Multilevel modelling and rendering of architectural scenes

Akash M Kushal

Gaurav Chanda

Kanishka Srivastava

Mohit Gupta

Subhajit Sanyal

T. V. N. Sriram

Prem Kalra

Subhashis Banerjee

Department of Computer Science and Engineering,
IIT Delhi, New Delhi 110016, India
email: {pkalra,suban}@cse.iitd.ernet.in

Abstract

We present a novel approach for multilevel modelling and rendering of architectural scenes using a small set of photographs. Our approach is based on interactive probing of intuitive measures like lengths, widths and heights from single view metrology. These measures can then be aggregated with additional input if need be, for defining high-level primitives such as planes, prismatic blocks, cuboids, spheres, and general surfaces of translation and revolution. The modelling approach is modular and incremental which enables ease and flexibility in building large architectural scenes. Further, our approach allows multilevel modelling and rendering. That is, the model can be enhanced/reduced both in terms of changing details at the primitive level and at the model level. This requires inter-registration of two or more images with common contents. Our approach is computationally simple, fast and robust. We demonstrate our results by building a model of a magnificent historical monument in Delhi - Humayun's tomb.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling

1. Introduction

Recently the problem of interactive 3D modelling of architectures from a sparse set of photographs has attracted a lot of attention. First Debevec et al.[2] and then others[9, 8, 4] have shown that such image based methods can provide high quality reconstruction and photo-realistic rendering without going through tedious and cumbersome CAD or digitizing systems.

In this paper we describe a system for interactive modelling and rendering of architectural scenes from one or more photographs with multiple levels of details. We present an incremental technique that is based on probing of lengths and widths along parallel directions. Our method can also deal with spheres, cones and frustums, surfaces of translation and surfaces of revolution.

Most methods for recovery of 3D structure from images rely on calibrated stereo or uncalibrated images using epipolar or trilinear constraints [5]. In contrast, methods based on single-view analysis [2, 9, 8, 4] exploit the structural constraints available in architectural photographs to directly estimate the location and the intrinsic parameters of the camera and reconstruct 3D models through user interactions from one or more images. 3D reconstruction from a single image must necessarily be through an interactive process, where the user provides information and constraints about the scene structure. Such information may be in terms of vanishing points or lines [6, 1, 8, 9], co-planarity[11], spatial inter-relationship of features[2, 4] and camera constraints [12].

The Facade system[2], one of the most successful for modelling and rendering of architectural scenes, consists of a hybrid image and geometry based approach which requires decomposing the scene into prismatic blocks and cuboids which are the basic primitives for modelling. Each primitive is defined in terms of six parameters describing its location and a small number of other parameters describing its internal details. The user needs to interactively define constraints on placements of the primitive blocks and mark corresponding edge segments in the image. In general a non-linear optimization method is required to solve for all the model and camera parameters (external) simultaneously. The internal parameters of the camera need to be calibrated apriori. In some special cases, especially if the blocks are aligned with respect to each other, or their rotation with respect to each other are known, linear methods can be used. While model fitting using simultaneous bundle adjustment[5] appears to be an attractive approach, it may turn out to be problematic if the model requires a large number of primitive blocks oriented with respect to each other at unknown angles. In such a case the large number of unknown parameters in simultaneous bundle adjustment using non-linear optimization may make the method unsuitable. In addition, prismatic blocks and cuboids may not be the most convenient primitives to use for modelling in all situations (see for example Figure 1).

Other significant approaches to architecture modelling are based on estimation of vanishing lines in images [9] and metric rectification of planes[8]. In the *PhotoBuilder* system [9, 10] a user needs to mark parallel and orthogonal edges in at least two images. These are used to compute the vanishing points in the images corresponding to the three orthogonal coordinate axes in the world. In addition, the user needs to mark the projection of a cuboid of known size in the images. The vanishing points and the images of the cuboid are sufficient to both localize the cameras in a common world frame and to estimate their intrinsic parameters independently. 3D reconstruction of points in the world is achieved by stereo triangulation from image correspondences. Since the basic method of 3D reconstruction and metrology is based on stereo triangulation, at least two images are required. Leibowitz. et al.[8] propose a method for metric rectification of planes and camera estimation based on computation of circular points in images from vanishing directions. Both these methods are limited to blocks world reconstruction of the scene.

We propose a method based on single view metrology [1, 7] for incremental model building. Our method is based on simple interactive probing of length ratios along directions parallel to the world coordinate frame. It is not necessary that the three coordinate axes of the world frame are orthogonal and be specified by the user in the same scale. Often, accurate length measurements along the three axes directions in the world are not available, and, in such a case, since our probing technique is based on computing affine ratios along parallel directions in the world, we can still carry out an affine reconstruction of the world model. In case it is known that the three axes directions in the world are orthogonal, the affine reconstruction can be updated to Euclidean using rough knowledge of the internal parameters of the camera or experimenting with the scale factors of the coordinate axes to obtain visual alignment. In case the planes are orthogonal and the scales are known, we can recover the camera parameters and the model directly in Euclidean terms.

Our primitives are more atomic than the prismatic blocks and cuboids used in *Facade*[2]. At the same time the basic technique can be used to simultaneously compute parts or whole of the model in terms of higher level primitives like planes, sequence of planes, prismatic blocks, cuboids, spheres, cones and frustums, surfaces of translation and surfaces of revolution using simultaneous bundle adjustment similar to that in *Facade*[2]. Consequently, a user has more flexibility to experiment and evolve a right strategy on a case to case basis and decide on whether to use the basic probing technique or to solve for several primitive parameters simultaneously in a tightly constrained manner. We determine both the camera internal parameters and its location apriori (unlike in *Facade*[2]), and hence all subsequent modelling steps are linear and computationally efficient. Further, modelling errors in a portion of the scene can be corrected without affecting the modelling of other parts.

The basic probing technique gives us added flexibility for incremental model building at multiple levels of detail. Since the atomic primitives are just length ratios and not more complex blocks, our method is more suitable for multilevel analysis and modelling. For example, a planar face modelled at a coarse resolution can be further sub-divided into a set of primitives whose model parameters may be independently computed from metrology using a finer resolution image. These new primitives can be planes, blocks and even surfaces. The user needs to register the commonly visible parts in the set of images and replace coarser details with finer resolution geometry and texture. The registration requires only linear computations.

In the following sections we present the details of our method. In Section 2 we give an overview of our system. In Section 3 we present our basic method for single view metrology and describe how the basic probing technique can be used either incrementally or in aggregation. In Section 4 we extend the basic method to deal with curved objects. In Section 5 we describe our strategy and provide results for modelling and rendering of Humayun's tomb[†], and finally in Section 6 we conclude the paper.

2. Overview

In what follows we give a brief overview of our system. The method involves the following main steps:

Camera estimation: We first register two planes independently by identifying a rectangle (or parallelogram) on each in a coarse resolution photograph. The planes need not be orthogonal, though we refer to them as *vertical* and *horizontal* for convenience. See Figure 1(a) (also Figure 6). This sets up an affine coordinate system in the world and enables the recovery of corresponding camera parameters[7]. We obtain the vanishing points of the axes directions in the images directly as a by-product. In case the planes are orthogonal and the size of each rectangle is known accurately, then we can recover the camera in Euclidean terms and compute the internal parameters of the camera.

Probing heights and widths: Once the camera projection matrix is known we can compute the height (width) of any arbitrary 3D point by identifying the point (head) and its projection (foot) on the horizontal (vertical) plane in the image along the coordinate axis direction (See Figure 1(a)). In case the foot of the point is not visible on one of the reference plane but is clearly identifiable in a plane parallel to the reference plane, then the offset of the parallel plane from the reference plane must be computed apriori using a similar technique.

Modelling faces and blocks: The basic probing technique can be used directly to compute sizes of faces and blocks. Alternatively, several faces and blocks can be simultaneously modelled in a tightly constrained manner. For example, in the photograph of Figure 1(b), the user may choose to enforce the constraints that i) the entire front of the main monument consists of a sequence of planar faces adjacent to each other ii) several of them are parallel to each other iii) all the planar faces are orthogonal to the plane of the courtyard, hence the line joining the head and foot, sampled anywhere, must pass through the vanishing point of the Y direction iv) the top and bottom edges of several of the faces must pass through the vanishing point of the X direction v) the common height of the frontal faces can be estimated by considering a pencil of rays originating from the Y direction vanishing point and computing their intersections with the line through the top and bottom edges of the faces to mark of the *head*'s and *foot*'s. All the above constraints can be expressed as a single linear system which can be solved using *SVD*. Note that the entire model built as above is dependent on the height of the courtyard, and this dependence can even be expressed symbolically. This height can either be supplied, or computed separately by independent probing, or even be computed simultaneously using additional constraints. Thus, the user has the flexibility of modelling parts of the scene either incrementally or as an aggregate.

Modelling surfaces: The knowledge of the camera projection matrix and the axis is sufficient to compute an arbitrary surface of revolution from its silhouette edges. For the various surfaces of revolutions in the photograph of Figure 2(a) - the axis directions are known to be vertical, the *head* is directly visible in the image, and the foot positions are known from symmetry. In Section 4 we give the details of modelling arbitrary surfaces of revolutions.

Registering multiple levels of details: For faithful rendering of walk-throughs at close-ups we need to sub-divide a primitive modelled at a coarser resolution into finer primitives. For example, from the photograph in Figure 2(a) the big vertical face of the entrance to the main monument is modelled as a plane. However, at close-up, as in Figure 2(b), we need to carry out an independent metrology to model the inside in terms of several planes and surfaces of translation. For this we set up an independent coordinate system as depicted in Figure 2(b). The user needs to register the two independent metrologies by identifying correspondences as shown in Figure 2. The model with the internal detail of the face shown in wire-frames is depicted in Figure 11.

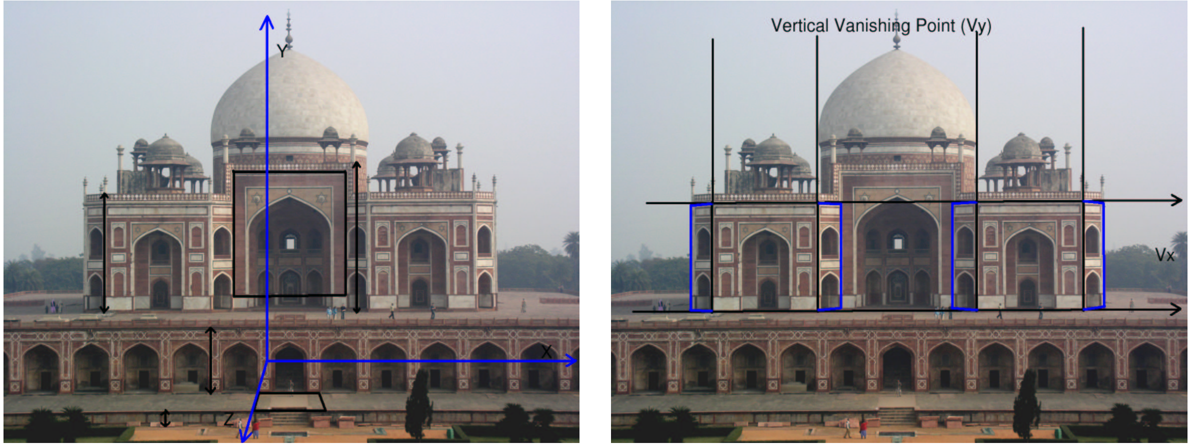
Rendering: For subsequent rendering of walk-throughs we employ a technique similar to view dependent texture mapping [2, 3]. The model is projected onto the original image(s) from which it is built. This establishes a correspondence of the model parameters with the images and provides the texture coordinates of the associated primitives. Often a single photograph will cover only a part of the model. This requires us to use, in most situations, multiple images at varying resolutions to be able to render the entire model. Compositing with α-channel blending is employed to decide the final pixel value. During the enhancement of the model (changing from one level of detail to another; see Figure 2) we do stenciling to separate the portions and pick appropriate textures.

In what follows we describe our basic probing technique.

3. Metrology

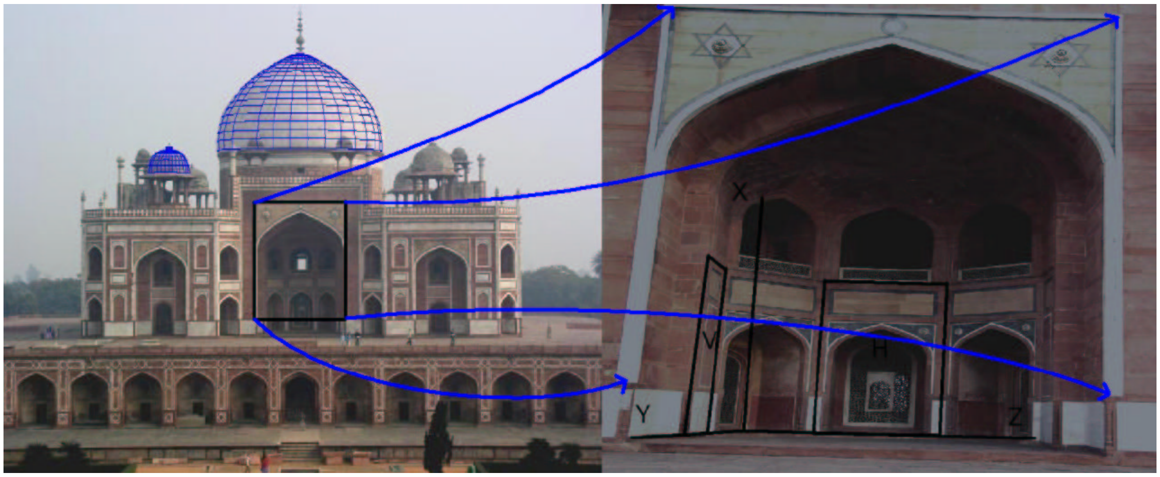
In a recent paper Criminisi et al.[1] show that under the customary assumption of the pin-hole camera model[5], the minimal geometric information that a user needs to provide for computation of 3D measurements is the affine registration of i) a reference plane and ii) a reference direction away from the plane. Once the vanishing line of the reference plane and the vanishing point of the reference direction are established, the user can probe for length ratios on planes parallel to the reference plane and lines parallel to the reference direction and thereby recover 3D affine structure.

[†] http://depts.washington.edu/uwch/silkroad/cities/india/delhi/humayun/humayun.html



(a) (b)

Figure 1: *(a) Camera determination and height measurements: The user marks a rectangle each on a vertical and a horizontal plane from which the world coordinate frame and the camera is recovered.* Head's *and* foot's *can then be clicked to compute heights. (b) Face Lifting: The user may specify constraints for modelling of faces.*



(a) (b)

Figure 2: *(a) Domes modelled as surfaces of revolution. (b) Independent metrology for multiple levels of details and tie-points.*

In what follows we present a simple and practical method[7] for implementation of the basic ideas of Criminisi et al.[1] for interactive 3D reconstruction from single images and also consider some extensions. The method is based on computation of homographies[5] from two world planes to the image, which directly yield the necessary vanishing points and lines reliably. The user needs to identify the two planes by clicking four points on each (See Figures 1(a) and 6) so that the two planes can be independently calibrated in affine terms.

We show that the interactive registration of two planes facilitate 3D affine reconstruction of structured scenes using simple similar triangles and that it is possible to recover the camera position in the given coordinate system. The extra degrees of freedom obtained as a result of independent registration of the two planes give us constraints on (i) translation of the planes along the coordinate axes, and (ii) rotation of the coordinate system of one plane with respect to the other[7]. In addition, if the two planes are registered in a Euclidean coordinate system, we provide methods to determine the orientation of an arbitrary plane.

3.1. The basic case for camera estimation

Consider first the situation of Figure 3 where the two world planes and their common axis of intersection is visible in the image. Though the two planes (and the XYZ coordinate system) need not be orthogonal, we will refer to them as *vertical* and *horizontal* for convenience. Let the world coordinate system of the X-Y (vertical) and X-Z (horizontal) planes be as indicated in the figure. It is not necessary that three axes are of same scale. Let $\mathbf{H}$ and $\mathbf{G}$ be the homographies from the world X-Z and X-Y planes to the image respectively. These homographies may be computed by hand-picking the six defining points in the image in a standard way [5].

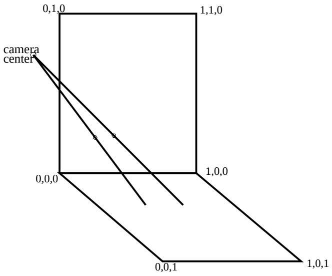


Figure 3: *The basic case.*

The first and the last column of both the homographies are equal up to scale since they represent in the image the vanishing point corresponding to the X direction and the projection of the world origin respectively. If we fix the relative scale of the homographies, by making the last columns equal, then since the unit length in the X direction for both the homographies is the same, the image of the (1,0) point for both the homographies will be the same (i.e. the image of the (1,0,1) point in the world). Hence the first columns would also have to be equal (including scale being equal). The second columns of $\mathbf{H}$ and $\mathbf{G}$ represent the vanishing points corresponding to the world Z and Y directions respectively.

So to obtain consistent homographies $\mathbf{H}$ and $\mathbf{G}$ we can solve for both the homographies together and decrease the number of independent variables from 16 (8 for each of the homographies) to just 11 (8 for any one and only 3 for the other since only the 2^{nd} column of the second is independent). These 11 degrees of freedom are exactly those required to define the full projection matrix (3×4) for the camera[5]. Registration of the two planes through $\mathbf{H}$ and $\mathbf{G}$ is also the minimal information required to set up the vanishing line of the reference plane (horizontal) and a vanishing direction (vertical) as required by Criminisi et al.[1].

As a consequence of the above, the 3×4 projection matrix of the camera can be read out from the columns of the homographies $\mathbf{H} = [\mathbf{H}(1), \mathbf{H}(2), \mathbf{H}(3)]$ and $\mathbf{G} = [\mathbf{G}(1), \mathbf{G}(2), \mathbf{G}(3)]$[7]. The projection matrix for the camera is $\mathbf{P} = [\mathbf{H}(1) \quad \mathbf{G}(2) \quad \mathbf{H}(2) \quad \mathbf{H}(3)]$.

3.2. Localization of camera center

The camera center can be directly computed from the homographies $\mathbf{H}$ and $\mathbf{G}$ defined above in the following way.

Clearly, $\mathbf{T} = \mathbf{G}^{-1}\mathbf{H}$ is the homography which maps from the X-Z plane to the X-Y plane through the camera center as shown Figure 3. Let the coordinates of the camera center in the world affine frame be (C_x, C_y, C_z). It can easily be verified geometrically that the homography that maps points on the X-Z plane to the X-Y plane is

$$\mathbf{T} = \mathbf{G}^{-1}\mathbf{H} = \lambda \begin{pmatrix} -C_z & C_x & 0 \\ 0 & C_y & 0 \\ 0 & 1 & -C_z \end{pmatrix} \qquad (1)$$

So we can now obtain the coordinates of the camera center. First we normalize the homography $\mathbf{T}$ by making $T_{32} = 1$. Then the camera center is $\mathbf{C} = (C_x, C_y, C_z) = (T_{12}, T_{22}, -T_{11})$. The camera center's coordinates (C_x, C_y, C_z) will be obtained in the world coordinate system. Here, the scales of X, Y and Z need not be identical.

3.3. Euclidean calibration of camera

If we want to calibrate the camera in Euclidean terms (i.e. determine the matrix of camera internals, $\mathbf{K}$)[5] we require to know i) the relative orientation of the XYZ coordinate axes and ii) the relative scales of each of the axes of the world coordinate system that we had initially specified. If the relative orientation is known then the camera matrix $\mathbf{P}$ can be re-computed to correspond to an orthogonal coordinate system in the world. To get the projection matrix in a coordinate system with unit lengths of the three axes equal we can just multiply $\mathbf{P}$ with a diagonal matrix as follows $\mathbf{P_{new}} = \mathbf{P_{old}} * diag(\alpha, \beta, \gamma, 1)$, where $1/\alpha$ is the unit length in the X-direction, $1/\beta$ is the unit length in the Y-direction and $1/\gamma$ is the unit length in the Z direction.

The projection matrix $\mathbf{P}$ obtained above can be written as $\mathbf{K}[\mathbf{R}|\mathbf{t}]$ where the 3×3 matrix $\mathbf{K}$ is the matrix of camera internals, $\mathbf{R}$ is the 3×3 rotation matrix which align the world coordinate system to the camera coordinate system and $-\mathbf{R}^t\mathbf{t}$ are the coordinates of the camera center [5]. Thus, the first 3×3 of $\mathbf{P}$ is $\mathbf{KR}$ (call it $\tilde{\mathbf{P}}$). Clearly $\tilde{\mathbf{P}} * \tilde{\mathbf{P}}^t = \mathbf{KK}^t$. Since we know $\mathbf{P}$, $\mathbf{K}$ can be obtained in a standard way[5].

Once $\mathbf{K}$ has been obtained, $\mathbf{R}$ and $\mathbf{t}$ can also be simply obtained. $\mathbf{R} = \mathbf{K}^{-\mathbf{1}} * \tilde{\mathbf{P}}$ where $\tilde{\mathbf{P}}$ is $\mathbf{KR}$ (the first 3×3 of $\mathbf{P}$). $\mathbf{t} = \mathbf{K}^{-1} * \mathbf{Kt}$ where $\mathbf{Kt}$ is the last column of $\mathbf{P}$. The camera center can then be computed as $-\mathbf{R}^t\mathbf{t}$.

3.4. Probing for heights and widths

Once the camera center coordinates have been determined, in either affine or Euclidean terms, the coordinates of 3D points in the scene can be computed using simple geometry. The user clicks a point (call it the *head*) and its projection on the X-Z plane (call it the *foot*) in the image along the $X - Y$ direction, for example the head and the foot of a person (see Figure 4). Now using the homography $\mathbf{H}^{-\mathbf{1}}$ (from the image plane to the X-Z horizontal plane) we can transfer the head and the foot of the point on to the horizontal (X-Z plane). We can then use simple similar triangles to calculate the height (Y coordinate) of the point.

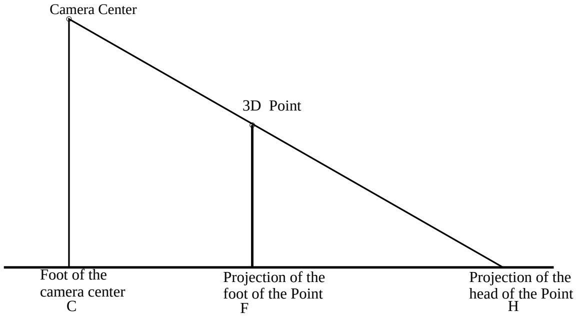

Camera Center
3D Point
Foot of the camera center C
Projection of the foot of the Point F
Projection of the head of the Point H

Figure 4: *Determining the height.*

The height can be computed as h = Camera height * $\frac{FH}{CH}$

Note that the height can also be obtained relative to the height of another object (with its head and foot given) by eliminating the camera height from the two similar triangle equations. The X and Z coordinates come out directly using the homography $\mathbf{H}^{-1}$. Thus we can get all the 3 affine coordinates of points in the scene coordinate system by simple geometry. To ensure that the head and foot are clicked along the $X-Y$ direction in the image, we enforce the straight line through the head and foot to pass through the vanishing point in the Y direction. Similarly, if we know that several points in 3D along a known straight line are of the same height, we enforce that the lines joining their 'heads' and 'foots' must pass through the vanishing point of the direction and solve for all the heights simultaneously from the image data.

Of course, the above procedure can also be applied relative to the X-Y plane to compute the width of objects as well.

Also, in case the foot of the object (on the horizontal plane) cannot be seen clearly in the image but is visible on a plane parallel to the horizontal plane (for example, when the object is placed on the horizontal courtyard of Figure 1), then we can first find out the height h of the courtyard by clicking the head and foot of any point on the courtyard and obtain the homography from the image to the horizontal (courtyard) plane $\mathbf{H} = [\mathbf{P}(1) \quad \mathbf{P}(3) \quad h\mathbf{P}(2) + \mathbf{P}(4)]$. Then, to compute a height relative to the courtyard, we can use a similar approach as above by taking the projection of the head on the courtyard plane and clicking the foot on the courtyard plane. Then, we can use similar triangles as before to calculate the height. In this case the height of the camera center will obviously be $C_y - h$ (offset computed from the courtyard).

3.5. Head/Foot click correction

The actual degrees of freedom of the *head-foot* clicks provided by the user is only 3 since the vanishing point in the Y direction is constrained to lie on the line connecting the head and the foot. We follow a technique similar to Criminisi et al.[1], wherein a user specifies uncertainty areas in the image corresponding to the head and the foot, and the best head and foot estimates are calculated by minimizing the sum of the *Mahalanobis distances* between the estimates and the uncertainty areas specified by the user.

3.6. Computing the orientation of an arbitrary plane

Angles have their standard meaning only in a Euclidean coordinate frame. Hence, for carrying out angle measurements we first need to correct the projection matrix $\mathbf{P}$ so that the unit lengths along the 3 axes are equal. Once $\mathbf{P}$ is available in Euclidean terms, we can carry out angle measurements as discussed below.

3.6.1. Angle of a vertical plane with the XY plane

If the intersection of the vertical plane with a known horizontal plane (whose height is known or can be determined) is visible in the image, the homography from the image to the horizontal plane can be calculated using $\mathbf{P}$. Then the angle can be found on the horizontal plane after correcting its image with the homography. Even if the intersection of the veridical plane with some known horizontal plane is not visible, one can calculate the angle (say ϕ) by just marking any horizontal line on the vertical plane. Let the projective representation of this line in the image be ℓ. The image of the point at infinity on this line is $\mathbf{P}(1)\cos\phi - \mathbf{P}(3)\sin\phi$, which yields $\tan\phi = \frac{\ell^T\mathbf{P}(1)}{\ell^T\mathbf{P}(3)}$.

3.6.2. Angle of a tilted plane from the ground plane

This angle (θ, see Figure 5) can be calculated if the user can specify a length ratio of two arbitrary vectors on the tilted plane, or alternatively, if the user can specify some length on the tilted plane in the world coordinate system. In the general scheme the user needs to first specify the intersection line of the tilted plane with the ground plane or any other plane at a known height as in the previous case. Then, we calculate the homography from the tilted plane to the image as

$$\mathbf{H}_\theta = \mathbf{P} * \begin{pmatrix} l_X & l_Z\cos\theta & X \\ 0 & \sin\theta & Y \\ l_Z & -l_X\cos\theta & Z \\ 0 & 0 & 1 \end{pmatrix} \tag{2}$$

where $(l_X, 0, l_Z)$ is the unit length direction vector of the intersection of the tilted plane with a horizontal plane and $\mathbf{X}_0$ = (X,Y,Z) is a point on the intersection axis. The coordinate system on the tilted plane has its X-axis along the intersection line with origin at $\mathbf{X}_0$.

The inverse of this homography ($\mathbf{H}_\theta{}^{-1}$) will take points on the image to the coordinate system on the tilted plane. Now, the user may mark two arbitrary vectors on the image of the tilted plane. These are projected to the tilted plane using $\mathbf{H}_\theta{}^{-1}$ and their length ratio provides the necessary constraint to calculate θ. Also, if the user can specify the length of any single vector on the tilted plane this length can be used directly to obtain θ by similarly projecting that vector to the tilted plane using $\mathbf{H}_\theta{}^{-1}$ and equating the resulting length to the one provided by the user.

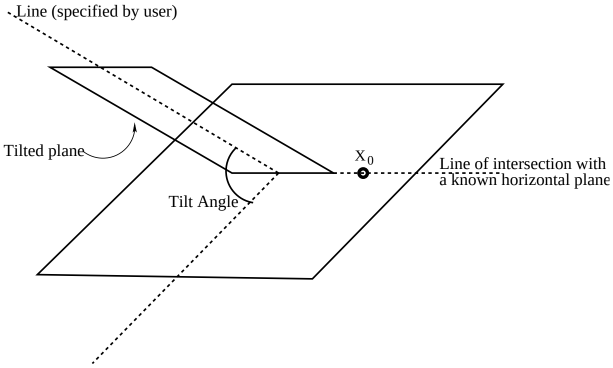


Figure 5: *Angle measurement*

Using an approach similar to that for vertical planes, the knowledge of a special line on the tilted plane in the image is also enough to determine the tilt angle as shown in Figure 5. If the user can specify the line on the tilted plane which is perpendicular to the intersection axis then one can use this line to calculate the tilt angle θ. Let the unit vector in the intersection line direction be $(l_X, 0, l_Z)$. Now, the image of the point at infinity in the direction as a function of θ is $-\mathbf{P}(1) l_Z \cos\theta + \mathbf{P}(3) l_X \cos\theta + \mathbf{P}(2)\sin\theta$ which yields $\tan\theta = \frac{\ell^T \mathbf{P}(1) * l_Z - \ell^T \mathbf{P}(3) * l_X}{\ell^T \mathbf{P}(2)}$.

3.7. Extracting Faces

Apart from just the basic probing technique we also provide a method for extracting faces from the model. The user specifies points on the upper and lower edges of the face(s) to be extracted. We fit two lines through these sets of points. The user may specify the direction of these lines (see Figure 1(b)). In this case, the lines fitted are constrained to pass through the vanishing point in that direction. The heights are determined by picking up corresponding points (the line joining them passes through the vanishing point in the vertical direction) on the upper and lower lines. These corresponding points are taken as the *head-foot* pairs to calculate the height of the face. Here the user may also choose to enforce the constraint that many of the faces have the same height. Other constraints like the fact that two or more of the lines for different faces are parallel or coincident may also be imposed. In this case the height will be solved for by minimizing the error for all the faces together.

3.8. Extension to camera recovery from eight points

Consider the situation of Figure 6. Let us assume that the world X axis is not visible but we have four points on each plane necessary for their independent affine (or Euclidean) registration. Further, let us assume that the X directions in both the planes are parallel. This situation occurs commonly in practice (See Figure 1). We can use these points to define homographies $\mathbf{H'}$ and $\mathbf{G'}$ between the X-Z and X-Y planes and the image respectively.

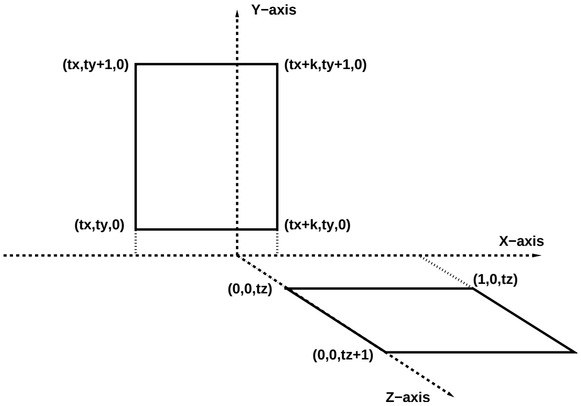


Figure 6: *Registration of planes using eight points. Common X axis not visible.*

Let us assume the X dimension of the ground plane rectangle to be the unit length of the world. Let $1/\gamma$ be the Z dimension of the ground plane rectangle and $1/\alpha$ and $1/\beta$ be the X and Y dimensions of the vertical plane rectangle. Also let t_x, t_y and t_z be the translations of the rectangles as shown in Figure 6.

Clearly, the homographies $\mathbf{H'}$ and $\mathbf{G'}$ are related to the homographies $\mathbf{H}$ and $\mathbf{G}$ defined before as follows:

$$\mathbf{H} = \mathbf{H'} * \begin{pmatrix} 1 & 0 & 0 \\ 0 & \gamma & -\gamma t_z \\ 0 & 0 & 1 \end{pmatrix}, \mathbf{G} = \mathbf{G'} * \begin{pmatrix} \alpha & 0 & -\alpha t_x \\ 0 & \beta & -\beta t_y \\ 0 & 0 & 1 \end{pmatrix} \tag{3}$$

Now substituting from Equation 3 in to Equation 1 and rewriting we have,

$$\mathbf{H'} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \gamma & -\gamma t_z \\ 0 & 0 & 1 \end{pmatrix} = \lambda * \mathbf{G'} * \begin{pmatrix} \alpha & 0 & -\alpha t_x \\ 0 & \beta & -\beta t_y \\ 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} -C_z & C_x & 0 \\ 0 & C_y & 0 \\ 0 & 1 & -C_z \end{pmatrix} \tag{4}$$

Using the fact that the first columns of both $\mathbf{H'}$ and $\mathbf{G'}$ is the vanishing point in the X direction and are hence equal up to scale, we have[7]

1. If one of αt_x, βt_y or γt_z is known then the other two can be found out by comparing the entries of the third column of Equation 4. We can also obtain λC_z from these 3 equations. Then by comparing the first columns we can compute $-\lambda\alpha C_z$. Now, from these two values we can also find the value of α.
2. Instead, if α is known then the ratio λC_z can be obtained from the first column. Plugging it in to the third column we have three equations in three unknowns (t_x, βt_y and γt_z) and we can solve in general.

Once we have obtained λC_z, αt_x, βt_y and γt_z, then from column 2 we can compute the other unknowns αC_x, βC_y and γC_z. Note that β and γ cannot be determined separately (unless given). They can be set to 1 without loss of generality and t_y, t_z, C_y and C_z can be computed up to scale.

Alternatively, we can use rough knowledge of the camera internal parameters $\mathbf{K}$ and experiment to adjust the parameters β and γ to obtain visual alignment from re-projections.

Once these parameters are computed we can use Equation 3 to compute $\mathbf{H}$ and $\mathbf{G}$. If the planes are orthogonal (or their relative orientation is known) then we can proceed as in Section 3.1 to compute the camera internals and 3D coordinates.

4. Metrology of curved surfaces

In this section we extend the basic method to deal with metrology of curved surfaces. As with plane measurements, modelling spheres and frustums from a single image also require the user to interactively specify some necessary constraints. In what follows we assume that the camera projection matrix $\mathbf{P}$ is known and describe the constraints for some cases.

4.1. Spheres

Modelling a sphere involves localization of its center and the computation of its radius. The silhouette of any sphere is an ellipse in the image. To resolve the depth scale ambiguity the user needs to provide additional information. We show that this can be achieved by specifying an intersecting or tangential plane.

4.1.1. Tangential Plane

The user specifies the silhouette in the image by clicking a minimum of 5 points on it. Having fitted an ellipse through these points, the system evaluates the equations of planes tangential to the sphere such that the camera center lies on them. Three such tangential planes (say $\mathbf{P_1}$, $\mathbf{P_2}$ and $\mathbf{P_3}$) localize the sphere center to a line passing through the camera center as shown in Figure 7. The constraint, that the known tangential plane (here, $\mathbf{P_0}$) is also equidistant from the sphere center, helps us localize the sphere center to two points on the line. One of these points is on the same side of the plane $\mathbf{P_0}$ as the camera center. This information can be used to resolve which of the two points is the required point. Note that one would encounter a degenerate case if the camera center lies on $\mathbf{P_0}$.

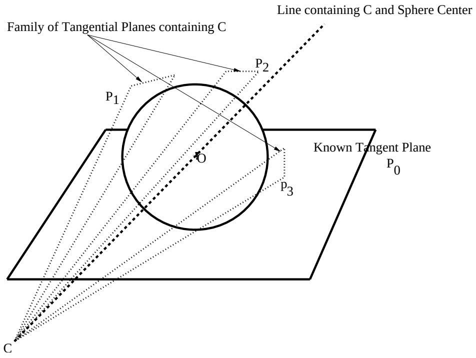

Line containing C and Sphere Center
Family of Tangential Planes containing C
P2
P1
O
Known Tangent Plane
P0
P3
C

Figure 7: *Tangent plane.*

4.1.2. Intersecting Plane

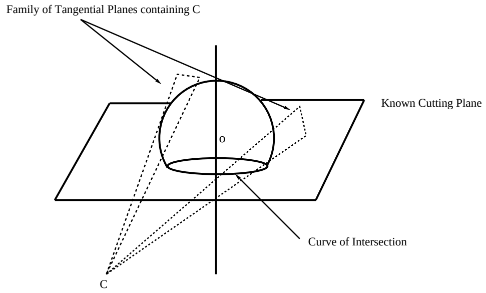


Figure 8: *Intersecting plane.*

The user may also choose to specify the curve of intersection of the sphere with a known intersecting plane in the image. This allows us to localize the sphere center to a line (say ℓ) normal to the intersecting plane and passing through the center of the curve of intersection as shown in Figure 8 (note that the curve of intersection of a sphere with an arbitrary plane is a circle in the world). As in the previous case the user is asked to specify the silhouette in the image. The two tangent planes $\mathbf{P_1}$ and $\mathbf{P_2}$ constrain the sphere center on their angle bisector plane. The intersection of the angle bisecting plane and ℓ is the sphere center.

4.2. Frustums

The user specifies the base circle of the frustum in the image. The user also specifies the left and right silhouettes of

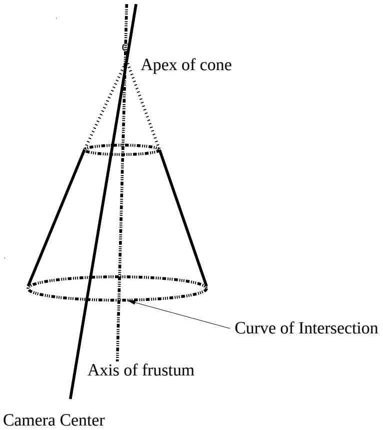


Figure 9: *Frustum.*

the frustums which are lines in the image. The intersection of these two lines in the image is the image of the apex of the cone, of which the frustum is a part as shown in Figure 9. This localizes the apex to the line $\mathbf{P}^{+}\mathbf{x_{apex}} + \lambda\mathbf{C}$ in the real world where $\mathbf{P}^{+}$ is the psuedo-inverse of $\mathbf{P}$ (i.e. $\mathbf{P}^{t}(\mathbf{P}\mathbf{P}^{t})^{-1}$) and $\mathbf{C}$ is the camera center. The intersection of this line with the axis of the base circle determines the apex in the world. The cone is thus determined. To find the height of the frustum, the system projects the line (say $\ell_{\mathbf{1}}$) joining the apex to the point nearest to the camera center on the base circle, onto the image. The user then clicks the point of intersection of this projected line with the upper curve of intersection. We shoot a ray through the camera center and this intersection point. The required point has thus been localized as the point of intersection of this ray and $\ell_{\mathbf{1}}$. The distance of this point from the base plane is the height of the frustum.

4.3. Modelling of general surfaces of revolution

Metrology of a general surface of revolution (SOR) involves computation of a space curve, which when rotated about a given axis produces that surface[13]. We find out the radii at different heights. As a result, we can now compute the generating curve and hence the surface of revolution.

The technique adopted requires the user to specify the axis of revolution in the world (by specifying a point on the axis **O** and its direction **dir** as shown in Figure 10). The axis direction can be computed using earlier methods of metrology. In addition, the user needs to mark the silhouette of the SOR. This method exploits the fact that the surface normal at any point on the SOR is co-planar with the axis of revolution.

A parametric curve is fitted to the silhouette and sampled uniformly. Consider the equation

$$\mathbf{C} = \mathbf{O} + \lambda_1\mathbf{dir} + \lambda_2\mathbf{n} + \lambda_3\mathbf{r}$$

where **n** is the surface normal at a silhouette point and **r** is the direction vector from the silhouette point to the camera center. The tangent line at a point on the curve in the image gives us the equation of the plane tangent to the SOR at that point. And thus we can determine **n** given a point on the curve. The direction vector **r** can be determined by shooting a ray from the camera center through the point on the silhouette. Note that there exists a unique solution to the 3 variables (the λ's).

Since the camera projection matrix is known, for a given point on the silhouette the corresponding point on the other silhouette at the same height can easily be computed. The radius for the height can be computed by enforcing the constraint that the corresponding points are at the same distance from the axis.

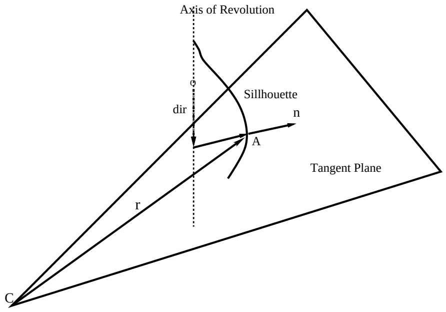


Figure 10: *General surfaces of revolution.*

4.4. Surfaces of translation

To model arches in monuments one also requires surfaces of translation. These are parameterized by splines and the depth of translation. The user clicks points on one of the end planes of the surface of translation. We require that this plane be known or computed by earlier metrology. Now using **P**, we can project these points from the image to the plane to get their location in the world plane. Subsequently, we fit a spline on these points and translate the spline by the required depth to get the model of surface of translation. This depth must also be determined by earlier metrology.

5. Results

We have modelled the Humayun's Tomb, in New Delhi using the primitives described in the previous section. The gross modelling was done from the image in Figure 1.

First the camera center was localized by registering two planes in the image as shown in Figure 1. We specified the quantity βt_y mentioned in Section 3. Recall that this is just the height of the origin on the vertical plane from the horizontal plane in terms of the vertical plane's unit length (refer to Figure 6). This sets up a coordinate system in the image and determines the projection matrix **P** of the camera in this coordinate system. We require the lengths of the units in the

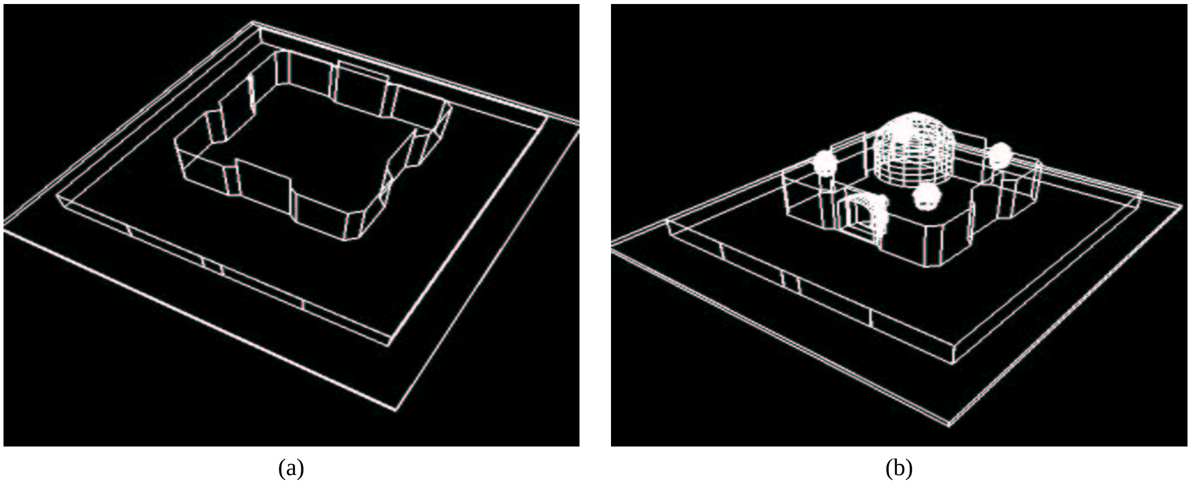


Figure 11: *Wire-frame model of (a) the faces, (b) showing the complete structure with the internal details of the front face registered.*

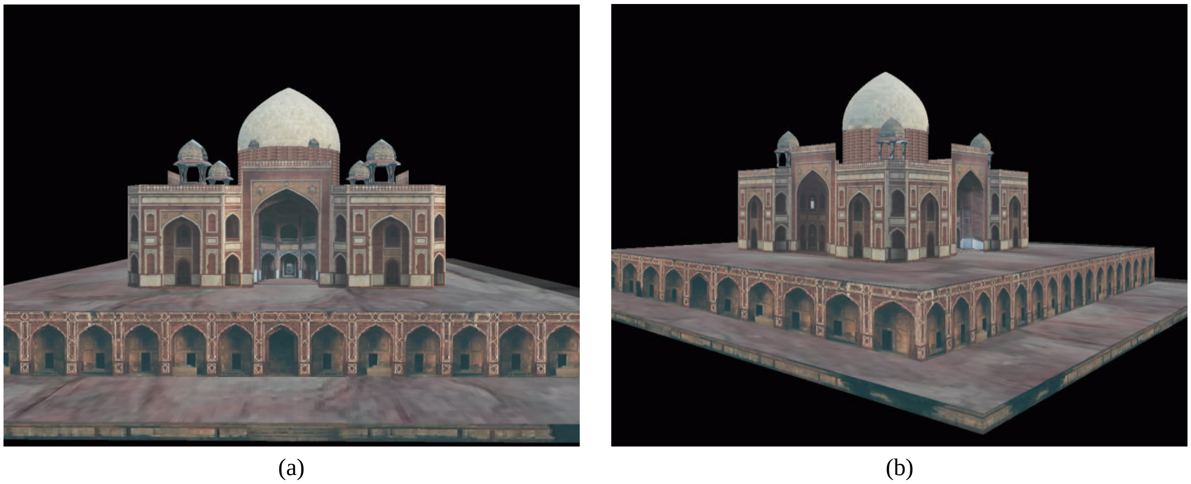


Figure 12: *Two snapshots from the final walk-through*

3-axes to correct the **P** to a Euclidean frame. The lengths in the X and Z directions were directly available from actual measurements, but the Y height was not possible to measure. This did not pose a serious difficulty because we could probe the height of the courtyard in terms of the unit of Y. We then used this height to relate the unit of Y to actual lengths in the world.

Once the height of the courtyard from the ground plane was known, all subsequent height measurements were carried out with respect to the plane of the courtyard. We solved for the common height of the planes together as explained in Section 3 and determined the height of the two side planes and the four angled planes visible in the image. The front plane's height was determined similarly and the rest of the sides of the monument which are not visible in the image were modelled using symmetry.

For modelling the various surfaces of revolutions in the images, the axes and a point on each axis were determined from earlier metrology using symmetry information. Recall that these are necessary for fitting of SOR's (see Section 4). The surfaces of revolution were then modelled from their respective silhouette edges.

We used the image in Figure 2 to carry out a detailed metrology of the area inside the front facade. For this we marked the two rectangles (H and V) on orthogonal planes as shown in Figure 2. Here the scales along the X and Z axis on the new axis were determined from the tie points shown in Figure 2(b). The parallel plane technique was used to move up from the H plane to the front face plane in two steps. The tie point coordinates in this new frame were computed by the probing technique and were equated with the ones obtained from the far metrology to get the scaling between the far and the detailed coordinate systems.

The unit length along the Z axis of the detailed coordinate system could not be obtained from our previous metrology and was measured separately. Then the heights of the planes and other detailed measurements were made on Figure 2(b) and the inner details were included in the model. In Figure 11 we show the model building process in wireframe. In Figure 12 we show two snapshots from the final walk-through.

6. Conclusion

We have presented an intuitive and interactive method for multilevel modelling of architectural scenes. The method is simple and flexible and yields high quality interactive walk-throughs.

References

1. A. Criminisi, I. Reid, and A. Zisserman. Single view metrology. *IJCV*, **40**(2):123–148, 2000. 2, 3, 4, 5, 6
2. P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: a hybrid geometry and image based approach. In *Proc. SIGGRAPH*, 1996. 1, 2, 3
3. P. E. Debevec, Y. Yu, and G. D. Borshukov. Efficient view-dependent image-based rendering with projective texture-mapping. In *Eurographics Rendering Workshop*, pages 105–116, 1998. 3
4. E. Grossmann, D. Ortin, and J. Santos-Victor. Single and multi-view reconstruction of structured scenes. In *Proc. ACCV*, 2002. 1, 2
5. R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000. 1, 2, 3, 4, 5
6. Y. Horry, K. .Anjyo, and K. Arai. Tour into the picture: using a spidery mesh interface to make animation from single image. In *Proc. SIGGRAPH*, pages 225–232, 1997. 2
7. A. M. Kushal, V. Bansal, and S. Banerjee. A simple method for interactive 3d reconstruction and camera calibration from a single view. In *Proc. Indian Conference in Computer Vision, Graphics and Image Processing*, 2002. (`http://www.cse.iitd.ernet.in/vglab/demo/single-view/2plane/`). 2, 3, 4, 5, 8
8. D. Leibowitz, A. Criminisi, and A. Zisserman. Creating architectural models from images. In *Proc. Eurographics*, pages 39–50, 1999. 1, 2
9. R.Cipolla and D. Robertson. 3D models of architectural scenes from uncalibrated images and vanishing points. In *Proc. 10th IAPR*, pages 824–829, 1999. 1, 2
10. R.Cipolla, D. Robertson, and E. G. Boyer. Photobuilder - 3D models of architectural scenes from uncalibrated images. In *Proc. IEEE International Conference on Multimedia Computing and Systems*, pages 25–31, 1999. 2
11. P. F. Strum and S. J. Maybank. A method for interactive 3D reconstruction of piecewise planar objects from single views. In *Proc. BMVC*, pages 265–274, 1999. 2
12. M. Wilczkowiak, E. Boyer, and P. Strum. Camera calibration and 3D reconstruction from single images using parallelepipeds. In *Proc. ICCV*, pages 142–148, 2001. 2
13. K.-Y. K. Wong, P. R. S. Mendonça, and R. Cipolla. Reconstruction of surfaces of revolution from single uncalibrated views. In P. L. Rosin and D. Marshall, editors, *Proc. British Machine Vision Conference 2002*, volume 1, pages 93–102, Cardiff, UK, September 2002. British Machine Vision Association. 9