

Large-Baseline Matching and Reconstruction from Symmetry Cells*

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

Abstract—In this paper, we study how the presence of symmetry in man-made environments may significantly facilitate the task of automatic matching features and recovering 3-D camera pose and scene structure from multiple perspective images. While conventional methods typically rely on small-motion tracking or robust statistic techniques to resolve the coupling between feature matching and 3-D recovery, we here propose a new symmetry-based approach which allows automatic feature matching between images taken with arbitrary (both large and small) camera motions. To this end, we develop the multiple-view geometry of symmetry cells. To resolve possible ambiguities that may arise in matching symmetry cells and camera pose recovery, we find a consistent solution by finding the maximal complete subgraph of a matching graph; we also use a topological check to avoid mismatches. As our experiments will show, the resulting algorithms are simple, accurate and easy to implement.

Index Terms—large-baseline feature matching, 3-D reconstruction, symmetry-based reconstruction, epipolar constraint, homography group.

I. INTRODUCTION

Automatic feature matching across multiple images of the same scene is a critical issue in many machine vision problems such as automatic 3-D reconstruction from multiple images. Traditionally, point features such as corners have been widely used for matching [11], [14]. However, in most situations, good matching results can only be achieved under small-baseline motions, i.e. the translation of the camera is short. The problem with small baseline is that the 3-D structures and motion cannot be accurately recovered due to the small signal-to-noise ratio (SNR). If large-baseline is to be used, some additional geometric constraints such as epipolar geometry need to be applied [7]. These algorithms often exploit iterative robust statistics such as RANSAC [5] and LMeds [19], which usually require good initialization and long time. Another approach for large-baseline matching is to use affine invariants [4]. Instead of using points, this approach chooses special regions as the feature. It is assumed that corresponding regions in different images are invariant up to an affine transformation. Establishing affine transformations for all possible regions can be time consuming.

In this paper, we show that symmetric objects in the scene can be used for matching with fast and accurate results for both large and small baselines. Regular shapes such as rectangles have been widely used in applications such as robot mapping and

navigation [9]. The fact that these regular shapes are conspicuous is largely due to the symmetry of these shapes. It has long been known that symmetry can facilitate 3-D reconstruction and recognition; especially the single-view geometry of symmetry has been extensively studied such as in references [3], [8], [12] studied how to reconstruct a 3-D object using reflective symmetry induced by a mirror. [10] studied geometric symmetry in robotics from a group theoretical point of view. Reflective and rotational symmetries under perspective projection were also studied by [15], and some more recent applications and studies can be found in [18], [6]. In 3-D object and pose recognition, [13] pointed out that the assumption of reflective symmetry can also be used in the construction of projective invariants. These invariants can also be formulated using Grassman-Cayley algebra, as pointed out in [2]. Recently, [16], [8] use symmetry group and multiple-view geometry to systematically designed a series reconstruction algorithms for all three types of symmetries based the notion of multiple equivalent views hidden in the single image of a symmetric object.

Contribution of this paper. Our previous work has shown it is possible to detect and extract symmetry cells from each image based on image segmentation and symmetry testing [16], [17]. In this paper, we will use symmetry cells as the new basic features (instead of points or lines) to study the associated matching and recovery problems.

- When multiple images of multiple symmetry cells are given, the combinatorial relations among all possible solutions are not well understood. In this paper, we develop a much needed multiple-view geometry for symmetry cells and provide a precise characterization of possible ambiguities in the solution.
- To automatically resolve the ambiguities that may arise from cell matching, we introduce a consistency measure between possible matches. We show that the problem of obtaining a consistent 3-D reconstruction is equivalent to the problem of finding a complete subgraph of a consistent matching graph. A complete algorithm for solving this problem is given.
- Based on the proposed theory and algorithms, we develop a prototype system that is able to automatically extract and match simple symmetric objects (e.g., rectangles and squares) from multiple images and return a consistent solution to all the camera poses and 3-D structure of the symmetric objects, despite large or small baselines between

*This material is based upon work partially supported by the U.S. Army Research Office under Contract DAAD19-00-1-0466 and UIUC ECE department startup fund. Any opinions, findings, and conclusions are those of the authors and do not necessarily reflect the views of the above agencies.

the views. A check on local topology is exploited to avoid certain mismatches.

Experiments show that our algorithms work remarkably well in situations for which the conventional approaches would fail. The recovered camera poses and scene structure are very accurate, usually within a 1 ~ 2 percent error to the ground truth without any nonlinear optimization.

II. MULTIPLE-VIEW GEOMETRY OF SYMMETRY CELLS

Although single-view geometry of any type of symmetry, i.e. reflective, rotational, and translational, is well studied (e.g., see [8]), less is known about the geometry related to multiple images of multiple symmetric objects which is typically the case in a man-made environment. It is clear that constraints satisfied by images of symmetric objects should be much richer than those for points and lines since images of a symmetric object contain extra 3-D information that images of a point or line do not. In this section, we characterize these extra constraints, which will be useful for matching and reconstructing symmetric objects.

Without loss of generality, in this paper we focus on a particular class of symmetric objects: rectangles. Rectangles are planar objects whose recognition, extraction, and reconstruction can be much simplified. They are also ubiquitous in man-made environments and hence easy to demonstrate the effectiveness of our approach. For simplicity, we call a rectangle a *symmetry cell*. The symmetry group G of a cell, as shown in Figure 1, is the

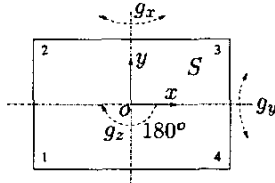


Fig. 1. A rectangular cell S whose symmetry includes the identity g_e , reflections g_x and g_y along the x and y axes, respectively, and a rotation g_z about the z -axis by 180° . These transformations forms a dihedral group of order 2, i.e. D_2 .

dihedral group D_2 of order 2. If the cell happens to be a square, the symmetry group G is enlarged to the dihedral group D_4 of order 4.

Now consider m perspective images $I_i(S)$ of S are taken by a camera at vantage points (or poses) $(R_i, T_i) \in SE(3), i = 1, 2, \dots, m$, respectively, as shown in Figure 2. Typically,

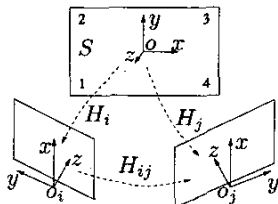


Fig. 2. Multiple views of a rectangular symmetry cell.

(R_i, T_i) is represented with respect to the object frame centered at the rectangle. Since S is planar, there exists a homography $H_i = R_i + \frac{1}{d_i} T_i N_i^T \in \mathbb{R}^{3 \times 3}$ which maps the plane $P \supseteq S$ in

space to the i^{th} image plane. Geometrically, $N_i \in \mathbb{R}^3$ is the unit normal vector of the plane and $d_i \in \mathbb{R}_+$ is its distance to the center of the i^{th} camera frame. The homography between the i^{th} and j^{th} views is denoted by $H_{ij} \doteq H_j H_i^{-1}$.

Since S has a symmetry group $G = \{g_e, g_x, g_y, g_z\}$, we know that each image of S is equivalent to four images and the homography matrices between these equivalent views are

$$H'_i(g) \doteq H_i g H_i^{-1}, \quad g \in G, i = 1, 2, \dots, m.$$

For each i , the matrices H'_i form a homography group $G'_i = H_i G H_i^{-1}$ that is conjugate to G . Given g and $H'_i(g)$, both H_i and H_{ij} can be determined from them by solving the following set of (Lyapunov type) linear equations:

$$H'_i(g) H_i - H_i g = 0, \quad (1)$$

$$H_{ij} H'_i(g) - H'_j(g) H_{ij} = 0. \quad (2)$$

The benefit of symmetry then becomes clear: the relative poses between different views or the pose between each view and the object can be recovered from information that is encoded in individual images – no point matching across different images is needed.

However, some ambiguity remains in the above equations. It is easy to see that if H_i is a solution to the equation (1), so is $H_i \tilde{g}$ for any element \tilde{g} in the orientation-preserving¹ subgroup $\tilde{G} \doteq G \cap SE(3)$. This is because

$$H'_i(g)(H_i \tilde{g}) - (H_i \tilde{g})(\tilde{g}^{-1} g \tilde{g}) = 0 \quad (3)$$

and $\tilde{g}^{-1} G \tilde{g} = G$. For a proper rectangular cell (i.e. not a square), there are only two possibilities: $\tilde{G} = \{g_e, g_z\} \simeq \mathbb{Z}_2$. This comes at no surprise since if one rotates the cell by 180° about its normal, one obtains the same image of the cell from the same vantage point.

Then, from m images of a rectangular cell, we can get up to 2^m possible solutions to the camera poses (relative to the object frame) if we only rely on the homography group G' from each view. However, the ambiguity problem can be mostly resolved if the images contain more than one cell.

Theorem 1 (Cell-to-cell matching): From $m \geq 2$ different views of $n \geq 2$ rectangular cells, given that each cell is matched correctly across different views, the solution of the camera poses (with respect to each object frame) is unique if and only if at least one cell has a different rotation axis.

Be aware that here “matching” means cell-to-cell matching not point-to-point matching.²

Proof: The proof is by examining every two views of two cells. In order to have a second solution, there must exist a rigid body transformation $g \in SE(3)$ whose restriction on each cell is the same as g_z . This situation is possible only when the two cells share the rotation axis. In this case, the ambiguity is the group $\mathbb{Z}_2 = \{g_e, g_z\}$. ■

The above theorem stipulates that as long as we are able to match correctly more than one rectangular cell among images,

¹Because H_i needs to be orientation-preserving.

²Recall that, in our context, symmetry cells (not points nor lines) are the most basic geometric features.

the recovery of the camera poses will be unique except for a degenerate situation which rarely occurs in practice.

The above theorem requires that every cell is correctly matched across different views. If we relax this condition and allow a set of cells, called a *complex*, to be matched to one another among themselves across different views, would this significantly increase the possible ambiguity?

Theorem 2 (Complex-to-complex matching): Given a set of $n \geq 2$ rectangular cells matched among themselves across $m \geq 2$ views, the ambiguity in a valid solution for the camera poses is only caused by the global rotational symmetry group of the set of rectangles as a single 3-D structure.

Proof: Let us look at two views at a time. Denote the set of cells as $S = \{S^1, S^2, \dots, S^n\}$. Without loss of generality, let

$$S_1^i \longleftrightarrow S_2^i, \quad i = 1, \dots, n$$

be the correct correspondence of cells between the 1st and the 2nd views. The relative camera pose is denoted as g . Now, the existence of a second solution means that there exists a permutation σ of the set of cells in the second image such that

$$S_1^i \longleftrightarrow \sigma(S_2^i), \quad i = 1, \dots, n$$

gives another valid solution to the camera pose, say g' . Then the second image can be interpreted as two views of the same set of cells with the associated cell correspondence

$$S_2^i \longleftrightarrow \sigma(S_2^i), \quad i = 1, \dots, n.$$

The relative pose between the two views is $g_\sigma = g'g^{-1}$. Since g_σ as a rigid body transformation maps a compact set S in space to itself, it must be a rotation. Furthermore, since σ is a permutation, there exists $r \in \mathbb{Z}_+$ such that $\sigma^r = id$. So we have $g_\sigma^r = I$. Geometrically, $\{I, g_\sigma, \dots, g_\sigma^{r-1}\}$ is a rotational subgroup of the symmetry group of S . ■

Obviously, Theorem 1 becomes a special case of Theorem 2. Despite its simplicity, this theorem is in fact quite strong. It stipulates that the only valid matchings among a set of symmetry cells are caused by the global symmetry of all the cells. Notice that point features in general do not have the same nice properties as symmetry cells.³

It is also possible that certain cells can be completely mis-matched across different images, outside of a given set of cells. This is particularly the case when there are multiple repeated patterns in the scene, e.g., windows on the side of a building. Since we know the 3-D pose and structure of each cell up to scale from each view, given the camera positions, there is essentially only one configuration in which a mis-match could occur. As shown in Figure 3,

the cell S_1^2 can be matched either to the cell S_2^2 or S_2^3 in the second image if the cells S^2 and S^3 in space are parallel to each other and their shape and texture differ by a similarity transformation. Such cells form exactly a 1-parameter family in the second image. This relation can be thought of as the “epipolar constraint” for cell matching between two views. Notice that a mis-match causes only a “depth” illusion in space. However,

³In fact, it is still an open problem under what extra conditions a similar statement would be true for a set of point features.

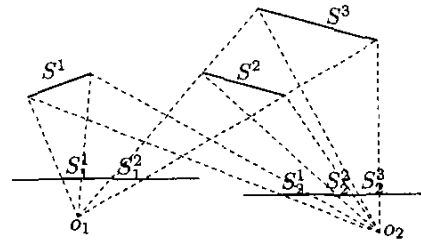


Fig. 3. “Epipolar constraint” for symmetry cells.

these types of mis-matches can be easily eliminated once we know something about S^2 in relation with any other matched cells, for example, the relative size or depth between S^1 and S^2 .

III. PICTORIAL MATCHING OF SYMMETRY CELLS

Now that we understand the multiple-view geometry of cells, we are ready to study algorithms which will allow us to automatically match cells across multiple images and then recover the camera poses and their 3-D structure.

A. Extraction of symmetry cells

In this paper, our system will be based on the assumption that symmetry cells such as rectangles and squares can already be automatically detected and extracted from images. As described in [17], based on color segmentation, polygon fitting and symmetry testing, we have built a prototype system which is able to perform this task. An example is shown in the Figure 4 below.



Fig. 4. Left: original image. Middle: image segmentation and polygon fitting. Right: symmetry cells detected and extracted – an object frame (in RGB color) is attached to each symmetry cell.

Once a set of symmetry cells are extracted from each image, we still need to know how to match them across different views. One should be aware that this is a much easier problem than matching points (with or without the camera pose known) or general regions since a cell contains much more geometric and pictorial information than a point.

B. Shape similarity of symmetry cells

For each symmetry cell in each image, we can recover its 3-D shape up to scale. If two cells in two images correspond to the same rectangle in space, the recovered two rectangles in space only differ by a planar similarity transformation. In other words, the length ratio between the edges of the recovered cells should be the same. In the presence of noise, we adopt the following simple criterion to test the shape similarity between any two recovered cells:

$$M_s(S_1, S_2) = \begin{cases} 1 & \text{if } |\tau_1 - \tau_2| \leq \tau_r, \\ 0 & \text{else} \end{cases}$$

where r_1 and r_2 are the length ratios of the two cells and $\tau_r \in \mathbb{R}_+$ is a threshold for the ratio comparison. The shape test is a very fast and effective way to determine whether the two cells can be matched to each other. It can already eliminate many impossible matches.



Fig. 5. The 3-D shape of the board cell and the table cell recovered from the image in Figure 4. They have different edge ratios and can not be matched in different views.

C. Texture similarity of symmetry cells

If two cells pass the shape matching, we can align their scales into two almost identical rectangles. The color texture in the rectangles is another cue to verify matching. We use the HSV (hue, saturation, and value) color representation to avoid the inconsistency of RGB color representation due to varying lighting conditions. However, in some cases, images with different intensity values could have the same hue, as shown by the example in Figure 6. So we also need to compare the value

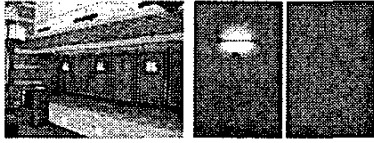


Fig. 6. The two wall panel cells recovered from the image have almost identical hues in the HSV space.

V especially when the hue is similar. We adopt the following simple criterion that combines both the hue and the value for a more robust color matching.

$$M_t(S_1, S_2) = \begin{cases} 1 & \text{if } SSD(H_1, H_2) < \tau_h, \sigma < \tau_\sigma, \\ & \text{or } NCC(V_1, V_2) < \tau_v, \sigma > \tau_\sigma, \\ 0 & \text{else} \end{cases}$$

where SSD is the “summed squared differences” and NCC is the “normalized cross correlation” and σ is the variance in the intensity values V_1 and V_2 of the two cells. Notice that we do not compare the intensity values unless there is sufficient variation since normalization by a small variance for a homogeneous texture magnifies the randomness in the intensity values.

Combine the shape and the texture matching criteria together, we claim that two cells S_1 and S_2 are matched (pictorially) if

$$M(S_1, S_2) = M_s(S_1, S_2) \wedge M_t(S_1, S_2) = 1. \quad (4)$$

IV. A CONSISTENT RECONSTRUCTION FROM COMPLETE MATCHING SUBGRAPHS

At the stage of pictorial matching, there still could be ambiguity in the orientation of the matched cells. For a rectangle like the white board in Figure 4, a rotation by 180° results in a rectangle with the same shape and color. For the square table in Figure 4, there could be four ambiguous solutions in matching.

The situation will be even worse if there are multiple similar rectangles in the scene. In general, a cell in the first image may have multiple candidate matchings in the second image in terms of its orientation and 3-D locations. The key to resolve these low-level matching ambiguities is that a correct matching of all cells should result in a consistent 3-D reconstruction of the camera poses, as claimed by Theorem 1 and 2 in the previous section.

A. Consistency between pairs of matched cells

Suppose a pair of cells are matched correctly between two views, we can use them as a reference to verify if the other matching pairs are matched correctly since ideally they should give rise to the same camera motions. For example, given two pairs of matched cells (S_1^a, S_2^a) and (S_1^b, S_2^b) , let the camera motions generated from them be $g_{21}^a = (R_{21}^a, T_{21}^a)$ and $g_{21}^b = (R_{21}^b, T_{21}^b)$. If the matching is correct, we have $R_{21}^a = R_{21}^b$ and $T_{21}^a = T_{21}^b$. The scale in the translation is determined from setting the distance from the first camera center to one of the cells to 1.

To calculate the difference between g_{21}^a and g_{21}^b , we need to measure the difference between their rotational and translational parts, respectively

$$d_R^{ab} = \arccos \left(\frac{\text{trace}(R_{21}^a R_{21}^{bT}) - 1}{2} \right), \quad (5)$$

$$d_T^{ab} = \frac{\|T_{21}^a\| \|T_{21}^b\| \arccos \left(\frac{T_{21}^{aT} T_{21}^b}{\|T_{21}^a\| \|T_{21}^b\|} \right)}{\|T_{21}^a\| \|T_{21}^b\|}. \quad (6)$$

The reason to scale the angle between translations by their magnitude is because when translation is small, the direction of translations is very sensitive to noise. A reasonable choice for a “distance” between the two motions can be

$$D(g_{21}^a, g_{21}^b) = \max \{d_R^{ab}, d_T^{ab}\}.$$

While ideally the difference between g_{21}^a and g_{21}^b should be 0, in practice we will set a threshold $\epsilon (> 0)$ for the difference. For two matched pairs with a distance less than ϵ , we say these two matched pairs are *consistent*. If we know that one pair matching is correct, then by searching for all the matchings that are consistent with this reference pair, we can find all correct matchings. In general, we do not know which matching is correct. Thus, we desire to find the maximal set of consistent matchings, which can be assumed to be the best matching for cells in the two images.

B. Cell matching graph

To find a maximal set of consistent matchings, we need to consider all pairs of matchings. For each pair of matching, there are also different ways of matching their orientations. An easy way of representing all possible matching configurations is to construct a *cell matching graph*. A matching graph \mathcal{G} includes a set of vertices \mathcal{V} and a set of edges \mathcal{E} . Each vertex $v \in \mathcal{V}$ corresponds to the k^{th} possible way of matching between the i^{th} cell in the first image and the j^{th} cell in the second image. For rectangular cells, $k = 1, 2$; for squares, $k = 1, 2, 3, 4$. So we denote $v = (i, j, k, g_{21}^{ijk})$ with g_{21}^{ijk} being the camera motion calculated from the matching. It is often convenient to use a *cell matching matrix* to generate the graph. To match m cells in the first image to n cells in the second image, a matching matrix is

an $m \times n$ matrix with the ij^{th} entry being the number of possible ways of matching between the i^{th} cell in the first image and the j^{th} cell in the second image.

The edges \mathcal{E} of \mathcal{G} are generated by connecting vertices for consistent matchings, i.e. the difference between the generated camera motions are small. Therefore, for each set of consistent matchings, the corresponding vertices and edges form a complete subgraph of \mathcal{G} . The problem of finding maximal set of consistent matchings becomes the problem of extracting maximal complete subgraphs, also called "cliques", from the graph \mathcal{G} , for which there exists a standard algorithm from graph theory [1]. Although the complexity of finding complete subgraphs is NP-complete, the computation is often simplified for sparsely connected graphs, which is exactly the case in our applications.

C. Consistent cell matching algorithm

Based on all the above steps, we can summarize as the following algorithm for matching cells between two images.

Given m cells in the first image and n cells in the second images, matchings between the two sets of cells can be established with the following steps:

1. Using the matching criteria given in Section III, construct the *matching matrix* M as described in Section IV-B: $M_{ij} = 0$ if they cannot be matched; $M_{ij} = 2$ for matched rectangular cells; and $M_{ij} = 4$ for squares.
2. Based on M , construct a set \mathcal{V} of vertices for a cell *matching graph* \mathcal{G} as described in Section IV-B. $|\mathcal{V}|$ equals the sum of all entries in M .
3. Generate a *distance matrix* D for all vertices in \mathcal{V} . D is a $|\mathcal{V}| \times |\mathcal{V}|$ matrix. For two vertices $v_s = v_{ijk}$ and $v_t = v_{i'j'k'}$, if $i = i'$ or $j = j'$, then the entry $D_{st} = \infty$. Otherwise D_{st} is the difference between g_{21}^{ijk} and $g_{21}^{i'j'k'}$ as calculated in Section IV-A.
4. Generate the edge set \mathcal{E} of the graph \mathcal{G} . Set a threshold $\epsilon > 0$. An edge (s, t) exists if and only if $D_{st} < \epsilon$.
5. Find all maximal complete subgraph \mathcal{G}_c of \mathcal{G} . Use all the matching pairs given by the vertices in \mathcal{G}_c and recover all the relative camera poses.

In the presence of repeated patterns in a scene, we usually have several complete subgraphs with similar number of vertices. By examining all of them, we can obtain different consistent solutions for the camera motions. According to Section II, these solutions are either due to the global symmetry of all the cells or due to certain mismatches. In the case of global symmetry, unless further scene knowledge is imposed, the solutions can be considered as "illusions" of the scene seen from the given vantage points.

D. Topological check of mismatches

In the presence of noise, even with the consistent cell matching algorithm, we can still encounter mismatches. It usually occurs when the cells are small in size and close to each other. For example, given two images of two close symmetry cells (namely a^1, b^1 and a^2, b^2), due to the proximity of the two similar cells and the large camera motion, the mismatch (a^1 matched to b^2 instead of a^2) will induce only a slight change in camera motion

estimation and is hard to be excluded. On the other hand, the mismatch can induce significant error in 3-D relationship for the reconstructed cells. This fact can be used for checking local topologies. That is, given a set of match, we can check the 3-D relationship between all cells in the same image. If the 3-D relationship for any pair of cells in the same image is not consistent with the 3-D relationship between their corresponding matched cells in another image, the topology check failed and this is not a valid matching.

E. An example

In the two images shown in Figure 7, extracted are 4 symmetry cells in the left image and 3 cells in the middle image. As shown in (7), the 4×3 matching matrix M is calculated using the pictorial matching scheme. Hence, the matching graph \mathcal{G} will have a total of $2 + 4 = 6$ vertices that corresponds 6 possible configurations from the two pairs of matched cells. The 6×6 distance matrix D is also shown in (7).



Fig. 7. Left and Middle: Two images with symmetry cells extracted for further matching. Right: Matching graph \mathcal{G} for the two images.

$$M = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, D = \begin{bmatrix} \infty & \infty & 174.7 & 177.7 & 177.6 & 174.6 \\ \infty & \infty & 179.2 & 89.2 & 2.43 & 90.8 \\ 174.7 & 179.2 & \infty & \infty & \infty & \infty \\ 177.7 & 89.2 & \infty & \infty & \infty & \infty \\ 177.6 & 2.43 & \infty & \infty & \infty & \infty \\ 174.6 & 90.8 & \infty & \infty & \infty & \infty \end{bmatrix}. \quad (7)$$

If we set the threshold for consistency, for instance 5 degrees, the resulting matching graph with edges is shown in Figure 7 Right. The only edge leads to the only consistent matching. This matching passes the topological check.

V. EXPERIMENTS

Based on the proposed algorithm, we have build a prototype system which is able to automatically match symmetry cells consistent through multiple images and recover the camera poses and cell structures. In all the experiments, the camera is calibrated using the Intel OpenCV calibration package. Images are all 1280×960 pixel color images.

Figure 8 shows that two symmetry cells are automatically extracted and matched across three images of an indoor scene. Correspondence of the corners of the cells are derived from



Fig. 8. Two symmetry cells matched in three images. From the raw images, symmetry cell extraction, to cell matching, the process needs no manual intervention.

the only consistent solution for the camera poses found by our algorithm, as shown in Figure 9. Due to a lot of symmetry in the scene, conventional point-based matching methods would

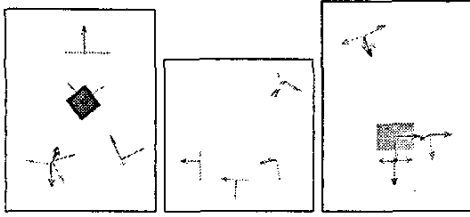


Fig. 9. Camera poses and cell structure recovered. From left to right: top, side, and frontal views of the cells and camera poses.

have difficulty with these images because similar corners lead to many mis-matches and outliers. Notice that there is only a very large rotation between the first and second views but a very large translation between the third view and the first two views. Feature tracking is impossible for such large motions and robust matching techniques based on epipolar geometry also fail since the near zero translation between the first two views makes the estimation of the fundamental matrix extremely noisy – in fact, the translation estimated from different sets of point correspondences in these two images could differ by up to 35° .

Our algorithm gives very accurate solutions. The ground truth for the length ratios of the white board and table are 1.51 and 1.00, and the recovered length ratio are 1.506 and 1.003, respectively. Errors in all the right angles is less than 1.5° .

Figure 10 shows the result of our algorithm applied to a pair of outdoor images. In the experiments, the only time-consuming

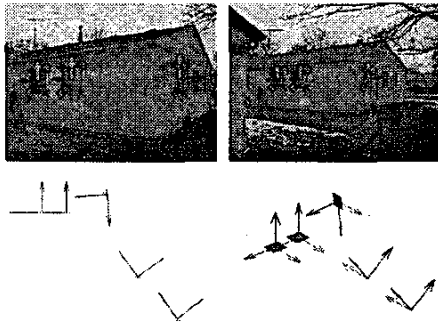


Fig. 10. From top to bottom; extracted and matched symmetry cells; recovered 3-D camera poses relative to the cells.

part is the color-based mean shift segmentation which takes a few minutes per image (on a 850 MHz laptop in Matlab). But the rest of the steps takes less than 10 seconds (in Matlab).

VI. DISCUSSION AND CONCLUSIONS

In this paper, we studied the multiple-view geometry of symmetry cells. Although the extra geometric information in symmetry cells allows us to match and recover them more easily, there are complicated ambiguities associated with their matching and reconstruction. To obtain a valid reconstruction, we need to verify the consistency among all possible matches, a task that is equivalent to the problem of finding maximal complete subgraphs in a graph.

A prototype system based on the proposed algorithms is developed. Experiments show that it gives very accurate 3-D

recovery of the camera pose and scene structure even in the cases when existing techniques have difficulties. Although the proposed method is limited to scenes with symmetric objects, the type of symmetry can be very flexible and the number of symmetric objects needed is surprisingly small.

Our method does not use epipolar geometry, which we plan to incorporate in the future into our system. This is expected to reduce further the algorithm complexity. We will also introduce a wider class of symmetry cells, besides rectangles and squares. Another important line of future work is to introduce statistical scheme to handle large number of symmetry cells, e.g. the windows on a large building.

REFERENCES

- [1] C. Bron and J. Kerbosch. Algorithm 457—finding all cliques of an undirected graph. *Communications of the ACM*, 16(9):575–577, 1973.
- [2] S. Carlsson. Symmetry in perspective. In *Proceedings of European Conference on Computer Vision*, pages 249–263, 1998.
- [3] A. Criminisi, I.Reid, and A. Zisserman. Single view metrology. *Int. Journal on Computer Vision*, 2000.
- [4] V. Ferrari, T. Tuytelaars, and L. Van Gool. Wide-baseline multiple-view correspondences. In *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2003.
- [5] M. Fishler and R. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [6] A. Francois, G. Medioni, and R. Waupoitisch. Reconstructing mirror symmetric scenes from a single view using 2-view stereo geometry. In *Proceedings of Int. Conference on Pattern Recognition*, 2002.
- [7] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge, 2000.
- [8] W. Hong, Y. Yang, and Y. Ma. On group symmetry in multiple view geometry: Structure, pose and calibration from a single image. *To appear in International Journal of Computer Vision*, 2004.
- [9] J. Kosecka, L. Zhou, P. Barber, and Z. Duric. Qualitative image-based localization in indoors environments. In *Int. Conference on Computer Vision & Pattern Recognition*, 2003.
- [10] Y. Liu and R. Popplestone. A group theoretic formalization of surface contact. *International Journal of Robotics Research*, 13(2):148–161, 1994.
- [11] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. 7th International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.
- [12] 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.
- [13] 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.
- [14] J. Shi and C. Tomasi. Good features to track. In *Int. Conference on Computer Vision & Pattern Recognition*, pages 593–600, 1994.
- [15] L.J. van Gool, M. Proesmans, and T. Moons. Mirror and point symmetry under perspective skewing. In *Int. Conference on Computer Vision & Pattern Recognition*, pages 285–292, 1996.
- [16] Allen Y. Yang, Wei Hong, and Yi Ma. Structure and pose from single images of symmetric objects with applications to robot navigation. In *EEE Int. Conf. On Robotics and Automation (ICRA) 2003*, September 2003.
- [17] A.Y. Yang, S.S.Rao, K. Huang, W. Hong, and Y. Ma. Geometric segmentation of perspective images based on symmetry groups. *Proceedings of International Conference on Computer Vision 2003*, 2003.
- [18] H. Zabrodsky and D. Weinshall. Using bilateral symmetry to improve 3d reconstruction from image sequences. *Comp. Vision and Image Understanding*, 67:48–57, 1997.
- [19] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78(1-2):87–119, 1995.