Symmetry-Based Photo Editing

Kun Huang Wei Hong Yi Ma Department of Electrical and Computer Engineering Coordinated Science Laboratory University of Illinois at Urbana-Champaign {kunhuang, weihong, yima}@uiuc.edu

Abstract

Based on high-level geometric knowledge, especially symmetry, imposed upon objects in images, we demonstrate in this paper how to edit images in terms of correct 3-D shape and relationships of the objects, without explicitly performing full 3-D reconstruction. Symmetry is proposed as the central notation that unifies both conceptually and algorithmically all types geometric regularities such as parallelism, orthoganality, and similarity. The methods are extremely simple, accurate, and easy to implement, and they demonstrate the power of applying scene knowledge.

Keywords: vision and scene understanding, high-level knowledge, structure from symmetry, image editing, 3-D scene recovery.

1 Introduction

One of the central issues in computer vision is to perform realistic 3-D reconstruction based on images of a scene taken from different vantage points. A common approach to achieve this goal is the structure from motion technique that involves reconstruction from multiple images. This line of work has led to the development of multiple view geometry. In classic multiple view geometry, we usually do not apply any knowledge about the scene. Typically, only image primitives such as points, lines and planes are used, and no knowledge about their spatial relationships is assumed. However, recently more and more work has unveiled the usefulness of scene knowledge in reconstruction [17, 2, 13, 7]. While various types of scene knowledge and simplifying assumptions can be imposed upon photometry and shape, it is the geometric knowledge that we will be focusing on in this paper. Geometric knowledge, such as parallelism and orthoganality, prevails in man-made environment. It provides useful cues in retrieving from images shapes of objects and spatial relationships between objects. As we will demonstrate in this paper, if we no longer confine ourselves to primitive geometric features but instead begin to apply global geometric information, it opens many

new avenues and possibilities such as editing images even without explicitly performing 3-D reconstruction.

Geometric scene knowledge such as object shapes and spatial relationships between objects are always related to some types of "regularities." For object shapes, regular shapes such as rectangles, squares, diamonds, and circles always capture our attention more than the others. For spatial relationship between objects, parallelism, orthoganality and similarity are the most conspicuous ones. Interestingly, all such regularities can be encompassed under the notion of *symmetry*. For instance, a rectangular window has one rotational symmetry (by 180°) and two reflective symmetries; windows on the side of a wall display translational symmetry; the corner of a cube admits a rotational symmetry, etc. There are many studies using symmetries in the scene for reconstruction purposes [8, 6, 1, 3, 18, 19, 14, 15, 9].

Recently, a set of algorithms using symmetry for reconstruction from a single image or multiple images [7] have been developed, which leads to further studies on geometric segmentation [15], large baseline matching and 3-D reconstruction [9]. In [15], symmetry information has been applied for segmentation purposes. In each image, by identifying symmetry cells-regions that are images of symmetric objects such as rectangles, 3-D information about these symmetry cells are obtained. The image is then segmented based on the geometric information such as coplanarity and shape similarity of the symmetry cells. Identified symmetry cells can also be used as high-level features for matching purposes [9]. For symmetry cells found in different images, by comparing their 3-D shape and color information, feature matching is established and camera motion is calculated. With known camera motions, 3-D reconstruction is efficient and accurate. These are examples of utilizing highlevel geometric knowledge in modeling and motion analysis. In this paper, we will extend these techniques to another application - photo editing.

Photo editing has become an important part in digital photography and graphics. In many cases it is desired to edit the photos such that consistent geometry and perspective effects are conserved. Examples of such actions include removing unwanted reflections from a building facade and adding a logo on the ground for a football game scene (watermarks). However, common photo editing software such as Photoshop still lacks such functions with easy interfaces. Currently, the photos are still manipulated at the level of image regions, in which the preservation of geometry and perspective is usually "achieved" by manual adjustment. With the symmetry-based algorithms, it is possible to manipulate such 2-D image regions in terms of their correct 3-D shapes and relationships. For instance, when we apply the knowledge that one object is rectangular in 3-D, its pose and size is automatically determined from its image. This opens many new functionalities and applications that would otherwise be very difficult without applying such scene knowledge. In this paper, we demonstrate a primitive system that can easily and accurately perform a series of photo editing functions based on 3-D shape and relationship with minimal amount of manual intervention.

Relation to prior work. The effects of symmetry in 3-D reconstruction and recognition have been studied extensively. It leads to the single-view geometry of symmetry. [11] studied how to reconstruct a 3-D object using reflective symmetry induced by a mirror. [4] used a series of regular shapes to reconstruct the building structure based on highlevel knowledge. [6] studied reflective and rotational symmetries under perspective projection. Some more recent applications and studies can be found in [16, 17, 5]. [13, 10, 2] showed that other scene knowledge (e.g., length ratio, vanishing point, line etc.) also allows accurate reconstruction of 3-D structural metric and camera pose from images. In 3-D object and pose recognition, [12] pointed out that the assumption of reflective symmetry can also be used in the construction of projective invariants and is able to eliminate certain restrictions on the corresponding points. These invariants can also be formulated using Grassman-Cayley algebra, as pointed out in [3]. Recently [7] has systematically studied reconstruction from symmetry. Its implication on geometry-based image segmentation and large-baseline feature matching is given in [15, 9] As a continuation, this paper is to show some possible applications of applying symmetry knowledge about the scene.

2 Symmetry-based cell reconstruction

Here we briefly introduce some techniques for 3-D pose and shape recovery using symmetry knowledge. To use the symmetry-based algorithm, we start from images the basic symmetric objects called *symmetry cells*. While the symmetry cell can be image of any symmetric object, we just use one of the simplest symmetric object–rectangle to illustrate the reconstruction process. Once a (rectangle) symmetry cell in a plane is identified, it is then used to recover the 3-D pose of the plane. When multiple planes are present, a further step of alignment is necessary to obtain their correct 3-D relationships.

2.1 Reconstruction from a single view of one symmetry cell

First let look at 3-D reconstruction of plane pose using symmetry cells. The 3-D reconstruction of symmetric objects from a single image has been thoroughly studied in [7]. However, in the case of a rectangle, the reconstruction process can be significantly simplified using the fact that the two pairs of parallel edges of the rectangle give rise to two vanishing points in the image, as shown in Figure **1**. A van-



Figure 1: Image formation for a rectangle. v_1 and v_2 are two vanishing points. u is the intersection of the diagonal of the four-sided polygon and is the image of the rectangle center.

ishing point $\boldsymbol{v} = [x, y, z]^T \in \mathbb{R}^3$, expressed in homogeneous coordinates,¹ is exactly the direction of the parallel lines in space that generates \boldsymbol{v} . The two vanishing points \boldsymbol{v}_1 and $\boldsymbol{v}_2 \in \mathbb{R}^3$ associated with the image of a rectangle should be perpendicular to each other:

$$\boldsymbol{v}_1^T \boldsymbol{v}_2 = 0. \tag{1}$$

In addition, the unit normal vector $N \in \mathbb{R}^3$ of rectangle plane can be obtained by $N \sim v_1 \times v_2$, where \sim means equality up to a scalar factor.

If we attach an object coordinate frame on the plane with the frame origin being the center of the rectangle, the normal vector of the plane being the z-axis and the other two axis parallel to the two pairs of edges of the rectangle, then our goal is to find out the transformation $g = (R,T) \in SE(3)$ between the camera frame and the

¹Homogeneous coordinates of an image point can be viewed as the vector from the camera center to the image point in 3-D space.

object frame. Here $R \in SO(3)$ is the rotation and $T \in \mathbb{R}^3$ in the translation. Note that R is independent of the choice of the object frame origin. In the absence of noise, the pose (R, T) is simply:

$$R = \left[\frac{\boldsymbol{v}_1}{\|\boldsymbol{v}_1\|}, \frac{\boldsymbol{v}_2}{\|\boldsymbol{v}_2\|}, \frac{N}{\|N\|}\right], \quad T = \alpha \boldsymbol{u},$$

where $\boldsymbol{u} \in \mathbb{R}^3$ is the (homogeneous) image of the center of the rectangle and $\alpha \in \mathbb{R}_+$ is some scale factor to be determined. In the presence of noise, the so-obtained Rmay not be in SO(3), and we need to project it onto SO(3). The projection can be obtained by taking the singular value decomposition (SVD) of $R = U\Sigma V^T$ with $U, V \in O(3)$. Then the rotation is $R = UV^T$. To fix the scale in T, we typically choose the distance d from the camera center to the rectangle plane to be 1, which means that $T = \alpha \boldsymbol{u}$ with $\alpha = \frac{d}{\boldsymbol{u}^T N} = \frac{1}{\boldsymbol{u}^T N}$.

2.2 Alignment of multiple symmetry cells in one image

In practice, we may have multiple rectangular symmetry cells in different planes. Using the methods from above section, we can recover the pose of each cell up to a scale. However, the scales for different cells often are different. Therefore, we must take a further step to align the scales. For example, as shown in the left panel of Figure 2, each plane is recovered with the assumption that the distance from the camera center to the plane is d = 1. However, if we choose the reference plane to be the one on which cell q_1 resides with $d_1 = 1$, our goal is to find the distances from the camera center to the other two planes. Taking the plane on which cell q_2 resides, in order to find the distance d_2 , we can examine the intersection line L_{12} of the two planes. The length of L_{12} is recovered as $||L_{12}^1||$ in the reference plane and $||L_{12}^2||$ in the second plane. We then have the relation-ship $\frac{d_2}{d_1} = \frac{||L_{12}^1||}{||L_{12}^2||} = \alpha$. So the pose of the second symmetry cell is modified as $g_2 = (R_2, T_2) \leftarrow (R_2, \alpha T_2)$. The results are shown in the right panel of Figure 2.



Figure 2: Left. Recovered plane poses using symmetry cells before alignment. Right. Recovered plane poses after alignment. Now the axis are all in the same scale.

For planes without explicit intersection line, as long as

line segments with same length in 3-D space can be identified on each plane, the above scheme can also be used.

3 Symmetry-based plane reconstruction

In order to perform photo editing on the level of 3-D objects, we need to register these objects. In this paper, we only demonstrate this using planar object while our system is not limited to planar objects. To register a planar object, first we characterize the pose and location of the plane on which it reside; then we can define the object from the image and project it into 3-D space.

In order to characterize a plane, we only need to know its pose (normal vector N) and its distance d to the camera center. This can be accomplished by introducing a coordinate frame on the plane such that the z-axis is the normal vector. Then the pose and distance information are contained in the transformation g = (R, T) between this coordinate frame and the camera frame with the third column of R being the normal vector N and $T^T N = d$. Apparently, the choice of such coordinate system is not unique since the origin of the frame can be anywhere on the plane and the coordinate frame can rotate around the z-axis with arbitrary angle. The two components R and T of the transformation g can usually be determined separately. Since the pose N is only related to R and the distance d is mainly related to T, we can determine N and d in different steps.

3.1 Plane pose determination

There are basically two means of determining R and hence N. The first is a direct method using symmetry cells. By identifying a rectangular symmetry cell in the plane from the image and applying the method provided in previous section, the transformation g = (R, T) between the camera frame and the frame attached to the cell can be obtained. From Section 2.1 we know that R is a desired rotation and N is the third column of R.

The second method is to use the spatial relationship between the plane and other plane with known pose. This usually occurs in the case when it is difficult to identify a "good" symmetry cell on the plane or the spatial relationship is compelling and easy to apply. The spatial relationship considered here is symmetry relationship between planes; it includes reflective, translational and rotational symmetry. The commonly seen parallelism can be viewed as either reflective or translational symmetry while orthoganality is an example of rotational symmetry. In any case, the rotation R of the plane frame can be easily determined from the known plane having symmetry relationship with the plane in query.

3.2 Plane distance determination

Knowing *N*, we only need to solve *T* to obtain *d*. Here we discuss three ways of deciding *T*. Note that any point *p* on the 3-D plane can be chosen as the object frame origin and hence T = X with $X \in \mathbb{R}^3$ being the coordinate of *p* in the camera frame. Therefore, if the image of *p* is *x* in homogeneous coordinates, $T = \alpha x$ with $\alpha \in \mathbb{R}_+$ being the depth of *p*. Our first method is to directly determine α for a point with known 3-D location. If a point with image *x* on the intersection line between the plane in query and a plane with known pose and distance, its coordinate *X* in the camera frame can be determined using the knowledge about the second plane. Then $\alpha = \frac{X^T X}{X^T x}$ and $T = \alpha x$. The second method is to apply the alignment technique

The second method is to apply the alignment technique introduced in previous section to obtain α . By identifying one line segment on the plane of query and another line segment on a known plane with the understanding that these two line segments are of the same length in 3-D space, α can be solved using the alignment technique in Section 2.2.

Finally, if none of the above techniques can be applied but we have some knowledge about the spatial relationship between the plane in query and some unknown plane(s), we can use these knowledge to solve T. The useful knowledge are also the symmetry relationships. An example of reflective symmetry is illustrated in Figure 3. In any case, if the symmetry transformation between the desired plane and the known plane is g_s and the transformation between the camera frame and the known plane frame is g_r , then the desired transformation is $g = g_r g_s^{-1}$. So the key point is to find out the symmetry transformation g_8 in 3-D space. For reflective symmetry, this means to identify the reflective plane. For translational symmetry, this means to find out the translational vector in 3-D. And for rotational symmetry, we need to known the location of the rotational axis and the rotation angle. These information can be obtained from the known planes.

3.3 Object registration

When planes are defined, objects can be chosen by selecting polygons in the image. For any image x of a point in a plane with pose (R, T), its depth can be calculated as $\lambda = \frac{T^T N}{x^T N}$ where N is the the normal of the plane or the third column of the R. And the coordinate of the point in 3-D is $X = \lambda x$. Therefore, for any image polygon $S = \{x_1, \dots, x_n\}$, the corresponding 3-D object in the plane can be determined easily.

4 Photo editing

The above technique allows us to perform many operations on images. Most of these operations are related to



Figure 3: Obtain the transformation between a plane coordinate frame and the camera frame using known reflective symmetry.

the "copy-and-paste" function. That is, by identifying objects in the image, we can put them onto different locations, which is not limited to a single image. The symmetry-based techniques allow us to perform all such actions with our high-level knowledge.

4.1 Copy-and-paste within one image

Given a photo, many times we want to overwrite certain places with other objects or images. For example, we may want to eliminate some unwanted occlusion or shadow. Or we want to "copy" the image of an object from one place to another. While in commercial photo editing software such as Photoshop we can do this purely on image, it is usually hard to get the correct geometry due to perspective effects. However, with the knowledge of the scene and the registered 3-D objects, correct perspective image can be obtained.

The key point for the "copy-and-paste" is the following: for a given image point, find the image of its symmetry correspondence. To do this we need to specify the symmetry transformation g_s in 3-D space. For reflective and translational symmetries, this can be achieved by selecting only one pair of symmetric points on known planes. For rotational symmetry, we need also to point out the direction of rotation axis and rotation angles besides the pair of symmetric points. For any point x, first we can project it to 3-D space as X; then perform symmetry transformation on it to obtain its 3-D symmetric correspondence $g_s(X)$; finally we can obtain the corresponding image of $g_s(X)$.

So the "copy-and-paste" operation can be performed in three steps:

- 1. Define the source region (object) on an known plane;
- 2. Define the symmetry between the destiny and source

regions;

3. For each point in the destiny region, find its symmetric corresponding image point.

Here we show three examples involving this "copy-andpaste" function. As shown in Figure 4, we want to recover the occlusion caused by the lights, this is accomplished by simply "copying" the region above using the translational symmetry of the wall pattern. In Figure 5, we want to rerender the region with sunlight shadows, this is done by "copying" the region on the other side of the wall using the reflective symmetry between them. Besides removing unwanted objects or regions, we can also add virtual objects use the same technique. Figure 6 shows the added windows on both sides of the walls using both translational and reflective symmetry. The last demo is an example of extending current pictures. For example, if we extend the picture in Figure 7 to one, our scene knowledge tells us that the extension is just the translational copy of part of the wall. Therefore, by applying translational symmetry on the new regions, we can obtain the result in the right. Figure 8



Figure 4: Using multiple translation symmetry to remove the occlusion caused by lights.



Figure 5: Using reflective symmetry to re-render the areas with shadows and occlusion.

is an example of applying multiple copy-and-paste actions based on reflective and translational symmetry on an outdoor scene. The complicated foreground has been successfully removed and the window panels with reflections of the trees have been replaced by clean ones. This privodes a good basis for further graphical manipulation on the building image.



Figure 6: Using translational symmetry to "copy-andpaste" the areas with windows onto another blank area (the bottom row of the right side windows is the "copy" of the row above it).



Figure 7: Extending the picture to the right using translational symmetry (the image is extended to the right by 300 pixels).

4.2 Copy an image patch onto another image

The above techniques can also be extended to copy a patch in a plane of one image onto a plane in another image. This is done by aligning corresponding planes in different images. The aligning process includes three aspects:

- 1. Alignment of plane orientations. Aligning plane orientations means that the two coordinate frames on the two planes should be aligned in the every axis.
- 2. Alignment of scales. Aligning scales means that the corresponding object in the two planes should have the same size. This usually affects the scale of translation T of the second plane coordinate frame.
- 3. Alignment of corresponding points. Usually one pair of corresponding points are enough.



Figure 8: Using multiple symmetry to clear the foreground of the building as well as reflections of the trees on the window panels.

The above three steps are for the case of calibrated cameras. For the case of uncalibrated camera, it is necessary to identify all four corresponding vertices of the corresponding rectangle region. Figure **9** is an example of "copy-and-paste" between two calibrated images. While Figure **10** shows "copy-and-paste" with an uncalibrated picture.

Finally, Figure **11** shows a comprehensive example of above operations. A long Chinese caligraphy strip is folded correctly on the two adjacent walls; and windows on the left side are "overwritten" by windows from the right using reflective symmetry.

4.3 Image mosaicing

Generating panorama or image mosaicing from multiple images is always an interesting problem in computer graphics. Traditional approach usually requires the images are taken with camera center fixed. However, by applying knowledge about planes in the scene and using symmetry cells to align the planes, we can piece together different images taken at different viewpoints. The key point is to use the corresponding symmetry cell to align the two images of the same plane using the alignment steps in the previous



Figure 9: Paste the painting "Creation" on the wall of in the indoor picture. The calibration information about the left bottom painting is known.



Figure 10: Combining the two photos in the left. The calibration information about the photo on the left bottom is not available.

section. The "byproduct" is to recover the transformation between the two cameras. As shown in Figure **12**, by aligning symmetry cell from different images, we can obtain the camera motion and then combining these two pictures.

4.4 User interface

The photo editing software is part of our effort on a computer vision related application package. For test purposes, a user interface based on Matlab has been developed. User can use mouse to click on the necessary points as input. The visual effect depends partially on the precision of user input. While it is desired to have precise input, subpixel precision is not required. All the results shown in this paper are running in Matlab on a 850MHz CPU laptop without any optimization of code. Most of the running time is due to the rendering process, which ranges from several seconds to less than three minutes depending on the sizes of the image and patches.



Figure 11: A comprehensive example of various operations on the original image of Figure 2.

5 Discussion

In this paper, we have demonstrated several applications in photo editing by applying high-level knowledge and using symmetry-based techniques. As shown in Figure **11** and Figure **12**, the symmetry-based approach admits a much broader range of applications than traditional approaches. With the characterization of symmetry cells, we can generate panoramas without fixing camera center. With symmetry-based matching techniques [9], it is possible to build an automatic matching and reconstruction system. The symmetry-based approach displays the power of highlevel knowledge in motion analysis and structure recovery.

Despite its advantage, applying high-level knowledge also provides us some new challenges. First, we need to know how to effectively represent and compute the knowledge. This paper gives an example of representing the knowledge of the existence of rectangles in 3-D space. The characterization of general symmetric shapes can be performed similarly. Secondly, we need to know how to integrate results obtained from different types of knowledge. For each image, when the spatial relationships between objects are taken into account, the objects can no longer be treated individually. Then the reconstruction process needs to find an "optimal" solution that is compatible with assumptions on the shapes of individual objects as well as their relationships. In Figure 2, we dealt this problem by performing alignment of the symmetry cells with respect to a reference cell. If more symmetry cells were involved, we would have to balance the alignment process by considering all adjacent symmetry cells. Last but not the least, in the process of incorporating knowledge in any new applications, we want to identify which part can be computed automatically by the machine and how much manual intervention is really needed. For the photo editing process, we



Figure 12: Joining the two pictures on the top using corresponding symmetry cells (in this case windows) on the front side of the building. The middle picture is the bird view of the recovered 3-D shape of the two sides of the building and the camera orientations (the two co-ordinate frames).

point out the minimal amount of input required from the user for different actions. In general, such minimal requirement might not be unique and should be taken into account in designing user interfaces.

Besides above challenges, there are also many graphics problems to be solved in the future for this photo editing system. For example, some graphical artifacts arise in the process of "copy-and-paste." Although traditional image processing techniques can be applied, it is also possible to process the image by incorporating the correct 3-D geometry. For instance, at the boundary between the copied region and the original region, it is better to filter (blur) along the direction of the vanishing point. Other problems such as how to combine results from multiple symmetry cues are also an on-going research topic.

References

- A.R.J.Francois, G.G.Medioni, and R.Waupotitsch. Reconstructing mirror symmetric scenes from a single view using 2-view stereo geometry. In *Proceedings of International Conference on Pattern Recognition*, 2002.
- [2] I.Reid A. Criminisi and A. Zisserman. Single view metrology. Int. Journal on Computer Vision, 2000.
- [3] S. Carlsson. Symmetry in perspective. In *Proceedings of European Conference on Computer Vision*, pages 249–263, 1998.
- [4] P. Debevec, C.J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry and image-based approach. In *Proceedings of SIGGRAPH'96*, 1996.
- [5] A. Francois, G. Medioni, and R. Waupotitsch. Rconstructing mirror symmetric scenes from a single view using 2-view stereo geometry. In *Proceedings of Int. Conference on Pattern Recognition*, 2002.
- [6] L. Van Gool, T. Moons, and M. Proesmans. Mirror and point symmetry under perspective skewing. In *Proceedings of Int. Conference on Computer Vision & Pattern Recognition*, San Francisco, USA, 1996.
- [7] W. Hong, A. Y. Yang, and Y. Ma. On symmetry and multiple view geometry: Structure, pose and calibration from a single image. *Int. Journal on Computer Vision*, submitted 2002.
- [8] W. Hong. Geometry and reconstruction from spatial symmetry. *Master Thesis, UIUC*, July 2003.
- [9] K. Huang, A. Y. Yang, W. Hong, and Y. Ma. Symmetrybased 3-D reconstruction from perspective images (part II): Matching and reconstruction. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, submitted 2003.
- [10] D. Liebowitz and A. Zisserman. Detecting rotational symmetries using normalized convolution. In *Proceedings of Int. Conference on Computer Vision & Pattern Recognition*, pages 482–488, 1998.
- [11] H. Mitsumoto, S. Tamura, K. Okazaki, and Y. Fukui. 3-D reconstruction using mirror images based on a plane symmetry recovering method. *IEEE Transactions on Pattern Analysis* & *Machine Intelligence*, 14(9):941–946, 1992.
- [12] C. A. Rothwell, D. A. Forsyth, A. Zisserman, and J. L. Mundy. Extracting projective structure from single perspective views of 3D point sets. In *Proceedings of IEEE International Conference on Computer Vision*, pages 573–582, 1993.
- [13] D. Svedberg and S. Carlsson. Calibration, pose and novel views from single images of constrained scenes. In *the 11th Scandinavian Conference on Image Analysis*, pages 111– 117, 1999.
- [14] A.Y. Yang, W. Hong, and Y. Ma. Structure and pose from single images of symmetric objects with applications in robot navigationi. In *Proceedings of International Conference on Robotics and Automation*, Taipei, 2003.

- [15] A. Y. Yang, S. Rao, W. Hong, and Y. Ma. Geometric segmentation of perspective images based on symmetry groups. In *Proceedings of IEEE International Conference on Computer Vision*, Nice, France, 2003.
- [16] H. Zabrodsky, S. Peleg, and D. Avnir. Symmetry as a continuous feature. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 17(12):1154–1166, 1995.
- [17] H. Zabrodsky and D. Weinshall. Using bilateral symmetry to improve 3D reconstruction from image sequences. *Comp. Vision and Image Understanding*, 67:48–57, 1997.
- [18] A. Zisserman, D. P. Mukherjee, and J. M. Brady. Shape from symmetry-detecting and exploiting symmetry in affine images. *Phil. Trans. Royal Soc. London A*, 351:77–106, 1995.
- [19] A. Zisserman, C. A. Rothwell, D. A. Forsyth, and J. L. Mundy. Extracting projective structure from single perspective views of 3d point sets. In *Proceedings of IEEE International Conference on Computer Vision*, 1993.