depth information between neighboring pixels. To estimate the relative depth between two given pixels, we consider the multi-resolution model shown in Fig. 9, and find a mapping of pixels among low-, medium-, and high-resolution images. We assume that the relative depth between two pixels is proportional to the difference between the respective relative feature vectors. Thus, the depth information from a single image is represented as a composite feature vector. We then compute the confidence in the prediction using linear logistic regression with the features described in the feature extraction section. Note that the depth predictions for smooth regions rely more on pair-wise constraints with their neighbors than on their own features. We model the posterior distribution of depth d , given the feature vectors \mathbf{F} parameterized by σ , which is the smoothness of the depth map, and θ , which is the orientation of the depth map, as given below:

$$P(d|\mathbf{F}) = \frac{1}{C} \exp(-G_{\sigma,\theta}(d, \mathbf{F})), \quad (6)$$

$$G_{\sigma,\theta}(d, \mathbf{F}) = \sum_{i=1}^N \frac{(d_i - \mathbf{F}_i \theta)^2}{\sigma^2} + \sum_{i=1}^N \sum_{j \in N_s(i)} \frac{(d_i - d_j)^2}{\sigma^2}, \quad (7)$$

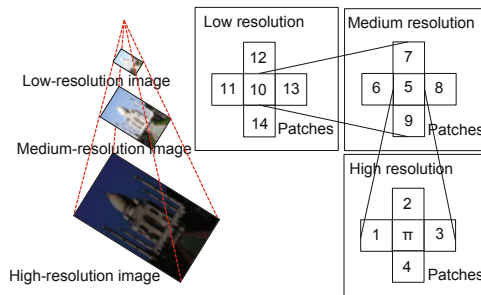


Fig. 9 Image pyramid depicting a multi-resolution model used to find the mapping of pixels among low-, medium-, and high-resolution images

where C is the normalization constant, $G_{\sigma,\theta}(d, \mathbf{F})$ is the Gibbs energy function, N is the number of patches, d_i is the depth at patch i , and \mathbf{F}_i is the feature vector at patch i . The first term of the Gibbs energy function in Eq. (7) gives the local depth at patch i , and the second term gives the global depth at patch i in terms of the depth at its four neighboring patches, $N_s(i)$.

3.6 Mesh generation

The goal of the mesh generation step is to connect or stitch together the patches $P = \{P_1, P_2, \dots, P_N\}$ resulting from the feature extraction module into a connected mesh structure, and approximate their outlines using coarse polygons. From the surface layout step, we extract their ordered boundary points, which act as the initial polygons. Since the extracted boundaries are generally noisy, we cannot assume that we have a good approximation of the vertex normal orientations. We therefore reconstruct a 3D vertex normal based on the initial values and neighborhood relationship derived from the multi-resolution mapping described earlier in the section on depth inference. This process straightens the initial boundaries and provides a coarse polygonal approximation of the 2D components. The 3D model generated from our approach consists of a front view of the building together with the data structure of the 3D information, including the positioning of the segments. These provide identifiable edges that enable easy correspondence between the generated 3D model and the 2D photograph. Finally, a structural hierarchical analysis of the components helps to simplify further the solution by providing the position of each component and the optimum symmetry line. All traced 3D structural information can be snapped together into a 3D model using any available modeling tool, and further adjustments can be made manually to finalize the 3D model.

Extruding editable objects from a single photograph was suggested by Chen *et al.* (2013). A limitation of this approach is that many shapes cannot be decomposed into generalized cylinders and cubes, and hence cannot be modeled. To convert a surface-based representation into a solid representation, we use an extrapolation technique (Çiçek and Gülesın, 2004; Nan *et al.*, 2010), i.e., extruding the components and converting them from a wire-frame into

a solid model. In our work, we apply an extrusion process only after the initial surface is reconstructed. The adjustment steps can be compared to any simple 2D vector graphics editing process. The edit mesh modifier in the software provides explicit editing tools for different sub-object levels of the selected object, i.e., the vertex, edge, or the whole polygon. The structural information on the Taj Mahal shows that its architecture follows a strict symmetry. Our approach takes strong advantage of the building's 3D symmetry (Bokeloh *et al.*, 2009) in assembling the 3D components. The important element to note here is that to fit together, each group of components needs to be scaled, as well as translated or moved, which is a simple one-step process for most 3D modelers. Manual adjustments can also be applied when a patch has no polygons assigned, or if the existing polygon is incorrectly estimated owing to the presence of noisy or missing vertices. Thus, our final model is generated automatically, and manual adjustments and/or editing can be conducted to enhance the model using the modeling software.

4 Results and discussion

We implemented the proposed approach and applied it to the reconstruction of various Mughal monuments including the Taj Mahal, Humayun's Tomb, and the Badshahi Mosque from single photographs. Table 2 shows a quantitative comparison of the time required for a reconstruction of the 3D models and their geometric anatomy for the two phases of our approach. We used an Intel Core i7-3770 CPU @ 3.40 GHz for the implementation. The implementation results are shown in Figs. 10 and 11. The test results for similar monuments of Mughal design style are shown in Fig. 12. We conducted a user study to verify our approach and its usability for both expert and non-expert users. We selected five candidate users for this study. Users A and C are

researchers in the area of real-time rendering. User B is a user with 3D modeling background. User D has basic computer graphics skills, and User E is a historian with no graphics skills. Before the start of the session, we gave only the general instruction to create a good looking model, since not all candidates had a computer graphics or modeling background. We stopped each session as soon as the user reached a closed model that resembled the reference model. The reference model for the Taj Mahal model was modeled by considering the multiple view images captured from a 360° 3D panorama of the Taj Mahal monument. About 30 images from different sides of the Taj Mahal were used. The mesh for the reference model was generated from point clouds using conventional modeling software. To quantify the modeling accuracy of our approach, we estimated the approximation error of the models created during the user study by our proposed approach with the reference model. This comparison was done using Hausdorff distances relative to the model's bounding box diagonal, measured using the Metro tool (Cignoni *et al.*, 1998). The results in Table 3 indicate that our method gives reasonable error rates when compared with the reference model. The results also show a very small range in the displacement error among the models generated by the different users who participated in our user study.

Performance analyses in terms of speed in recreating the 3D model in our user study were conducted. Each user was asked to participate in the experiment for three sessions. Session 1 was a 3D model refinement of the Taj Mahal using a 3D component library after obtaining the results of our approach for the initial automatic reconstruction phase. Session 2 was a manual modeling from scratch (using any available modeling software) using a single photograph of the Taj Mahal. In session 3, no real photographs were provided to the users, but instead only the overall structural and component descriptions they needed

Table 2 Quantitative comparison of the anatomy of the 3D models

Monument	Number of vertices		Number of triangles		Number of faces		Time (min)	
	Auto.	Manu.	Auto.	Manu.	Auto.	Manu.	Auto.	Manu.
Taj Mahal	2664	4256	4756	8090	1585	2696	0.80	3.50
Humayun's Tomb	961	1924	2796	3568	781	1810	0.75	2.00
Badshahi Mosque	1061	2094	2985	3848	989	1925	0.90	2.50

Auto.: automatic reconstruction phase; Manu.: manually refined reconstruction phase

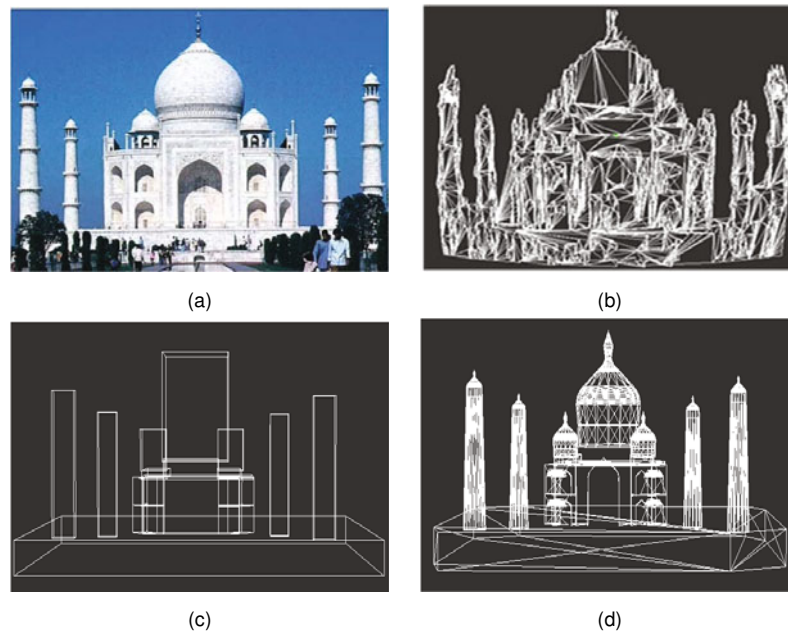


Fig. 10 Implementation results of the 3D reconstruction pipeline: (a) digital input image; (b) initial automatic reconstruction results; (c) bounding box showing the estimated symmetry; (d) wire-frame model of the final reconstruction



Fig. 11 Final reconstructed Taj Mahal 3D model in different views

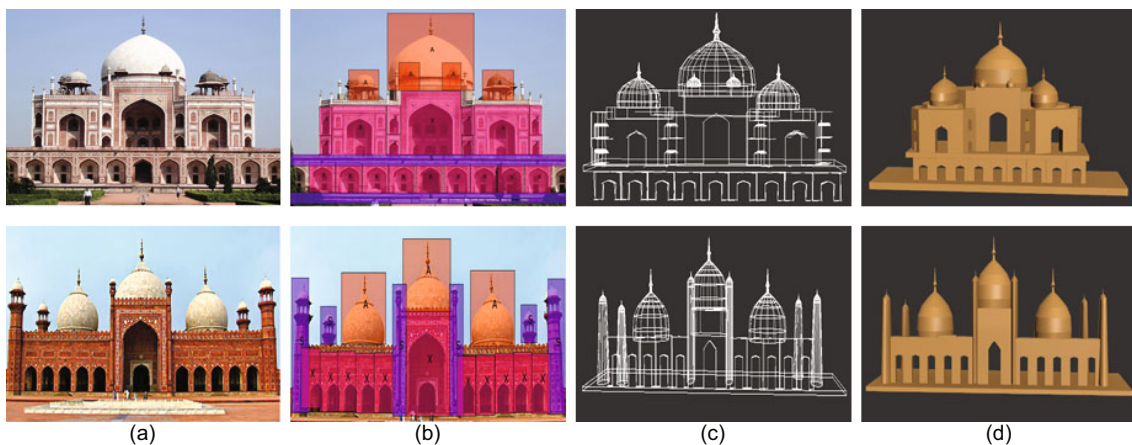


Fig. 12 Two typical examples of Mughal architecture with the first and second rows showing the reconstruction results of Humayun's Tomb and the Badshahi Mosque, respectively: (a) a single digital image; (b) semi-automatic component identification (where each randomly generated color represents one unique component and its repetition); (c) the wire-frame of the reconstructed model; (d) the final reconstructed model. References to color refer to the online version of this figure

to create the 3D model from their own imagination and creative ability were provided. The results are shown in Fig. 13. The speed of modeling was observed to be better with our approach (session 1), although this may vary based on the expertise of the users and their familiarity with the modeling environment.

Table 3 Comparison of the displacement errors measured using Hausdorff distances relative to the reference model

User	Mean (%)	Maximum (%)	RMS (%)
A	4.583654	5.566855	4.623318
B	4.578231	5.566855	4.618944
C	4.584550	5.566855	4.624117
D	4.586463	5.540374	4.623213
E	4.580921	5.540374	4.617344

RMS: root mean square

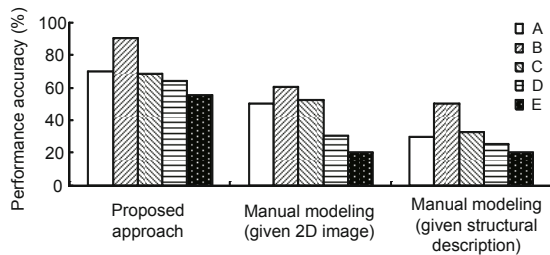


Fig. 13 Performance in terms of the accuracy of the modeling sessions of our user study. Each of the five bars represents a different user and the corresponding performance accuracy for the different sessions

Our main conclusion based on the user study is that participants without any prior computer graphics knowledge or modeling experience can create perfectly closed shapes of 3D building models that can match a given 2D photograph.

The 3D model created from our approach can be rendered as a proxy for digital effects used in real and imaginative virtual worlds. Fig. 14 shows a rendered scene of the Taj Mahal model created from our approach, which can be used in applications including games and entertainment. The 3D scene was rendered using commercial rendering software.

5 Conclusions and future work

We presented a faster and easier method for image-based 3D reconstruction that leverages techniques from modern digital technologies and image

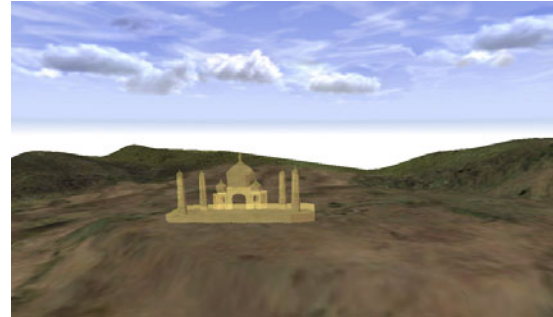


Fig. 14 Rendering of a 3D model created from our approach suitable for games and entertainment applications. The 3D model of the Taj Mahal was rendered using imagined terrain such as a large mountain range

processing algorithms. This approach is especially of value if just one image of an object and no other photographic information is available. The proposed system uses an initial automatic reconstruction from a single photograph by generating candidate planes and constructing coarse polygons upon these planes. A structural hierarchical analysis of the components in the image helps to reduce further the infinite number of solutions that might otherwise result from a 3D recovery of a 2D scene. The components generated to form a component library of a specific architectural style can be reused for reconstructing additional monuments of the same design style. To validate our approach, we conducted tests using a 2D image taken from the front view of three well-known monuments from the Mughal architecture style. The results of the reconstructed 3D models demonstrate the effectiveness of our proposed approach.

However, there are also limitations of our approach. The major limitation arising due to the use of a single image is incompleteness of the 3D object model because there is no information about the back of the object. Also, during our tests, with a wide range of different monument photographs, we observed that in some cases, the parts of the monument were only sparsely recreated. This was caused by occlusions (trees or humans) in the photograph. This limitation can be overcome by using a few more images instead of a single image. The second limitation of our approach is the need for additional object information. This information is usually available for the application area of architecture. In archaeology, however, our approach is not applicable. Another limitation is that this approach is able to work only for an image depicting the frontal view of a building. This in turn depends on the quality of the input

image and the unique characteristics of the identified components, like the symmetry and alignment of the components. The components reconstructed in our case study have an aligned symmetry axis. Non-axis aligned components will provide more plausible results for certain architectural styles. This will be considered in our future work.

In spite of these clear limitations, there are other potential applications of our 3D reconstruction approach. In the case of demolished historic buildings, for instance, sometimes no more than a single image is available. Furthermore, our approach reduces the number of images required to obtain a complete model. Like any other 3D model, the models reconstructed by our approach can also be explored and viewed at various scales in augmented reality (AR) based environments relating to virtual tourism and virtual museum exhibits. This type of AR environment can show the visualization relationship and interaction between the reconstructed 3D model and the real world objects. In future, our approach can be extended to urban environments to make it more useful for construction engineers to augment their design in real environments and explore interactions.

In future work, we would also consider applying our approach to reconstruct buildings from images taken from different side views. In addition, we would like to experiment with the components extracted from our approach to generate variations of ancient architectural building styles to provide users with an amalgam of different cultures and historical periods. It will also be possible to populate a virtual city with the same family of buildings. User content creation (UCC) is another potential application of our approach.

References

- AlHalawani, S., Yang, Y.L., Liu, H., *et al.*, 2013. Interactive facades analysis and synthesis of semi-regular facades. *Comput. Graph. Forum*, **32**(2pt2):215-224. [doi:10.1111/cgf.12041]
- Bao, F., Yan, D.M., Mitra, N.J., *et al.*, 2013. Generating and exploring good building layouts. *ACM Trans. Graph.*, **32**(4):122.1-122.10. [doi:10.1145/2461912.2461977]
- Bay, H., Tuytelaars, T., van Gool, L., 2006. SURF: speeded up robust features. Proc. 9th European Conf. on Computer Vision, p.404-417. [doi:10.1007/11744023_32]
- Bokeloh, M., Berner, A., Wand, M., *et al.*, 2009. Symmetry detection using feature lines. *Comput. Graph. Forum*, **28**(2):697-706. [doi:10.1111/j.1467-8659.2009.01410.x]
- Ceylan, D., Mitra, N.J., Li, H., *et al.*, 2012. Factored facade acquisition using symmetric line arrangements. *Comput. Graph. Forum*, **31**(2pt3):671-680. [doi:10.1111/j.1467-8659.2012.03046.x]
- Ceylan, D., Mitra, N.J., Zheng, Y., *et al.*, 2014. Coupled structure-from-motion and 3D symmetry detection for urban facades. *ACM Trans. Graph.*, **33**(1):2.1-2.15. [doi:10.1145/2517348]
- Chen, E., Williams, L., 1993. View interpolation for image synthesis. Proc. 20th Annual Conf. on Computer Graphics and Interactive Techniques, p.279-288. [doi:10.1145/166117.166153]
- Chen, T., Zhu, Z., Shamir, A., *et al.*, 2013. 3-Sweep: extruding editable objects from a single photo. *ACM Trans. Graph.*, **32**(6):195.1-195.10. [doi:10.1145/2508363.2508378]
- Çiçek, A., Gülesin, M., 2004. Reconstruction of 3D models from 2D orthographic views using solid extrusion and revolution. *J. Mater. Process. Technol.*, **152**(3):291-298. [doi:10.1016/j.jmatprotec.2004.04.368]
- Cignoni, P., Rocchini, C., Scopigno, R., 1998. Metro: measuring error on simplified surfaces. *Comput. Graph. Forum*, **17**(2):167-174. [doi:10.1111/1467-8659.00236]
- Criminisi, A., Reid, I., Zisserman, A., 2000. Single view metrology. *Int. J. Comput. Vis.*, **40**(2):123-148. [doi:10.1023/A:1026598000963]
- Davies, E.R., 2005. Machine Vision: Theory, Algorithms, Practicalities. Morgan Kaufman Press, San Francisco, USA.
- Debevec, P.E., Taylor, C.J., Malik, J., 1996. Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach. Proc. 23rd Annual Conf. on Computer Graphics and Interactive Techniques, p.11-20. [doi:10.1145/237170.237191]
- Dung, L.R., Huang, C.M., Wu, Y.Y., 2013. Implementation of RANSAC algorithm for feature-based image registration. *J. Comput. Commun.*, **1**:46-50. [doi:10.4236/jcc.2013.16009]
- Encyclopedia, 2014. Mughal Architecture, Britannica Online. Available from <http://global.britannica.com/EBchecked/topic/396119/Mughal-architecture> [Accessed on Dec. 11, 2014].
- Faugeras, O., Laveau, S., Robert, L., 1995. 3-D reconstruction of urban scenes from sequences of images. Automatic Extraction of Man-Made Objects from Aerial and Space Images, p.145-168. [doi:10.1007/978-3-0348-9242-1_15]
- Felzenszwalb, P.F., Huttenlochet, D.P., 2004. Efficient graph-based image segmentation. *Int. J. Comput. Vis.*, **59**(2):167-181. [doi:10.1023/B:VISI.0000022288.19776.77]
- Frahm, J.M., Fite-Georgel, P., Gallup, D., *et al.*, 2010. Building Rome on a cloudless day. Proc. 11th European Conf. on Computer Vision, p.368-381. [doi:10.1007/978-3-642-15561-1_27]
- Garcia-Gago, J., Gomez-Lahoz, J., Rodríguez-Méndez, J., *et al.*, 2014. Historical single image-based modeling: the case of Gobierna Tower, Zamora (Spain). *Remote Sens.*, **6**(2):1085-1101. [doi:10.3390/rs6021085]
- Geman, S., Geman, D., 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Patt. Anal. Mach. Intell.*, **6**(6):721-741. [doi:10.1109/TPAMI.1984.4767596]

- Gormen, T.H., Leiserson, C.E., Rivest, R.L., *et al.*, 1990. Introduction to Algorithms. MIT Press, McGraw-Hill Book Company, New York, USA.
- Guillou, E., Meneveaux, D., Maisel, E., *et al.*, 2000. Using vanishing points for camera calibration and coarse 3D reconstruction from a single image. *Vis. Comput.*, **16**(7):396-410. [doi:10.1007/PL00013394]
- Hoiem, D., Efros, A.A., Hebert, M., 2005. Geometric context from a single image. Proc. 10th IEEE Int. Conf. on Computer Vision, p.654-661. [doi:10.1109/ICCV.2005.107]
- Horn, B.K.P., 1990. Height and gradient from shading. *Int. J. Comput. Vis.*, **5**(1):37-75. [doi:10.1007/BF00056771]
- Horry, Y., Anjyo, K., Arai, K., 1997. Tour into the picture: using a spidery mesh interface to make animation from a single image. Proc. 24th Annual Conf. on Computer Graphics and Interactive Techniques, p.225-232. [doi:10.1145/258734.258854]
- Kang, S., 1998. Depth Painting for Image-Based Rendering Applications. Technical Report, Compaq Computer Corporation, Cambridge Research Lab.
- Laveau, S., Faugeras, O., 1994. 3D scene representation as a collection of images. Proc. 12th Int. Conf. on Pattern Recognition, p.689-691. [doi:10.1109/ICPR.1994.576404]
- Liebowitz, D., Criminisi, A., Zisserman, A., 1999. Creating architectural models from images. *Comput. Graph. Forum*, **18**(3):39-50. [doi:10.1111/1467-8659.00326]
- Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.*, **60**(2):91-110. [doi:10.1023/B:VISI.0000029664.99615.94]
- Ma, J., Chan, J.C., Canters, F., 2010. Fully automatic subpixel image registration of multiangle CHRIS/Proba data. *IEEE Trans. GeoSci. Remote Sens.*, **48**(7):2829-2839. [doi:10.1109/TGRS.2010.2042813]
- Manferdini, A.M., 2012. A methodology for the promotion of cultural heritage sites through the use of low-cost technologies and procedures. Proc. 17th Int. Conf. on 3D Web Technology, p.180. [doi:10.1145/2338714.2338747]
- McMillan, L., Bishop, G., 1995. Plenoptic modeling: an image-based rendering system. Proc. 22nd Annual Conf. on Computer Graphics and Interactive Techniques, p.39-46. [doi:10.1145/218380.218398]
- Mitra, N.J., Pauly, M., 2008. Symmetry for architectural design. *Advances in Architectural Geometry*, p.13-16.
- Mitra, N.J., Pauly, M., Wand, M., *et al.*, 2013. Symmetry in 3D geometry: extraction and applications. *Comput. Graph. Forum*, **32**(6):1-23. [doi:10.1111/cgf.12010]
- Müller, P., Zeng, G., Wonka, P., *et al.*, 2007. Image-based procedural modeling of facades. *ACM Trans. Graph.*, **26**(3):85.1-85.9. [doi:10.1145/1276377.1276484]
- Nagai, T., Ikehara, M., Kurematsu, A., 2007. HMM-based surface reconstruction from single images. *Syst. Comput. Jpn.*, **38**(11):80-89. [doi:10.1002/scj.10685]
- Nan, L., Sharf, A., Zhang, H., *et al.*, 2010. SmartBoxes for interactive urban reconstruction. *ACM Trans. Graph.*, **29**(4):93.1-93.10. [doi:10.1145/1778765.1778830]
- Nevatia, R., Babu, K.R., 1980. Linear feature extraction and description. *Comput. Graph. Image Process.*, **13**(3):257-269. [doi:10.1016/0146-664X(80)90049-0]
- Oh, B.M., Chen, M., Dorsey, J., *et al.*, 2001. Image-based modeling and photo editing. Proc. 28th Annual Conf. on Computer Graphics and Interactive Techniques, p.433-442. [doi:10.1145/383259.383310]
- Poulin, P., Ouimet, M., Frasson, M.C., 1998. Interactively modeling with photogrammetry. Proc. Eurographics Workshop on Rendering, p.93-104. [doi:10.1007/978-3-7091-6453-2_9]
- Pylvanainen, T., Berclaz, J., Korah, T., *et al.*, 2012. 3D city modeling from street-level data for augmented reality applications. Proc. 2nd Int. Conf. on 3D Imaging, Modeling, Processing, Visualization and Transmission, p.238-245. [doi:10.1109/3DIMPVT.2012.19]
- Remondino, F., 2011. Heritage recording and 3D modeling with photogrammetry and 3D scanning. *Remote Sens.*, **3**(6):1104-1138. [doi:10.3390/rs3061104]
- Saxena, A., Chung, S.H., Ng, A.Y., 2008a. 3-D depth reconstruction from a single still image. *Int. J. Comput. Vis.*, **76**(1):53-69. [doi:10.1007/s11263-007-0071-y]
- Saxena, A., Sun, M., Ng, A.Y., 2008b. Make3D: depth perception from a single still image. Proc. 23rd AAAI Conf. on Artificial Intelligence, p.1571-1576.
- Shade, J., Gortler, S., He, L., *et al.*, 1998. Layered depth images. Proc. 25th AAAI Annual Conf. on Computer Graphics and Interactive Techniques, p.231-242. [doi:10.1145/280814.280882]
- Shen, C.H., Fu, H., Chen, K., *et al.*, 2012. Structure recovery by part assembly. *ACM Trans. Graph.*, **31**(6):180.1-180.11. [doi:10.1145/2366145.2366199]
- Styliadis, A.D., Sechidis, L.A., 2011. Photography-based facade recovery & 3D modeling: a CAD application in cultural heritage. *J. Cult. Herit.*, **12**(3):243-252. [doi:10.1016/j.culher.2010.12.008]
- Super, B.J., Bovik, A.C., 1995. Shape from texture using local spectral moments. *IEEE Trans. Patt. Anal. Mach. Intell.*, **17**(4):333-343. [doi:10.1109/34.385983]
- Wang, Y., Olano, M., 2011. A framework for GPU 3D model reconstruction using structure-from-motion. Proc. 38th Annual Conf. on Computer Graphics and Interactive Techniques, p.27.1. [doi:10.1145/2037715.2037748]
- Wei, Y.M., Kang, L., Yang, B., *et al.*, 2013. Applications of structure from motion: a survey. *J. Zhejiang Univ.-Sci. C (Comput. & Electron.)*, **14**(7):486-494. [doi:10.1631/jzus.CIDE1302]
- Yang, M.D., Chao, C.F., Huang, K.S., *et al.*, 2013. Image-based 3D scene reconstruction and exploration in augmented reality. *Autom. Constr.*, **33**:48-60. [doi:10.1016/j.autcon.2012.09.017]
- Zhang, H., Xu, K., Jiang, W., *et al.*, 2013. Layered analysis of irregular facades via symmetry maximization. *ACM Trans. Graph.*, **32**(4):121.1-121.10. [doi:10.1145/2461912.2461923]
- Zhang, L., Dugas-Phocion, G., Samson, J.S., *et al.*, 2002. Single-view modeling of free-form scenes. *J. Visual. Comput. Animat.*, **13**(4):225-235. [doi:10.1002/vis.291]