PERGAMON

Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

Pattern Recognition 38 (2005) 825–834

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Symmetry-based photo-editing☆

Kun Huang[a,*], Wei Hong[b], Yi Ma[b]

[a]*Department of Biomedical Informatics, Ohio State University, Room 3172A, Graves Hall, 333 West 10th Avenue, Columbus, OH 43210, USA*
[b]*Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, 1308 West Main Street, Urbana, IL 61801, USA*

Received 9 August 2004

Abstract

In this paper, we demonstrate how to edit digital photos based on the understanding of high-level geometric knowledge imposed upon objects in the photos. This is achieved by the correct recovery of the 3-D shape and relationships of the objects, without explicitly performing full 3-D reconstruction. Symmetry is proposed as the central notion that unifies both conceptually and algorithmically all types of geometric regularities such as parallelism, orthogonality, and similarity. The methods are extremely simple, accurate, easy to implement, and they demonstrate the utility of applying scene knowledge to image understanding and editing.

Keywords: Image understanding; High-level knowledge; Structure from symmetry; Photo-editing; 3-D scene recovery

1. Introduction

Photo-editing has become an important part of digital photography for achieving high photo quality or special graphic effects. In most of the existing popular photo-editing packages, such as Adobe Photoshop, the photos are manipulated at the level of pixels in regions or layers. However, it is often desirable to edit the photos in such a way that geometry and perspective effects are preserved. Examples of such tasks include the removal of undesired reflections from a building facade and the addition of a logo on the ground for a football game scene (watermarks). In most common photo-editing softwares, there is no interface to aid in the preservation of geometry and perspective. These effects are usually achieved by manual and intuitive adjustments which are tedious and require a lot of experience. In this paper, we introduce a set of interactive symmetry-based techniques for editing digital photographs. These symmetry-based algorithms enable us to manipulate 2-D image regions by understanding their correct shapes and relationships in the 3-D space and, therefore, to preserve the scene geometry with a minimal amount of manual intervention.

The recovery of 3-D shapes from images is a classic problem in computer vision. The common approach is to perform 3-D reconstruction from multiple images using structure-from-motion (SFM) techniques. This line of work has led to the development of multiple view geometry. Classic multiple view geometry typically does not take advantage of any external knowledge about the scene geometry. Only image primitives such as points, lines, and planes are used for reconstruction purposes, and no knowledge about their spatial relationships is assumed. However, recently more and more work has taken advantage of scene knowledge in reconstruction [1–4]. While various types of scene knowledge

☆ This work is partially supported by DARPA, the startup funding from the Department of Electrical and Computer Engineering in University of Illinois at Urbana-Champaign and the startup funding from the Department of Biomedical Informatics in Ohio State University.

* Corresponding author. Tel.: +1 614 2921084; fax: +1 614 6886600.
E-mail address: khuang@bmi.osu.edu (K. Huang).
URL: http://www.bmi.osu.edu/∼khuang/index.html.

0031-3203/$30.00 © 2004 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2004.11.014

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 structure, including patterns, parallelism, and orthogonality, is common in the man-made environments. It provides useful cues in retrieving shapes of objects and spatial relationships between objects from images. 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. For instance, when we apply the knowledge that one object is rectangular in 3-D, its pose and size are automatically determined from its image. There are many new functionalities and applications that would be very difficult without applying such scene knowledge.

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 tend to capture our attention more than the others. For spatial relationships between objects, parallelism, orthogonality, and similarity are the most conspicuous ones. Interestingly, all such regularities are encompassed by the notion of *symmetry*. For instance, a rectangular window has one rotational symmetry (by 180°) and two reflective symmetries; identical 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 [4–12].

Recently, a set of algorithms using symmetry for reconstruction from a single image or multiple images has been developed [4], which leads to further studies on geometric segmentation [11] and large baseline matching [12]. In Ref. [11], 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. Symmetry cells found in different images can also be used as high-level features for matching purposes [12], which is critical for calculating camera motion across the images. With known camera motions, 3-D reconstruction is efficient and accurate. These are examples of the utilization of high-level geometric knowledge in modeling and motion analysis. In this paper, we will extend these techniques to another application: photo-editing.

1.1. Relation to prior work

Various techniques have been applied to photo-editing. For example, Oh et al. [13] uses a mixture of geometry and layers to obtain geometric relationship between objects. In Ref. [14], the authors exploited the line segments extracted from the image for photo-editing. Our work is also related to the image-based modeling problems. In Ref. [15], the authors used a series of regular shapes to reconstruct the building structure based on high-level knowledge. Furthermore, Horry et al. [16] developed a technique called "tour into picture", which utilized the geometric relationship of planes. The effects of symmetry in 3-D reconstruction and recognition have been studied extensively. It led to the single-view geometry of symmetry. In Ref. [17], the authors studied how to reconstruct a 3-D object using reflective symmetry induced by a mirror. Van Gool et al. [5] studied reflective and rotational symmetries under perspective projection. Some more recent applications and studies can be found in Refs. [18,1,6]. Svedberg and Carlsson [3], Liebowitz and Zisserman [19] and Criminisi et al. [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, Rothwell et al. [9] 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 Ref. [7]. Recently, Hong et al. [4] has systematically studied reconstruction from symmetry. Its implication on geometry-based image segmentation and large-baseline feature matching is given in Refs. [11,12]. As a continuation, this paper is to show some possible applications of utilizing symmetry knowledge about the scene.

2. Symmetry-based cell reconstruction

Here we briefly introduce some techniques for 3-D pose and shape recovery using knowledge about symmetry. To use the symmetry-based algorithm, we start from images of the basic symmetric objects called *symmetry cells*. While a symmetry cell can be the image of any symmetric object, we just use one of the simplest symmetric objects, a rectangle, to illustrate the reconstruction process. Once a (rectangular) symmetry cell in a plane has been 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 us look at the recovery of the 3-D pose of a plane using symmetry cells. The 3-D reconstruction of symmetric objects from a single image has been thoroughly studied in Ref. [4]. 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 Fig. 1.

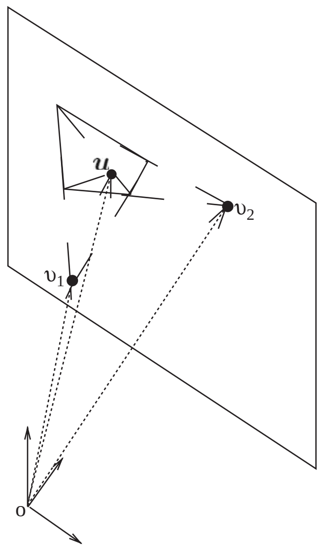

Fig. 1. Image formation for a rectangle. $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ are two vanishing points. $\boldsymbol{u}$ is the intersection of the diagonal of the four-sided polygon and is the image of the rectangle center.

A vanishing point $\boldsymbol{v}=[x, y, z]^{\mathrm{T}} \in \mathbb{R}^3$, 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:

$$\boldsymbol{v}_1^{\mathrm{T}}\boldsymbol{v}_2 = 0. \quad (1)$$

In addition, the normal vector $N \in \mathbb{R}^3$ of rectangle plane can be obtained by $N \sim \boldsymbol{v}_1 \times \boldsymbol{v}_2$, where $\sim$ means equality up to a scalar factor.

We can 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 axes parallel to the two pairs of edges of the rectangle. Thus our goal is to determine the transformation $g = (R, T) \in SE(3)$ between the camera frame and the object frame. Here $R \in SO(3)$ is the rotation and $T \in \mathbb{R}^3$ is the translation. Notice 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 R may 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 $R = U\Sigma V^{\mathrm{T}}$ with $U, V \in O(3)$. Then the rotation is

$$R = UV^{\mathrm{T}}.$$

To fix the scale in T, we typically choose the distance d from the camera center to the rectangle plane to be one, which means that $T = \alpha\boldsymbol{u}$ with

$$\alpha = \frac{d}{\boldsymbol{u}^{\mathrm{T}}N} = \frac{1}{\boldsymbol{u}^{\mathrm{T}}N}.$$

Notice that when the plane orientation N and distance d are determined, the shape (aspect ratio) of the symmetry cell (rectangle) can also be fixed up to a scale. The above method has been applied in our other works such as symmetry cell detection and matching [11,12]. In these works, the errors for orientation estimation are within the range of 1.5° and the errors for shape estimation are within 1%.

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 Section 2.1, 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 Fig. 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 the 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 the cell q_2 resides, in order to find the distance d_2, we can examine the intersection line L_{12} of the two planes as shown in Fig. 2.[2] 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 relationship

$$\frac{d_2}{d_1} = \frac{\|L_{12}^1\|}{\|L_{12}^2\|} = \alpha.$$

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 Fig. 2.

For planes without an 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. We demonstrate

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

[2] In theory, a single point on the intersection line is enough to determine the distance d_2. Here we use a line segment to make the estimation more robust.

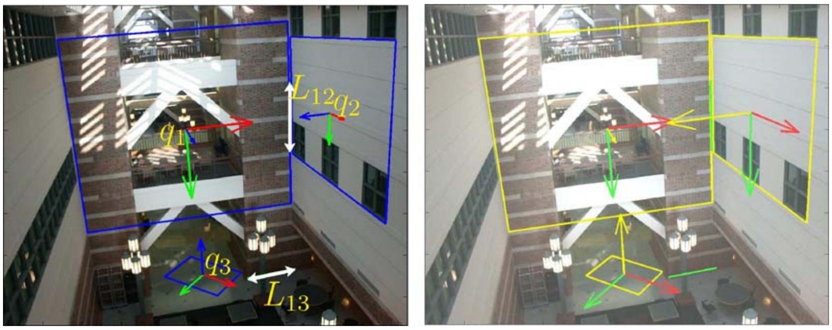

Fig. 2. Left: Recovered plane poses using three symmetry cells q_1, q_2, and q_3 before alignment. L_{12} is a line segment on the intersection between the planes of cells q_1 and q_2. L_{13} is on the intersection between the planes of q_1 and q_3. Right: Recovered plane poses after alignment. Now the axes are all in the same scale.

this using planar objects. However, our system is not limited to just planar objects. To register a planar object, first we characterize the pose and location of the plane on which it resides; then we can define the object from the image and project it into the 3-D space.

In order to characterize a plane, we need only 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 $T^{\mathrm{T}}N = d$. The choice of this 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 Section 2, 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 use. 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 orthogonality 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 from the simple fact

$$d = T^{\mathrm{T}}N. \tag{2}$$

Here we discuss three ways of determining T. Notice 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 $\boldsymbol{x}$ in homogeneous coordinates, $T = \alpha\boldsymbol{x}$ 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 $\boldsymbol{x}$ 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 $\alpha = X^{\mathrm{T}}X/X^{\mathrm{T}}\boldsymbol{x}$ and $T = \alpha\boldsymbol{x}$.

The second method is to apply the alignment technique introduced in Section 2 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 question and some unknown plane(s), we can use this knowledge to solve T. Symmetry relationships can also be useful. An example of reflective symmetry is illustrated in Fig. 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

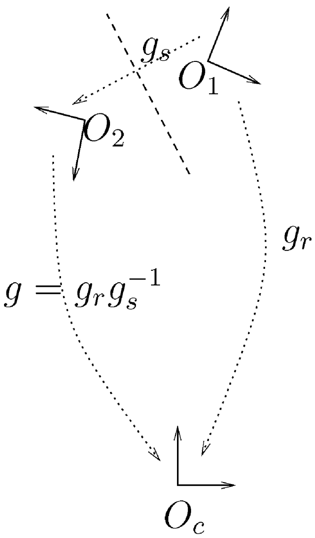

Fig. 3. Obtain the transformation g between a plane coordinate frame O_2 and the camera frame O_c using known reflective symmetry g_s and the known transformation g_r between the plane frame O_1 and the camera frame O_c.

transformation g_s 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. As for rotational symmetry, we need to know the location of the rotational axis and the rotation angle. This information can be obtained from the known planes.

3.3. Object registration

Once planes are defined, objects can be chosen by selecting polygons in the image. For any image $\boldsymbol{x}$ of a point in a plane with pose (R, T), its depth can be calculated as

$$\lambda = \frac{d}{\boldsymbol{x}^{\mathrm{T}} N} = \frac{T^{\mathrm{T}} N}{\boldsymbol{x}^{\mathrm{T}} N},$$

where N is the normal of the plane or the third column of the R, and d is the distance defined in Section 2. The coordinate of the point in 3-D is then $X = \lambda \boldsymbol{x}$. Therefore, for any image polygon $S = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, the shape of the corresponding 3-D object 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 the "copy-and-paste" function. That is, by identifying objects in the image, we can move them into different locations.

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, a shadow, or 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 in the image plane, 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 $\boldsymbol{x}$, first, we can project it to a 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 destination and source regions;
(3) for each point in the destination region, find its symmetric corresponding image point.

Here we show three examples involving this "copy-and-paste" function. In Fig. 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 Fig. 5, we want to re-render 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. Fig. 6 shows the added windows on one side of the walls using translational symmetry. The last demo is an example of extending current pictures. For example, if we extend the picture in Fig. 7 to one side, our scene knowledge tells us that the extension is just the translational copy of part of the wall. Therefore, by applying translational symmetry to the new regions, we can obtain the result on the right.

Fig. 8 is an example of taking 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 provides a good basis for further graphical manipulation on the building image.

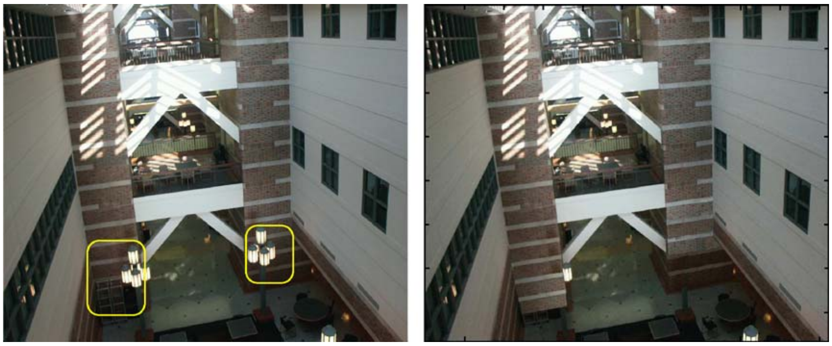

Fig. 4. Using multiple translation symmetry to remove the occlusion caused by lights.

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

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

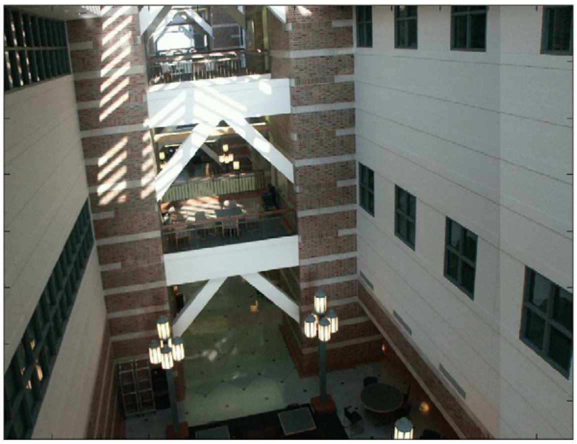

Fig. 7. Extending the picture to the right using translational symmetry (the image is extended to the right by 300 pixels).

Finally, Fig. 11 shows a comprehensive example of above operations. In addition to the results displayed in the Figs. 4–7, windows on the left side are "overwritten" by windows from the right using reflective symmetry. From these images, we can see that the algorithm correctly determines the geometry in all the actions.

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 the corresponding planes in different images. The alignment process includes three

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

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

Fig. 10. Combine the two photos in the left. The calibration information about the photo on the left bottom is not available.

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 is enough.

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. Fig. 9 shows an example of "copy-and-paste" between two calibrated images. While Fig. 10 shows "copy-and-paste" with an uncalibrated picture. Besides the above examples, Fig. 11 shows how to add a long Chinese calligraphy strip to the image, which is correctly folded on the two adjacent walls.

4.3. Image mosaicing

Generating panorama or image mosaicing from multiple images is always an interesting problem in computer graphics. The traditional approach usually requires that the images are taken with a fixed camera center. 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 cells to align the two images of the same plane using the alignment steps in Section 2. A byproduct is that we also recover the transformation between the two cameras. As shown in Fig. 12, we can obtain the camera motion and combine these two pictures by aligning symmetry cell from different images.

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. Users can use a mouse to click on the necessary points as input. The visual effect depends partially on the precision of user input. While it is desirable to have input as precise as possible, subpixel precision is not required in practice. If small error is present in the user input, our approach can be a very effective initialization step before any fine tuning. In the copy-and-paste action, every pixel in the target region is mapped to a point in the origin region and the pixel closest to that point is copied to the target pixel without using any interpolation technique. All the results shown in this paper are running in MATLAB on a 850 MHz CPU laptop without any optimization of code. Most of the running time is consumed by the rendering process, which ranges from a few seconds to about 3 min depending on the size of the image and the patches being transformed.

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

5. Conclusion and discussion

In this paper, we have applied symmetry-based 3-D reconstruction techniques to photo-editing applications. As demonstrated in Figs. 11 and 12, the symmetry-based approach applies to a much broader range of applications than the traditional approaches. By characterizing symmetry cells, we can generate panoramas without having to fix the camera center. With symmetry-based matching techniques [12], it is possible to build an automatic matching and reconstruction system. The symmetry-based approach exemplifies the power of high-level knowledge in motion analysis and structure recovery.

Despite its advantages, application of high-level knowledge also provides us with some new challenges. First, we need to know how to effectively represent and process the knowledge computationally. This paper illustrates how to represent knowledge about rectangles in 3-D space. The characterization of general symmetric shapes can similarly be performed. Second, we need to know how to integrate results obtained from different types of knowledge. For instance, when the spatial relationships between objects in the image are taken into account, the objects can no longer be treated individually. In this case 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 Fig. 2, we dealt with this problem by performing alignment of the symmetry cells with respect to a single reference cell. If more symmetry cells were involved, we could improve 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 point out one minimal set of input required from the user for different actions. In general, the minimal input required will not be unique and this should be taken into account in designing user interfaces, i.e. by giving the user multiple input options with sensible defaults.

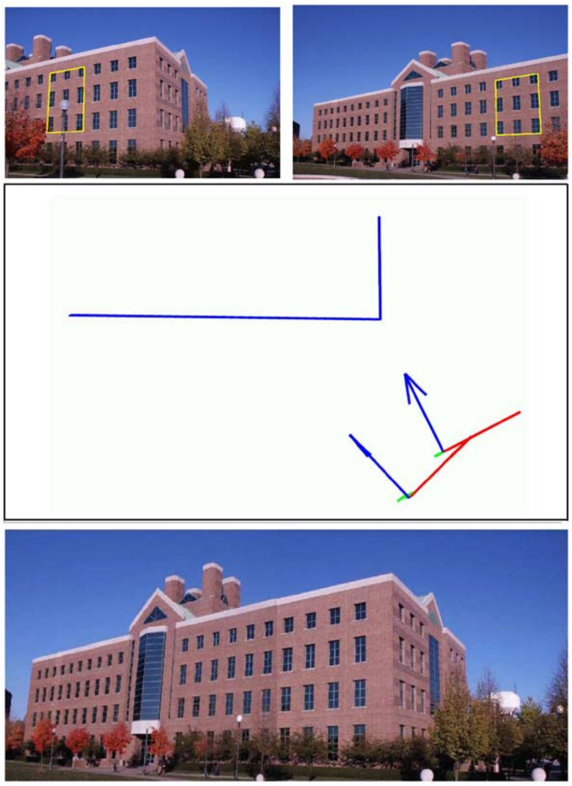

Fig. 12. Join the two pictures on the top using corresponding symmetry cells (in this case windows) on the front side of the building. The middle picture is a bird's eye view of the 3-D shape of the two sides of the building and the camera orientations (the two coordinate frames) as recovered by the algorithm. The two coordinate frames are the two camera frames. The two perpendicular lines are the two sides of the building.

Besides the above challenges, there are many graphical issues to be resolved for the photo-editing tasks in order to improve the visual effects. In this paper, we have focused on obtaining the correct geometry and perspective effects in photo-editing. As shown in the experiments, the results obtained using the our methods are visually satisfactory from a geometric point of view. As we have indicated above, our methods can also be used as a initialization step for further refinement if error and noise are present in the system. Nevertheless, the visual effects are also significantly affected by other factors such as lighting, shading, and texture. For example, the shadow of an object can affect our interpretation on the 3-D position of the object. Thus it is possible to use our knowledge in both geometry and lighting to enhance the visual effects. In addition, 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. Therefore, an important future direction of our research is to integrate the geometric knowledge with the graphics techniques in our system.

References

[1] H. Zabrodsky, D. Weinshall, Using bilateral symmetry to improve 3D reconstruction from image sequences, Comput. Vision Image Understand. 67 (1997) 48–57.

[2] A. Criminisi, I. Reid, A. Zisserman, Single view metrology, Int. J. Comput. Vision (2000).

[3] D. Svedberg, S. Carlsson, Calibration, pose and novel views from single images of constrained scenes, in: Proceedings of the 11th Scandinavian Conference on Image Analysis, 1999, pp. 111–117.

[4] W. Hong, A.Y. Yang, K. Huang, Y. Ma, On symmetry and multiple view geometry: structure, pose and calibration from a single image, Int. J. Comput. Vision 60 (3) (2004) 241–265.

[5] L. Van Gool, T. Moons, M. Proesmans, Mirror and point symmetry under perspective skewing, in: Proceedings of the International Conference on Computer Vision & Pattern Recognition, San Francisco, USA, 1996.

[6] A. Francois, G. Medioni, R. Waupotitsch, Reconstructing mirror symmetric scenes from a single view using 2-view stereo geometry, in: Proceedings of the International Conference on Pattern Recognition, 2002.

[7] S. Carlsson, Symmetry in perspective, in: Proceedings of the European Conference on Computer Vision, 1998, pp. 249–263.

[8] A. Zisserman, D.P. Mukherjee, J.M. Brady, Shape from symmetry-detecting and exploiting symmetry in affine images, Philos. Trans. R. Soc. London A 351 (1995) 77–106.

[9] C.A. Rothwell, D.A. Forsyth, A. Zisserman, J.L. Mundy, Extracting projective structure from single perspective views of 3D point sets, in: Proceedings of the IEEE International Conference on Computer Vision, 1993, pp. 573–582.

[10] A.Y. Yang, W. Hong, Y. Ma, Structure and pose from single images of symmetric objects with applications in robot navigation, in: Proceedings of the International Conference on Robotics and Automation, Taipei, 2003.

[11] A.Y. Yang, S. Rao, W. Hong, Y. Ma, Geometric segmentation of perspective images based on symmetry groups, in: Proceedings of the IEEE International Conference on Computer Vision, Nice, France, 2003.

[12] K. Huang, A.Y. Yang, W. Hong, Y. Ma, Large-baseline matching and reconstruction using symmetry cells, in: Proceedings of the IEEE International Conference on Robotics & Automation, 2004.

[13] B.M. Oh, M. Chen, J. Dorsey, F. Durand, Image-based modeling and photo editing, in: Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH), 2001.

[14] J.H. Elder, R.M. Goldberg, Image editing in the contour domain, IEEE Trans. Pattern Anal. Mach. Intell. 23 (3) (2001) 291–296.

[15] P. Debevec, C.J. Taylor, J. Malik, Modeling and rendering architecture from photographs: a hybrid geometry and image-based approach, in: Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH), 1996.

[16] Y. Horry, K.-I. Anjyo, K. Arai, Tour into the picture: using a spidery mesh interface to make animation from a single image, in: Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH), 1997.

[17] H. Mitsumoto, S. Tamura, K. Okazaki, Y. Fukui, 3-D reconstruction using mirror images based on a plane symmetry recovering method, IEEE Trans. Pattern Anal. Mach. Intell. 14 (9) (1992) 941–946.

[18] H. Zabrodsky, S. Peleg, D. Avnir, Symmetry as a continuous feature, IEEE Trans. Pattern Anal. Mach. Intell. 17 (12) (1995) 1154–1166.

[19] D. Liebowitz, A. Zisserman, Detecting rotational symmetries using normalized convolution, in: Proceedings of the International Conference on Computer Vision & Pattern Recognition, 1998, pp. 482–488.

About the Author—KUN HUANG received the two Bachelors' degree in Biology and Computer Science from Tsinghua University, Beijing, China, in 1996. He then studied in the University of Illinois at Urbana-Champaign (UIUC) from 1996 to 2004, where he obtained the Master of Science degree in Physiology in 1998, the Master of Science degree in Electrical Engineering in 2000, the Master of Science degree in Mathematics in 2002, and the Ph.D. degree in Electrical and Computer Engineering. Currently he is an assistant professor in the Department of Biomedical Informatics in Ohio State University. His research interests include computer vision, machine learning, control theory, robotics, and computational biology.

About the Author—WEI HONG received the Bachelor degree in Information Science and Technology from Xi'an Jiaotong University, Xi'an, China, in 1998. He has been studying in UIUC since 2001, where he obtained the Master of Science degree in Electrical Engineering in 2003. Currently he is a Ph.D. candidate in Electrical and Computer Engineering at UIUC. His research interests include computer vision, computer graphics, and machine learning.

About the Author—YI MA received the two Bachelors' degree in Automation and Applied Mathematics from Tsinghua University, Beijing, China, in 1995. He received the Master of Science degree in Electrical Engineering and Computer Sciences in 1997, the Master of Arts degree in Mathematics in 2000, and the Ph.D. degree in Electrical Engineering and Computer Sciences in 2000, all from the University of

California at Berkeley. Upon graduation in August 2000, he joined the faculty of the Department of Electrical and Computer Engineering, UIUC as an assistant professor. He also serves as a research assistant professor both in the Decision & Control Group of the Coordinated Science Laboratory and in the Image Formation & Processing Group of the Beckman Institute on the same campus. His research interest includes systems theory and computer vision. Recent research topics include geometric and algebraic theory of computer vision, vision based control and navigation, generalized principal component analysis, and hybrid systems theory.