# A Closed-form Solution to 3D Reconstruction of Piecewise Planar Objects from Single Images


Zhenguo Li[1], Jianzhuang Liu[1]
[1]Dept. of Information Engineering
The Chinese University of Hong Kong
{zgli5,jzliu}@ie.cuhk.edu.hk

Xiaoou Tang[1,2]
[2]Visual Computing Group
Microsoft Research Asia
xitang@microsoft.com



## Abstract

*This paper proposes a new approach to 3D reconstruction of piecewise planar objects based on two image regularities, connectivity and perspective symmetry. First, we formulate the whole shape of the objects in an image as a shape vector consisting of the normals of all the faces of the objects. Then, we impose several linear constraints on the shape vector using connectivity and perspective symmetry of the objects. Finally, we obtain a closed-form solution to the 3D reconstruction problem. We also develop an efficient algorithm to detect a face of perspective symmetry. Experimental results on real images are shown to demonstrate the effectiveness of our approach.*


## 1. Introduction

3D object reconstruction from images is an important research area in computer vision. It has many applications such as user-friendly query interface for 3D object retrieval and 3D photo-realistic scene creation for game, movie, and webpage design. The tremendous effort devoted to this area has indicated that inference of 3D shapes from 2D images is very challenging, and is generally underconstrained, even from multiple views of a scene or an object.

Extensive research has been done on 3D reconstruction from multiple views [1]. In most cases, however, we have only one view of an object or a scene. 3D object reconstruction from one single view is obviously a harder problem because less information is available in one view than in multiple views. Although this problem is generally ill-posed, the human visual system can easily perceive the 3D shapes of the objects in an image based on the knowledge learnt. In this paper, we focus on 3D reconstruction of piecewise planar objects from single images. These objects are often seen in daily life. Because so far there are no edge detection algorithms that can provide only the useful edges reliably in a common scene, some human interaction is required. In our system, the user draws the edges of the objects in an image, and then the system recovers the 3D shapes of the objects. In our approach, we formulate the whole shape of the objects in an image as a shape vector consisting of the normals of all the faces of the objects. On the shape vector, we develop several linear constraints using connectivity and perspective symmetry of the objects. A closed-form solution to the 3D reconstruction problem can be obtained. We also develop an efficient algorithm to detect a face of perspective symmetry.

## 2. Related work

Many methods have been proposed for 3D reconstruction from single images [2], [3], [4], [5], [6], [7], [8], and [9]. Zhang et al. [2] tried to reconstruct free-form 3D model from a single image. This method needs a lot of user interaction and may take more than an hour for the user to specify constraints from an image. Prasad et al. [3] recovered a curved 3D model from its silhouettes in an image, which is a development of that in [2] and does not need so much interaction, but the reconstructed objects are restricted to those whose silhouettes are contour generators where the surface normals are known. The Facade system [4] models a 3D building using parametric primitives from a single view or multiple views of the scene. Liebowitz et al. [5] created architectural models using geometric relationships from architectural scenes. Their method requires to rectify two or more planes and to compute the vanishing lines of all the planar faces. Besides, the reconstruction errors may accumulate. Sturm and Maybank's method [6] first does camera calibration, then recovers part of the points and planes by assuming the availability of some vanishing points and lines, and finally obtains the 3D positions of other faces by propagation. The drawbacks of this method are that the parts of the objects have to be sufficiently interconnected and the reconstruction errors may be accumulated. Jelinek and Taylor [7] proposed a method of polyhedron reconstruction from single images using several camera models. The


**1-4244-1180-7/07/$25.00 ©2007 IEEE**

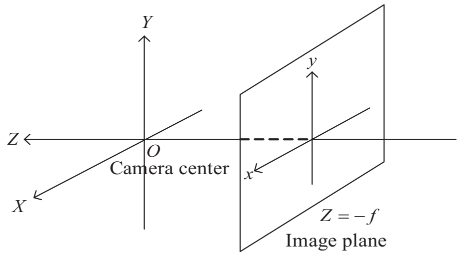


Figure 1. The camera model and the coordinate system.

main restriction of this method is that the polyhedra have to be linearly parameterized, which limits the application of the method. Shimodaira [8] used the shading information, one horizontal or vertical face, and convex and concave edges to recover the shape of polyhedra in single images. This method handles very simple polyhedra only. Li et al. [9] utilized image regularities, such as connectivity, parallelism, and orthogonality, to infer the 3D structure of piecewise planar objects. The common point in 3D reconstruction from single images is that user interaction is a necessary step.

The above methods do not exploit the regularity of symmetry for 3D reconstruction. Symmetry is known as one of the basic features of a large variety of objects and shapes. The fact that symmetry can impose strong constraints on 3D shapes has been noted for a long time. Kanade [10] first used skewed symmetry to help infer 3D shapes under orthography projection. Ulupinar and Nevatia [11] extended the skewed symmetry to convergent symmetry. Some researchers focused on 3D reconstruction of one symmetric object [12], [13] [14], [15].

## 3. The imaging model and the shape vector

In this paper, homogeneous coordinates are used for the analysis of the problem unless Euclidean coordinates are specified somewhere. Besides, due to limited cues presented in a common image, we assume that a simplified camera model is given with such a calibration matrix

$$\mathbf{K} = \begin{pmatrix} -f & 0 & 0 \\ 0 & -f & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{1}$$

where $f$ is the focal length that needs to be found. Furthermore, without loss of generality we align the world frame with the camera frame as shown in Fig. 1, where the image plane is $Z = -f$, and the projection matrix takes a simple form $\mathbf{P} = [\mathbf{K}|\mathbf{0}]$. The relation between a point $\mathbf{M} = (X, Y, Z, 1)^T$ in the world frame and its projection $\mathbf{m} = (x, y, 1)^T$ in the image is $\lambda\mathbf{m} = \mathbf{PM}$, from which we have $\lambda = Z$, $X = -Zx/f$, and $Y = -Zy/f$.

A scene is said to be projected from a *generic view* if perceptual properties in the image are preserved under slight variations of the viewpoint. We suppose that the objects in an image are the projection in a generic view where no any face is projected to a line. Let $\boldsymbol{\pi} = (a, b, c, d)^T$ denote a plane $ax + by + cz + d = 0$ in 3D space. It is easy to show that $d \neq 0$ when $\boldsymbol{\pi}$ is in a generic view. Assume, to the contrary, that $d = 0$. Then the camera center, $\mathbf{v} = (0, 0, 0, 1)^T$, satisfies $\boldsymbol{\pi}^T\mathbf{v} = d = 0$, implying that $\boldsymbol{\pi}$ passes through $\mathbf{v}$ and is projected as a line, which contradicts the assumption that $\boldsymbol{\pi}$ is in a generic view.

Therefore, a plane in a generic view can be written as $\boldsymbol{\pi} = (a, b, c, 1)^T = (\mathbf{R}^T, 1)^T$, where $\mathbf{R} = (a, b, c)^T$ is the normal of the plane. Since we are dealing with planar objects consisting of faces, $\mathbf{R}$ is also called the normal of a face that is passed through by $\boldsymbol{\pi}$. We represent the objects of interest in a scene with a vector $\mathbf{q}$ consisting of all the normals of the faces of the objects, i.e., $\mathbf{q} = (\mathbf{R}_1^T, \mathbf{R}_2^T, ..., \mathbf{R}_{N_f}^T)^T$, where $N_f$ is the number of faces. We call $\mathbf{q}$ the *shape vector* of the objects. It is worth noting that the objects are determined by the shape vector up to a scale. In Sections 5 and 6, we impose geometric constraints on the shape vector using the regularities of connectivity and perspective symmetry in the image.

## 4. Finding the focal length and the faces

We employ the well-known method presented in [16] to find the focal length as long as two finite vanishing points associated with two perpendicular directions in 3D space can be located in the image. From most man-made planar objects, it is not difficult for the user to find two sets of lines where in 3D space the lines in each set are parallel and two lines from different sets are perpendicular. For example, if the user can identify a quadrilateral that is the projection of a rectangular, then the two sets of lines are available.

Let $\mathbf{v}_1 = (x_1, y_1, 1)^T$ and $\mathbf{v}_2 = (x_2, y_2, 1)^T$ be the two vanishing points obtained from such two sets. Then $((\mathbf{K}^{-1}\mathbf{v}_1)^T, 0)^T$ and $((\mathbf{K}^{-1}\mathbf{v}_2)^T, 0)^T$ are the associated directions in 3D space [16]. Thus $(\mathbf{K}^{-1}\mathbf{v}_1)^T(\mathbf{K}^{-1}\mathbf{v}_2) = 0$, from which we have

$$f = \sqrt{-(x_1x_2 + y_1y_2)}. \tag{2}$$

In the case where there are more than one pair of such vanishing points, we take the average of the values obtained by (2) as the final focal length. In what follows, we assume that the focal length $f$ has been obtained using this method (or some other methods).

After the user draws a line drawing along the edges of the objects in an image, we use the algorithm proposed in [20] to identify the faces of the objects. Therefore, we also assume that the faces of the objects are available.

## 5. Connectivity

In a planar object, a vertex is often shared by more than one face. This connectivity leads to constraints that relate the normals of these faces through the vertex.

Let $\mathbf{x} = (x, y, 1)^T$ be the imaged vertex of $\mathbf{X} = (X, Y, Z, 1)^T$ which lies on both the $i^{th}$ face (plane) $\boldsymbol{\pi}_i = (a_i, b_i, c_i, 1)^T = (\mathbf{R}_i^T, 1)^T$ and the $j^{th}$ face (plane) $\boldsymbol{\pi}_j = (a_j, b_j, c_j, 1)^T = (\mathbf{R}_j^T, 1)^T$. Then, $\lambda\mathbf{x} = \mathbf{PX}$, $\boldsymbol{\pi}_i^T\mathbf{X} = 0$, and $\boldsymbol{\pi}_j^T\mathbf{X} = 0$, where $\lambda$ is some nonzero scalar. From these relations, we have

$$\mathbf{R}_i^T\mathbf{x}' = \mathbf{R}_j^T\mathbf{x}' = f/Z, \tag{3}$$

where $\mathbf{x}' = (x, y, -f)^T$ is the 3D Euclidean coordinate of $\mathbf{x}$. Furthermore, we have

$$(\mathbf{x}'^T, -\mathbf{x}'^T) \cdot (\mathbf{R}_i^T, \mathbf{R}_j^T)^T = 0. \tag{4}$$

The constraint in (4) is called the shared vertex constraint. Similarly, if the vertex $\mathbf{x}'$ is shared by $n$ faces, $\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, ..., \boldsymbol{\pi}_n$, we have $n-1$ independent constraints:

$$(\mathbf{x}'^T, -\mathbf{x}'^T) \cdot (\mathbf{R}_k^T, \mathbf{R}_{k+1}^T)^T = 0, \;\; k = 1, 2, ..., n-1. \tag{5}$$

In terms of the shape vector $\mathbf{q} = (\mathbf{R}_1^T, \mathbf{R}_2^T, ..., \mathbf{R}_{N_f}^T)^T$, (5) can be written in matrix form:

$$\mathbf{A}_1\mathbf{q} = \mathbf{0}, \tag{6}$$

where $\mathbf{A}_1$ is a matrix of size $(n-1) \times (3N_f)$. This is the shared vertex constraint contributed by one imaged vertex $\mathbf{x}'$ shared by the $n$ faces. Putting the constraints from all the shared vertices together, we have

$$\mathbf{Aq} = \mathbf{0}, \tag{7}$$

where $\mathbf{A} = (\mathbf{A}_1^T, \mathbf{A}_2^T, ..., \mathbf{A}_{N_v}^T)^T$, $N_v$ is the number of all the vertices of the objects. We call (7) the *connectivity constraints*.

# 6. Perspective symmetry

A planar face of bilateral symmetry usually does not exhibit bilaterally symmetric in its perspective projection. However, this projection still preserves some kind of symmetry, which is called the *perspective symmetry*. In this projection, all lines joining corresponding points intersect at a common point called the *perspective point*, and the projection of the symmetry axis is called the *perspective symmetry axis*. In this section, we formulate constraints on the shape vector $\mathbf{q}$ using perspective symmetry. We also develop an efficient algorithm to detect faces of perspective symmetry.

It should be noted that a non-bilaterally symmetric face may be projected as a face of perspective symmetry. For example, a face of perspective symmetry is always projected as another face of perspective symmetry. However, it is a common experience that a face of perspective symmetry in an image is usually the projection of a face of bilateral symmetry in 3D space. Previous research has shown that this phenomenon is non-accidental [19]. Therefore, it is reasonable to make the following assumption.

**Assumption 1.** *A face of perspective symmetry in an image is the projection of a face of bilateral symmetry.*

## 6.1. Constraints on the Shape Vector

The orientation of a face of bilateral symmetry turns out to be encoded in its projection, as stated in the following theorem.

**Theorem 1.** *The normal $\mathbf{R}$ of a face of bilateral symmetry in 3D space is given by*

$$\mathbf{R} = (\mathbf{K}^{-1}\mathbf{x}) \times (\mathbf{K}^{-1}\mathbf{x} \times \mathbf{K}^T\mathbf{l}). \tag{8}$$

*where $\mathbf{K}$ is the calibration matrix, $\mathbf{x}$ and $\mathbf{l}$ are the perspective point and the perspective symmetry axis of the projection of the face.*

The proof can be found in [17]. Below we impose constraints on the shape vector $\mathbf{q}$ using this theorem. Let $F_i$, $\mathbf{x}_i$ and $\mathbf{l}_i$, $i = 1, ..., m$, be the faces of perspective symmetry, the corresponding perspective points, and the perspective symmetry axes, respectively. Let $\mathbf{c}_i^1 = \mathbf{K}^{-1}\mathbf{x}_i$ and $\mathbf{c}_i^2 = \mathbf{K}^{-1}\mathbf{x}_i \times \mathbf{K}^T\mathbf{l}_i$. Then from (8), we have $\mathbf{c}_i^{1T}\mathbf{R}_i = \mathbf{c}_i^{2T}\mathbf{R}_i = 0$. Thus

$$\mathbf{B}_i^T\mathbf{q} = \mathbf{0}, \;\; i = 1, ..., m, \tag{9}$$

where $\mathbf{B}_i = (\mathbf{0}_{2\times3(i-1)} \;\; \mathbf{C}_i \;\; \mathbf{0}_{2\times3(N_f-i)})^T$, $\mathbf{C}_i = (\mathbf{c}_i^1, \mathbf{c}_i^2)^T$, and $\mathbf{0}_{2\times3(i-1)}$ and $\mathbf{0}_{2\times3(N_f-i)}$ are two zero matrices of size $2 \times 3(i-1)$ and $2 \times 3(N_f - i)$, respectively. Putting all these equations in matrix form, we have

$$\mathbf{Bq} = \mathbf{0}, \tag{10}$$

where $\mathbf{B} = (\mathbf{B}_1, \mathbf{B}_2, ..., \mathbf{B}_m)^T$. We call (10) the *symmetry constraints*.

## 6.2. Detecting faces of perspective symmetry

In this section, we discuss how to detect faces of perspective symmetry, as well as the corresponding perspective points and the perspective symmetry axes.

Given a face of perspective symmetry, there exists a transformation that maps a point to its corresponding point [18]. In particular, under this transformation, the perspective point is a fixed point, and the perspective symmetry axis is a line of fixed points. Such a transformation can be represented as a $3 \times 3$ nonsingular matrix, denoted by $\mathbf{H} = (h_{ij})_{3\times3}$. Note that $\mathbf{H}$ has 8 degrees of freedom in homogeneous coordinates. Once $\mathbf{H}$ is known, the perspective point is given by the eigenvector of $\mathbf{H}$ corresponding to the eigenvalue with algebraic multiplicity 1, and the perspective line is determined by the two eigenvectors of $\mathbf{H}$ corresponding to the eigenvalue with algebraic multiplicity 2 [18]. Below we show how to find $\mathbf{H}$.

Let $(\mathbf{x}_i, \mathbf{x}_i^*)$, $i = 1, 2, ..., n$, be $n$ pairs of corresponding points. Then we have $\mathbf{x}_i^* \times \mathbf{H}\mathbf{x}_i = \mathbf{0}$ and $\mathbf{x}_i \times \mathbf{H}\mathbf{x}_i^* = \mathbf{0}$, $i = 1, 2, ..., n$, which can be expressed in matrix form as

$$\mathbf{S}\mathbf{h} = \mathbf{0}, \tag{11}$$

where $\mathbf{h} = (h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, h_{31}, h_{32}, h_{33})^T$. The optimal solution to (11) in the sense of least squared error is well known as the eigenvector of $\mathbf{S}^T\mathbf{S}$ associated with the smallest eigenvalue.

Before giving the algorithm for detecting a face $F$ of perspective symmetry, it is necessary to discuss two facts. Let $F$ be a polygon with $\{\mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_{N-1}\}$ being the set of clockwise ordered corner points. Then a corner point of $F$ must be mapped to another corner point or itself. The second fact is that given one pair of corresponding corner points, other pairs of corresponding corner points can be deduced. Specifically, if $\mathbf{x}_i$ corresponds to $\mathbf{x}_j$, then $\mathbf{x}_{(i-k)\mathrm{mod}N}$ corresponds to $\mathbf{x}_{(j+k)\mathrm{mod}N}$, $k = 0, 1, ..., N-1$. Based on $\mathbf{H}$ and these facts, we develop Algorithm 1.

**Algorithm 1** Detecting a Face of Perspective Symmetry

1: Let $\{\mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_{N-1}\}$ be the ordered corner points of the face.
2: For every $\mathbf{x}_i$, take $(\mathbf{x}_{(1+k)\mathrm{mod}N}, \mathbf{x}_{(i-k)\mathrm{mod}N})$ as the pair of corresponding points, $k = 0, 1, ..., N-1$.
3: Form $S$ in (11) using $(\mathbf{x}_{(1+k)\mathrm{mod}N}, \mathbf{x}_{(i-k)\mathrm{mod}N})$, $k = 0, 1, ..., N-1$.
4: Solve for $\mathbf{h}_i$ ($\mathbf{H}_i$) from (11).
5: Compute the perspective symmetry cost $c_i = \sum_{k=0}^{N-1} d(\mathbf{H}_i(\mathbf{x}_{(1+k)\mathrm{mod}N}), \mathbf{x}_{(i-k)\mathrm{mod}N})$, where $d(\cdot,\cdot)$ denotes the Euclidean distance between two points in Euclidean coordinates.
6: If $c_j = \min_i\{c_i\} < t$, where $t$ is a threshold, then the face is judged as a face of perspective symmetry, and the perspective point and the perspective symmetry axis are computed from $\mathbf{H}_j$.

## 7. Constraint by fixing a vertex

Prior to 3D reconstruction, we set a depth (Z-coordinate) $Z_0$ to one vertex $\mathbf{x}_0 = (x_0, y_0)$ which is assumed to be on the $i^{th}$ face with normal $\mathbf{R}_i$. Note that it should satisfy $-Z_0 > f$ since the 3D objects are on the right-hand side of the image plane (see Fig. 1). Then by (3) we have

$$\mathbf{R}_i^T\mathbf{x}_0' - f/Z_0 = 0, \tag{12}$$

where $\mathbf{x}_0' = (x_0, y_0, -f)^T$ is the 3D Euclidean coordinate of $\mathbf{x}_0$. Let $\mathbf{E} = (\mathbf{0}_{1\times3(i-1)}, \mathbf{x}_0'^T, \mathbf{0}_{1\times3(N_f-i-1)})$. Then (12) can be rewritten as

$$\mathbf{E}\mathbf{q} = f/Z_0. \tag{13}$$

## 8. Constraint by a common plane

In many cases, there are more than one object present in an image. To obtain the 3D reconstruction of these objects simultaneously, we need to impose an additional constraint to relate them. A direct and simple cue comes from the observation that these objects are usually located on a common plane, such as those shown in the experiments. This cue is pointed out by the user specifying which vertices are on this common plane. Suppose that the common face is the $j^{th}$ face of the objects. For instance, for the objects in Fig. 2(a), the common plane is the one passing through the $6^{th}$ face (see Fig. 2(b)), and the vertices 1, 3, 11, 12, 16, 21, 30, and 31 are on this face (see Fig. 2(c)). Thus the $1^{st}$ vertex is on both the $1^{st}$ face and the $6^{th}$ face, the $3^{rd}$ vertex is on the $1^{st}$, the $5^{th}$, and the $6^{th}$ faces, and so on. Similar to the analysis in Section 5, we can impose these connectivity constraints on the shape vector in matrix form

$$\mathbf{G}\mathbf{q} = \mathbf{0}. \tag{14}$$

## 9. Algorithm for 3D reconstruction

So far we have obtained a number of constraints on the shape vector with connectivity, perspective symmetry, fixing a vertex, and a common plane, which result in the following equation:

$$\mathbf{M}\mathbf{q} = \mathbf{b}, \tag{15}$$

where $\mathbf{M} = (\mathbf{A}^T, \mathbf{B}^T, \mathbf{G}^T, \mathbf{E}^T)^T$, $\mathbf{b} = (0, ..., 0, f/Z_0)^T$. Thus the optimal solution, in the sense of least squared error, to this system is $\mathbf{q} = (\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T\mathbf{b}$. The complete algorithm for 3D reconstruction is listed in Algorithm 2.

**Algorithm 2** 3D Reconstruction of Piecewise Planar Objects from Single Images

1: Draw lines along the edges of the objects in an image.
2: Use the algorithm in [20] to find the faces of the objects.
3: Form the connectivity constraints in (7): $\mathbf{A}\mathbf{q} = \mathbf{0}$.
4: Find the faces of perspective symmetry with Algorithm 1.
5: Form the symmetry constraints in (10): $\mathbf{B}\mathbf{q} = \mathbf{0}$.
6: Form the depth constraint in (13): $\mathbf{E}\mathbf{q} = f/Z_0$.
7: Form the common plane constraint in (14) when there are multiple objects: $\mathbf{G}\mathbf{q} = \mathbf{0}$.
8: Form the system in (15): $\mathbf{M}\mathbf{q} = \mathbf{b}$.
9: Compute the shape vector $\mathbf{q} = (\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T\mathbf{b}$.
10: For any vertex $(X, Y, Z)$ (with $\mathbf{x}' = (x, y, -f)^T$ as its image in the image plane), compute all the Z-coordinate $Z_i$ by $Z_i = f/(\mathbf{R}_i^T\mathbf{x}')$ (see (3)) if it is on $i^{th}$ face (with normal $\mathbf{R}_i$). Take the average as the final Z-coordinate $Z$ if there are multiple $Z_i$. Compute the X-coordinate and Y-coordinate by $X = -Zx/f, Y = -Zy/f$.

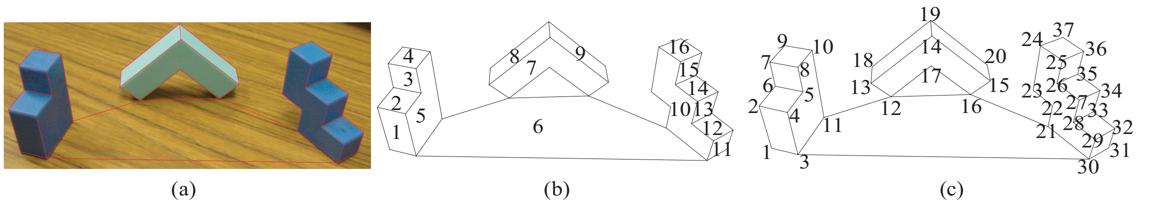


Figure 2. (a) An image with lines drawn on the edges of the desired objects. (b) The 16 faces. (c) The 36 vertices of the objects.

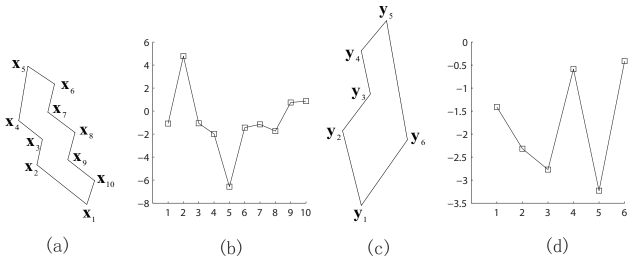


Figure 3. (a) The $10^{th}$ face with ordered corner points. (b) The $\log(c)$ vs. vertex figure for (a). (c) The $5^{th}$ face with ordered corner points. (d) The $\log(c)$ vs. vertex figure for (c).

## 10. Experiments

To illustrate our approach, we first focus on detecting faces of perspective symmetry with the image in Fig. 2(a), then give a number of 3D reconstruction examples.

### 10.1. Detecting faces of perspective symmetry

We first draw lines along the edges of the objects in Fig. 2(a). Then the 16 faces found by the algorithm in [20] are shown in Fig. 2(b). Next we use Algorithm 1 to identify the faces of perspective symmetry from these 16 faces.

Let us take two faces in the objects as examples. Fig. 3(a) shows the $10^{th}$ face (see Fig. 2(b)) with ordered corner points. Fig. 3(b) is the $\log(c)$ vs. vertex figure where $c$ is the perspective symmetry cost (see Algorithm 1). It is clear that the pair $(\mathbf{x}_1, \mathbf{x}_5)$ is of the smallest cost 0.0014. In all our experiments, the threshold $t$ is set to 0.01. Thus this face is considered as a face of perspective symmetry.

For the $5^{th}$ face with ordered corner points shown in Fig. 3(c), Fig. 3(d) is the $\log(c)$ vs. vertex figure. In this case, the pair $(\mathbf{x}_1, \mathbf{x}_5)$ results in the smallest perspective symmetry cost 0.040. since $0.040 > t$, this face is not considered as a face of perspective symmetry.

For the objects in Fig. 2(a), 14 faces (1, 2, 3, 4, 7, 8, 9, 10, 11, 12, 13, 14, 15, and 16) are identified as faces of perspective symmetry.

### 10.2. 3D Reconstruction

We have conducted a number of experiments on real images to verify the effectiveness of our approach. Due to the space limitation, only part of them are given here. In Fig. 4, (a1)–(d1) are the original images with the drawn red lines superimposed on the edges of the objects, and (a2)–(d2) and (a3)–(d3) are the reconstructed 3D objects with texture mapped, with each result shown in two different views.

Our algorithm can reconstruct objects with hidden lines drawn by the user. This provides more useful 3D information from the reconstruction result. In Figs. 4(a1) and (b1), no hidden edges are drawn. In Fig. 4(c1), all the hidden edges are drawn which are guessed by the user. Fig. 4(c3) shows the backs of the buildings. In Fig. 4(d1), part of the hidden edges of the buildings are drawn. These results clearly demonstrate that our method successfully creates the desired 3D objects from the images.

## 11. Conclusions

We have proposed an approach to reconstructing 3D piecewise planar objects from single images based on connectivity and perspective symmetry. The objects in an image are represented by a shape vector consisting of the normals of the faces in the objects. On the shape vector, a number of linear constraints are imposed using the connectivity and perspective symmetry of the objects. Finally, a closed-form solution for the shape vector can be obtained. We also develop an efficient algorithm for detecting faces of perspective symmetry. If the user provides the hidden edges, our algorithm can recover both the visible and invisible shapes of the objects. Experiments have demonstrated the effectiveness of our approach.

## Acknowledgements

This work was supported by the Research Grants Council of the Hong Kong SAR (Project No. CUHK 414306) and the CUHK Direct Grant.

## References

[1] R. I. Hartley and A. Zisserman. Multiple View Geometry in Computer Vision. Cambridge University Press, second edition, 2004.

[2] L. Zhang, G. Dugas. Phocion, J. S. Samson, and S. M. Seitz. Single view modeling of free-form scenes. *CVPR*, pp. 990–997, 2001.

[3] M. Prasad, A. Zisserman, and A. W. Fitzgibbon. Fast and controllable 3D modelling from silhouettes. *Proc. of the 26th Annual Conference of the European Association for Graphics, Dublin*, pp. 9–12, Sept. 2005.

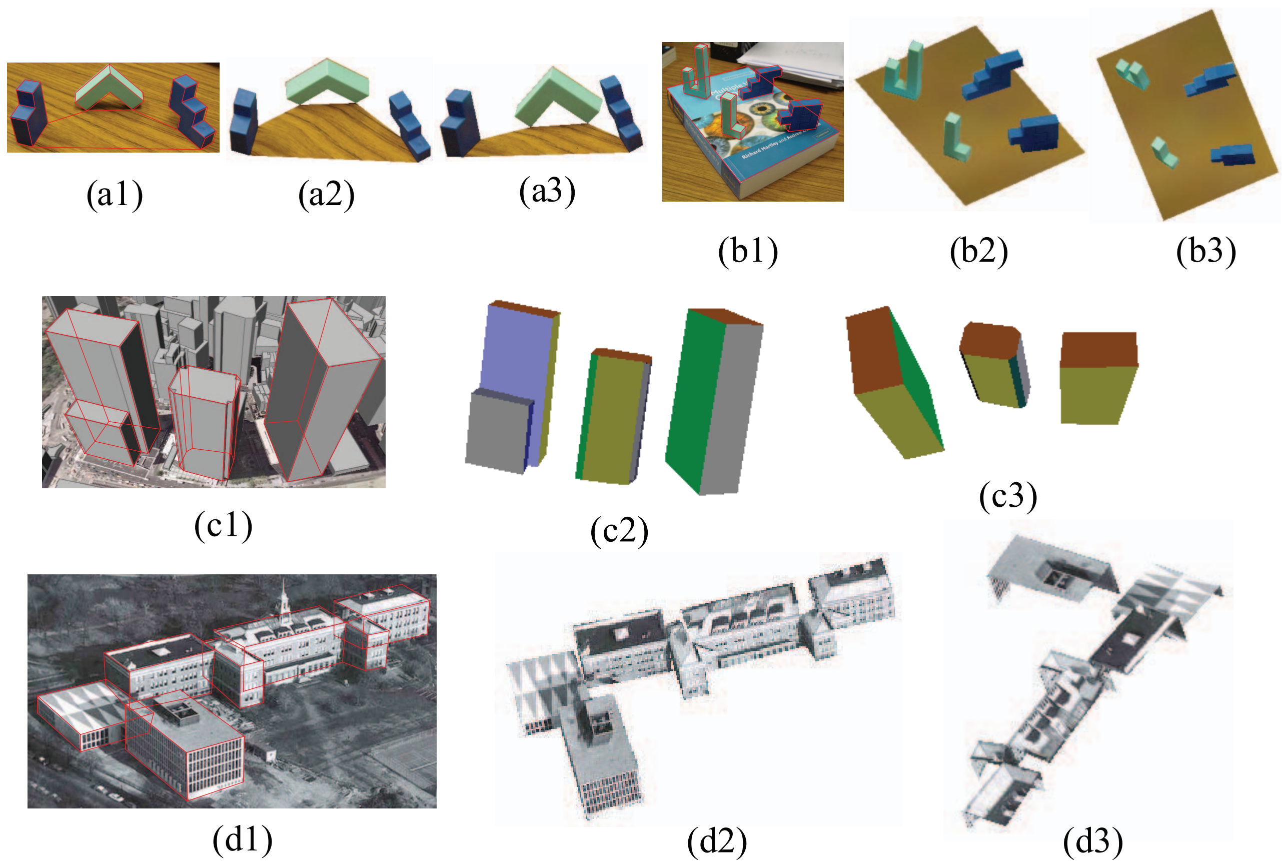


Figure 4. (a1)–(d1) Original images with drawn edges (red lines) superimposed. (a2)–(d2) One view for each reconstructed result. (a3)–(d3) Another view for each reconstructed result.


[4] P. E. Debevec, C.J. Taylor and J. Malik. Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach. *Proc. SIGGRAPH*, pp. 11-20, 1996.

[5] D. Liebowitz, A. Criminisi, and A. Zisserman. Creating architectural models from images. *Proc. EuroGraphics*, vol. 18, pp. 39–50, Sept. 1999.

[6] P. F. Sturm and S. J. Maybank. A method for interactive 3d reconstruction of piecewise planar objects from single images. *BMVC*, 1999.

[7] D. Jelinek and C. J. Taylor. Reconstruction of linearly parameterized models from single images with a camera of unknown focal length. *IEEE Trans. PAMI*, vol. 23, no. 7, pp. 767–773, 2001.

[8] H. Shimodaira. A shape-from-shading method of polyhedral objects using prior information. *IEEE Trans. PAMI*, vol. 28, no. 4, pp. 612–624, 2006.

[9] Z. Li, J. Liu and X. Tang. Shape from regularities for interactive 3D reconstruction of piecewise planar objects from single images. *Proc. ACM Multimedia*, pp. 85–88, 2006.

[10] T. Kanade. Recovery of the three-dimensional shape of an object from a single view. *Artificial Intelligence*, vol. 17, pp. 409–460, 1981.

[11] F. Ulupinar and R. Nevatia. Constraints for interpretation of line drawings under perspective projection. *Computer Vision, Graphics, and Image Processing*, vol. 53, no. 1, pp. 88–96, 1991.

[12] A. Francois, G. Medioni, and R. Waupotitsch. Rconstructing mirror symmetric scenes from a single view using 2-view stereo geometry. *ICPR*, vol. 4, pp. 12–16, 2002.

[13] W. Hong, Y. Ma, and Y. Yu. Reconstruction of 3-D symmetric curves from perspective images without discrete features. *ECCV*, pp. 533–545, 2004.

[14] C. A. Rothwell, D. A. Forsyth, A. Zisserman, and J. L. Mundy. Extracting projective structure from single perspective views of 3-D point sets. *ICCV*, pp. 573–582, 1993.

[15] I. Shimshoni, Y. Moses, and M. Lindenbaum. Shape restruction of 3-D bilateral symmetric surfaces. *IJCV*, vol. 39, no. 2, pp. 97–110, 2000.

[16] B. Caprile and V. Torre. Using vanishing points for camera calibration. *IJCV*, vol. 4, no. 2, pp. 127–140, 1990.

[17] Z. Li. 3D Reconstruction of Piecewise Planar Objects from Single Images. Technical Report, Dept. of IE, The Chinese University of Hong Kong, 2007.

[18] C. E. Springer. *Geometry and Analysis of Projective Spaces*. Freeman, 1964.

[19] D. G. Lowe. Perceptual Organization and Visual Recognition. Boston: Kluwer Academic Publishers, 1985.

[20] J. Liu and Y. T. Lee. Graph-based method for face identification from a single 2D line drawing. *IEEE Trans. PAMI*, vol. 23, no. 10, pp. 1106–1119, 2001.