Shape from Angle Regularity

Aamer Zaheer, Maheen Rashid, and Sohaib Khan

LUMS School of Science and Engineering, Lahore, Pakistan.
http://cvlab.lums.edu.pk/SfAR

Abstract. This paper deals with automatic Single View Reconstruction (SVR) of multi-planar scenes characterized by a profusion of straight lines and mutually orthogonal line-pairs. We provide a new shape-from-X constraint based on this regularity of angles between line-pairs in man-made scenes. First, we show how the presence of such regular angles can be used for 2D rectification of an image of a plane. Further, we propose an automatic SVR method assuming there are enough orthogonal line-pairs available on each plane. This angle regularity is only imposed on physically intersecting line-pairs, making it a local constraint. Unlike earlier literature, our approach does not make restrictive assumptions about the orientation of the planes or the camera and works for both indoor and outdoor scenes. Results are shown on challenging images which would be difficult to reconstruct for existing automatic SVR algorithms.

1 Introduction

Single view reconstruction algorithms exploit different cues present in the projection of a scene to reconstruct its 3D structure. Examples of such cues include shading, texture, shadow, focus, perspective and groupings of vanishing points. These 'Shape-from-X' methods necessarily have to make assumptions about the scene structure to constrain the 3D solution — in general, a 2D projection has infinite 3D interpretations.

In this paper, we propose *angle regularity* as a new geometric constraint for reconstruction of 3D structure from a single image. Shape from angle regularity is based on the observation that if there are enough line-pairs that meet at the same angle in 3D, most commonly 90°, the distortion of this angle under projection can be used as a constraint for estimation of 3D structure[1]. Angle regularity is pervasive in architecture, which is characterized by a profusion of straight lines that meet orthogonally. Hence, appropriate exploitation of angle regularity can be a powerful cue for 3D reconstruction of man-made scenes.

The key idea in exploiting angle regularity is that the image of a 3D plane can be rectified to a fronto-parallel view by searching for the homography that

[1] For clarity of exposition, we will assume that the specified regular angle is 90° unless stated otherwise. However, the framework would hold for any other regular angle, for example 120° for the case of hexagonal tiling.

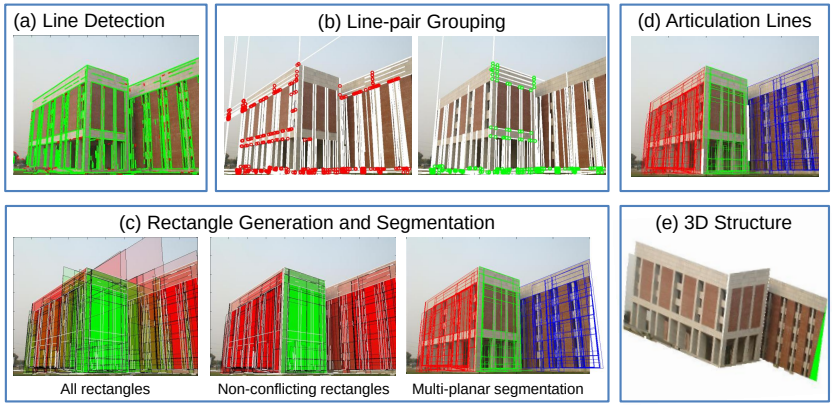

Fig. 1. Shape from Angle Regularity: (a) Original image superimposed with line detection. (b) Lines are extended to intersect, and two plane orientation hypotheses (red and green) are generated through RANSAC. (c) Line-pairs form rectangular regions and some overlapping rectangles have conflicting plane orientations. Three planar segments (red, green and blue) are identified after removing conflicts. (d) Articulation lines between planes are shown in white. (e) Novel view of 3D reconstruction.

maximizes the number of orthogonal angles between projected line-pairs (see Figure 2 for a conceptual illustration). This homography yields the normal vector of the 3D plane. For scenes containing more than one 3D plane, our approach has four main steps: 1) orthogonal line-pairs are assigned plane memberships by iteratively computing plane orientation hypotheses through RANSAC (Figure 1 a, b); 2) rectangles are generated from orthogonal line-pairs, resulting in planar segmentation of the image (Figure 1 c); 3) the adjacency of planar segments and their shared, articulating lines are computed, using global geometric analysis of all line-pairs and plane segments (Figure 1 d); and 4) the articulating lines and the plane normals are used to solve for the full 3D structure (Figure 1 e).

We identify four major limitations in earlier literature that have been removed by our approach. All the previous automatic SVR algorithms suffered from at least one of these. Firstly, they assumed the 3D scene to be one of several different 'worlds' – 'Manhattan world' [1], 'Indoor world' [1], and 'Pop-up world' [2, 3]. Each of these worlds restricted the allowable orientation of planes in 3D. In Manhattan world, planes were restricted to just three orientations; Indoor world further required that floor and roof planes were visible; and in Pop-up world, the planes were required to be vertical to a common ground plane. Secondly, the boundaries of world planes themselves were assumed to be rectilinear, spanned either by a single rectangle, or by a combination of axes-aligned rectangles. Thirdly, the camera was assumed to be in a typical orientation — at a certain height and vertically upright, often requiring that the ground plane and the ground-vertical boundary be visible. Finally, these approaches worked

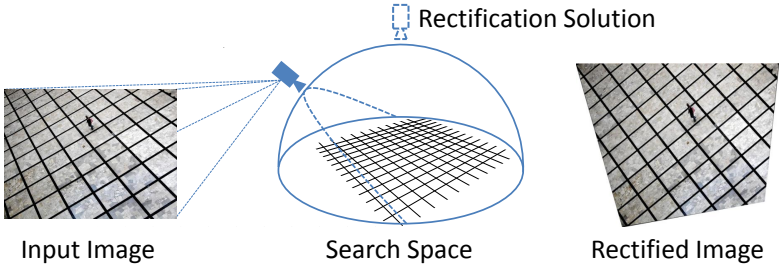

Fig. 2. 2D rectification for a single plane: The input image is taken from an unknown angle. The rectified image is computed by searching over possible pan-tilt angles of the plane, analogous to moving the camera over a hemisphere. The correct solution, given by the fronto-parallel camera, maximizes orthogonal angles in the image space, and is computed by a three-parameter optimization over camera pan, tilt and focal length.

in specific contexts, with different methods for indoor and outdoor scenarios, and required explicit removal of clutter, greenery and sky.

In contrast, our method does not place any restriction on plane orientations. In fact, by allowing planes to be oriented arbitrarily, the degrees of freedom of potential reconstructions is increased drastically over earlier work. The extents of each plane are also allowed to be more generic, where any line in the image can be a portion of the boundary between two planes. We allow the camera to be in any arbitrary orientation, and do not require visibility of the ground plane. Moreover, our approach works for both indoor and outdoor scenes, and removes clutter implicitly. While these cues can potentially be combined with texture or shading information, we demonstrate a full end-to-end system for 3D reconstruction of multi-planar man-made scenes relying exclusively on geometric cues. The limitation of the method is that it will work on man-made scenes which have sufficient orthogonal line-pairs on each plane. We show results on challenging and diverse images collected from the Internet, on some of which, none of the earlier automatic approaches are likely to work. Our experiments demonstrate that through *shape from angle regularity*, robust multi-planar segmentation, rectification, and 3D reconstruction of man-made scenes is possible — even when the camera view and plane orientation are arbitrary, line detection results are imperfect, and natural objects, such as trees, occlude part of the image.

2 Related Work

SVR algorithms usually exploit either geometry or texture, or both. A classic example of interactive single-view geometric modeling is *'Tour Into the Picture'* by Horry et al. [4], which takes the strict assumption of a Manhattan World with only one of the vanishing directions imaged on a finite vanishing point. The underlying building block of most subsequent SVR papers, based purely on geometric analysis of lines, has been the vanishing-point-based rectification

technique proposed by Liebowitz et al. [5] and by Sturm and Maybank [6]. The idea is to first group lines according to their vanishing points and then use two orthogonal vanishing points for rectification/reconstruction of a plane. They used an interactive approach for marking plane boundaries as well as vanishing points. Kang et al. [7] later extended *Tour into the picture* to work with two finite vanishing points, that is, a vanishing line. Recently, Barinova et al. [8] proposed an automatic method that removed the assumption of one infinite vanishing direction in the outdoor scenario by correcting for vertical tilt before reconstruction using a vanishing line. However, they assumed that the outdoor scene consisted of one ground plane and one or more, connected, vertical planes with rectangular boundaries. They also used a ground-building-sky segmentation through learning proposed by Hoiem et al. [9].

The outdoor model used by Barinova et al. was earlier proposed by Hoiem et al. in *Automatic Photo Popup* which combined their segmentation approach using texture based learning with geometric constraints in order to compute a *popup* model of the world [2]. Their vertical plane reconstruction was mainly based on the always visible ground-vertical boundary line. Similar learning-based idea was used in the indoor scenario by Delage et al. [10]. Saxena et al. learned a *Markov Random Field* model over *super-pixels* [11] to estimate a rough depth map of the scene in a continuous 2.5D mesh. A recent texture based approach uses *Transform Invariant Low Rank Textures (TILT)* to compute a vertical facade model similar to *Photo Popup* but, importantly, does not require availability of the ground plane [12].

Another interesting direction in geometric SVR has been an analysis of the 2D and 3D junctions between lines. The seminal work in this direction was Kanade's *Origami World* [13]. The recent *indoor world* work by Lee et al. [1] successfully combines the junction analysis paradigm with an Indoor Manhattan World assumption to recover the underlying structure of a cluttered indoor scene automatically. In addition to restricting the scene to be Manhattan, they further require that the only horizontal surfaces are floor and ceiling, and their intersection with the vertical planes is visible.

Both the state-of-the-art algorithms in geometric SVR i.e. Lee et al. in indoor scenario and Barinova et al. in the outdoor scenario depend on the two-step approach which groups the dominant vanishing points globally and assumes them to be mutually orthogonal in 3D [5]. Yu et al. showed that this global vanishing points based grouping may be ambiguous even in Manhattan environment, and proposed a local check called *spatial coherence* to protect against it [14]. They further computed rectangles in the scene and grouped them according to the order of their depth but failed to extend it to a full 3D reconstruction. In contrast, we propose a new, and inherently local, cue which allows us to both segment and reconstruct a general multi-planar structure in a bottom-up fashion. In addition to being the first automatic method for planes in arbitrary orientations, ours is the first automatic method that allows non-rectilinear plane boundaries and does not restrict the camera orientation while being robust to significant amounts of clutter.

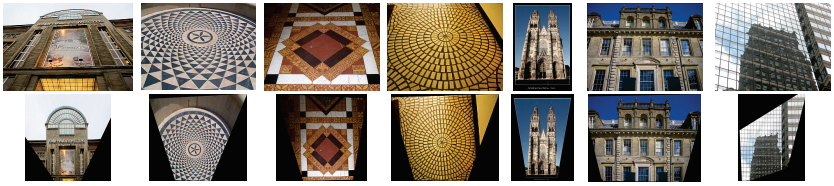

Fig. 3. Automatic rectification results on challenging images taken from the Internet. EXIF data was available, hence two-parameter optimization was used. Note that even when lines do not align to common vanishing points, as in the circular tiled patterns, the algorithm works because it exploits orthogonalities locally.

3 2D Rectification

The image of a plane can be rectified to a fronto-parallel view if some line-pairs on that plane meet orthogonally in 3D. The distortion of the angles between these lines under perpective projection constrains the rectifying homography, as illustrated conceptually in Figure 2. A homography induced by camera rotations is given by [15]

$$\mathtt{H} = \mathtt{K}\mathtt{R}_Z^{\gamma}\mathtt{R}_Y^{\beta}\mathtt{R}_X^{\alpha}\mathtt{K}^{-1}, \tag{1}$$

where $\mathtt{K}$ is the 3×3 matrix of intrinsic parameters, and $\mathtt{R}_X^{\alpha}$, $\mathtt{R}_Y^{\beta}$, and $\mathtt{R}_Z^{\gamma}$ denote rotations about the X, Y, and Z axes of the camera by α, β, and γ respectively[2]. We assume square pixels and the image origin at camera's principal point, which reduces $\mathtt{K}$ to $\mathtt{diag}\,[f, f, 1]$, containing a single focal length parameter, f. Under these assumptions, $\mathtt{K}\mathtt{R}_Z^{\gamma}$ simplifies to a similarity transform, which has no effect on angles and can therefore be ignored. Hence, the search space for the rectifying homography is reduced to just three parameters, α, β, and f. For most of our experiments, the focal length is known from the camera's EXIF data, leaving only two parameters, α and β — the rectifying homography is now given by $\mathtt{H}\,(\alpha, \beta) = \mathtt{R}_Y^{\beta}\mathtt{R}_X^{\alpha}\mathtt{K}^{-1}$. We have empirically found that these simplifying assumptions do not qualitatively degrade results[3].

To search for the rectifying homography $\hat{\mathtt{H}}$, consider two lines in the image $\mathbf{l}_i$ and $\mathbf{l}_j$ which were orthogonal in 3D world, but their mutual angle has been distorted by perspective projection. An arbitrary homography, $\mathtt{H}$, transforms these lines to $\mathtt{H}^{-\top}\mathbf{l}_i$ and $\mathtt{H}^{-\top}\mathbf{l}_j$ respectively and the correct rectifying homography should make them orthogonal. Let $\tilde{\mathbf{v}}_i$ and $\tilde{\mathbf{v}}_j$ be the first two elements of the

[2] We follow the mathematical notation of Hartley and Zisserman [15]; x is a scalar; $\mathbf{x}$ and $\mathbf{X}$ are homogeneous vectors in $\mathbb{P}^2$ and $\mathbb{P}^3$ respectively; $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{X}}$ are their inhomogeneous versions; and $\mathtt{X}$ denotes a matrix.

[3] If focal length is unknown, the two vanishing points of a plane must be at finite location; if any of them is at infinity, the relative scale of the two dimensions will be unconstrained for the case when lines are in just two dominant directions. This ambiguity does not arise when focal length is known.

vectors $\mathtt{H}^{-\top}\mathbf{l}_i$ and $\mathtt{H}^{-\top}\mathbf{l}_j$ normalized to unit norm respectively; then they represent unit vectors orthogonal to these lines in 2D. Hence, the cost of the rectifying homography, $\hat{\mathtt{H}}$, can be given by $||\tilde{\mathbf{v}}_i^\top \tilde{\mathbf{v}}_j||^2$. A homography that minimizes this cost over all line-pairs is the rectifying homography, which restores angles distorted by perspective projection to their original values. This cost function can be written as

$$\mathcal{C}\left(\mathtt{H}\left(\alpha,\beta\right)\right) = \sum_{i,j} ||\tilde{\mathbf{v}}_i^\top \tilde{\mathbf{v}}_j||^2. \tag{2}$$

Since we do not know beforehand which line-pairs are indeed orthogonal, we use RANSAC [16], by assuming any two line-pairs to perpendicular, and using them to compute the rectifying homography[4]. The inliers are computed by counting how many other line-pairs have become orthogonal in the rectified image. Line segments are detected in the original image using the implementation of von Gioi *et al.* [17], with an additional gap-filling step. Each input segment is extended till it intersects another line segment, and only such intersecting pairs are used for optimization.

It is useful to observe that this optimization could have been formulated for any known angle constraint between the lines, rather than for just 90° angles. In that case, the cost function in Equation 2 may be modified to

$$\mathcal{C}\left(\mathtt{H}\left(\alpha,\beta\right)\right) = \sum_{i,j} (||\tilde{\mathbf{v}}_i^\top \tilde{\mathbf{v}}_j||^2 - \cos^2\hat{\theta})^2, \tag{3}$$

where $\hat{\theta}$ is the known angle expected to occur frequently on the plane.

Results of automatic rectification for real images taken from the Internet are shown in Figure 3. Note that for the circular tiling patterns, vanishing point based approaches [5] will not work, because there is no dominant alignment of parallel lines. However, our method uses local orthogonalities and does not assume alignment between line-pairs; hence the orthogonal angles of the tiles, even though arranged in a circular orientation, are enough to constrain the solution. EXIF data was available for these images, and hence optimization was reduced to two parameters. The rectification shown in Figure 2 was computed without EXIF data, through a three-parameter search over α, β and f.

4 Computation of Multi-Planar Structure

The rectification process discussed above can be applied to a multi-planar scene, if each plane is segmented independently. The segmentation problem itself will be discussed in the next section; here we assume that it is available.

To compute the normal of a plane in 3D, let $\mathtt{R}_i$ be the 3×3 rotation the camera has to undergo in order to become fronto-parallel to the i-th plane. This rotation is available through rectification. The optical axis, $\mathbf{Z}$, of the rectified

[4] If focal length is not known, minimum three line-pairs are needed to uniquely constrain the rectifying homography.

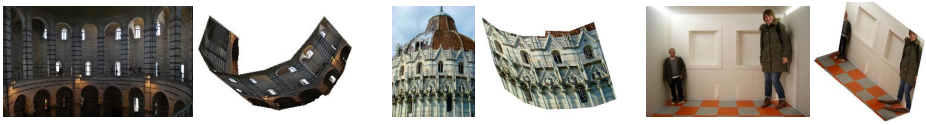

Fig. 4. Reconstruction with manual segmentation: The curved surfaces were treated as piece-wise planar, and their planar segments were identified. The right-most figure shows Ames Room illusion, where our reconstruction mimics human perception.

camera is perpendicular to the 3D plane (Figure 2). Hence, the normal vector, $\tilde{\mathbf{n}}_i$, of the plane in the original camera frame is computed by applying the inverse of the rectifying rotation to the Z-axis:

$$\tilde{\mathbf{n}}_i = \mathtt{R}_i^\top \, [0, 0, 1]^\top . \tag{4}$$

Once the plane normal is known, the only remaining parameter is its depth. Since the structure can be reconstructed only up to an arbitrary global scale, we need to recover just the *relative* depths of planes, that is, the ratios of their depths should be preserved. The relative depths of a pair of planes can be computed if we have one or more common points between them. Our solution is similar to the interactive approach used by Sturm and Maybank [6], which was based on finding plane normals through vanishing points, and then solving for the common points and relative depths together. Though we only compute relative depths, and use a different constraint, our linear system turns out to be similar.

Let $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$ be two planes with normal vectors $\tilde{\mathbf{n}}_1$ and $\tilde{\mathbf{n}}_2$, respectively; i.e. $\boldsymbol{\pi}_j = [\tilde{\mathbf{n}}_j^\top \, , d_j]^\top$, where d_j represents the depth of the plane. Assume that the two planes share a common point $\mathbf{x}$. Given that the camera is in the canonical view and the camera intrinsic matrix $\mathtt{K}$ is known, we may back-project the point into a 3D ray $\tilde{\mathbf{X}} = \mathtt{K}^{-1}\mathbf{x}$. The 3D point $\mathbf{X}$ imaged as $\mathbf{x}$ may lie anywhere on the ray $\alpha\tilde{\mathbf{X}}$ for some positive α i.e. $\mathbf{X} = [\alpha\tilde{\mathbf{X}}^\top, 1]^\top$. In our particular case, however, this ray must intersect both the planes $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$ at the same 3D point, therefore

$$\boldsymbol{\pi}_1^\top \begin{bmatrix} \alpha\tilde{\mathbf{X}} \\ 1 \end{bmatrix} = \boldsymbol{\pi}_2^\top \begin{bmatrix} \alpha\tilde{\mathbf{X}} \\ 1 \end{bmatrix} = 0. \tag{5}$$

Equating the two α values and rearranging yields

$$\begin{bmatrix} \tilde{\mathbf{n}}_2^\top\tilde{\mathbf{X}} \, , -\tilde{\mathbf{n}}_1^\top\tilde{\mathbf{X}} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = 0. \tag{6}$$

Generalizing to multiple planes and allowing for the possibility of more than one articulating point for each plane-pair, the set of constraints on relative depths can be written as a linear system,

$$\mathtt{A}\mathbf{d} = \mathbf{0}, \tag{7}$$

where $\mathbf{d} = \begin{bmatrix} d_1, \ldots, d_p \end{bmatrix}^\top$ contains the relative depths of p planes and every row of $\mathtt{A}$ contains one common point constraint between the j-th and k-th planes

such that $a_{i,j} = \tilde{\mathbf{n}}_k^\top \tilde{\mathbf{X}}_i$, $a_{i,k} = -\tilde{\mathbf{n}}_j^\top \tilde{\mathbf{X}}_i$ and rest of the elements in the i-th row are zeros.

The vector of relative depths, $\mathbf{d}$, is the right null vector of $\mathtt{A}$ and is computed through SVD. It is recovered up to an arbitrary scale because Equation 7 represents a homogeneous system. In order to fairly weigh the constraints, all back-projection rays are normalized to unit norm. Note that relative depths of planes can only be computed correctly for a set of planes if they are *connected*, i.e. they are either directly adjacent through some common points or by association through intermediate adjacent planes.

The linear solution discussed above may contain inaccuracies due to error in plane normals or noise in the observed common points. We refine the solution by using it as an initial guess in a non-linear optimization over all planes $\boldsymbol{\pi}_k$ and focal length f, which minimizes both the orthogonality cost for line-pairs $\mathcal{C}_k$ (Equation 2) on each plane as well as error in observed common points. The error $\mathcal{D}_{l,m}$ in common points is measured by projecting the articulation line and computing the sum of its normal distances from all common points for adjacent planes $\boldsymbol{\pi}_l$ and $\boldsymbol{\pi}_m$. The articulation lines are parameterized by the join of their planes, so that they are geometrically consistent with the solution of plane parameters. The objective function to be minimized is given by

$$\mathcal{E}(f, \boldsymbol{\pi}_1, ..., \boldsymbol{\pi}_p) = \sum_{\forall \text{ adjacent planes } (\boldsymbol{\pi}_l, \boldsymbol{\pi}_m)} \mathcal{D}_{l,m} + \sum_{\forall \text{ planes } \boldsymbol{\pi}_k} \mathcal{C}_k. \tag{8}$$

Figure 4 shows reconstruction of structure from single images when segmentation is provided. All other steps are performed automatically with no specific parameter tuning for different test cases.

5 Multi-Planar Segmentation and Single-View Reconstruction

Now we discuss the complete algorithm for shape from angle regularity, including a strategy to segment the image into its planar components based on geometric cues only. Our full algorithm can be organized into four main steps, which are elaborated below. Line segments and their adjacency relationships are computed the same way as in Section 3.

5.1 Plane Orientations

The method to find plane orientation from image of a plane is described in Section 3. For multi-planar scenes, we first apply RANSAC to find the adjacent line-pairs belonging to the most dominant plane orientation in the image, then remove these RANSAC inliers from the input. The process of RANSAC on remaining line-pair population and inlier removal is repeated until a sizable consensus set cannot be found anymore. This process detects arbitrary plane orientations since no global constraints on plane or line orientations are employed. Different but parallel planes are grouped together at this stage.

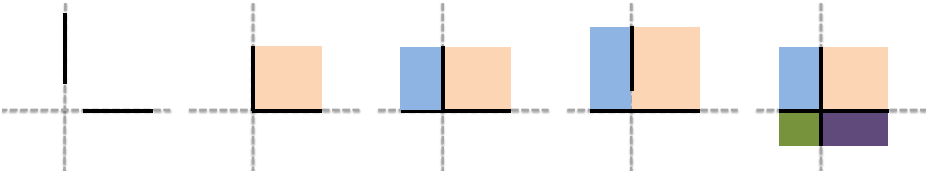

Fig. 5. Rectangles formed by line intersections: Detected line segments (solid black) and their supporting lines (dashed gray) generate the rectangles (colored boxes). Five different cases of intersection are illustrated from left to right, resulting in zero, one, two, two and four rectangles respectively. Note that the rectangles are in rectified view, and will perspectively distort according to the plane orientation in the original image.

Note, however, that one line pair may be rectified by more than one plane orientations and may have multiple classifications. A common example of this ambiguity is demonstrated in Figure 1b where the imaged ground-vertical boundary becomes almost collinear and participates in line-pairs rectified by both plane orientations. Here line-pairs are marked by circles at their point of intersection and some line-pairs near the bottom are marked in both orientation groups indicating dual grouping. In order to allow multiple labels for every line-pair, we compute orientation inliers again for each pre-computed plane orientation hypothesis but without removing any line-pairs from consideration as done in RANSAC. At the end of this step, most line-pairs in the image are given one or more labels, depending on the plane normal that they support. Only the line-pairs that have at least one label are considered during the planar segmentation step.

5.2 Planar Segmentation

Line-pairs have been assigned plane orientation labels in the previous step but the extents of planes are also needed for 3D reconstruction. We take a bottom-up approach for detecting planar segments in the image. First, the labeled line-pairs are used to generate oriented planar regions which serve as the basic unit for bottom-up segmentation. These region hypotheses are verified for correctness and grouped later on to form planar segments.

An orthogonal line-pair based on detected line segments may intersect in five different ways as illustrated in Figure 5. For each case, rectangular regions are created while making sure that the intersection point of the line-pair lies on at least one physical line segment, and the rectangle is supported by both the line segments. Note that these rectangles exist in the rectified view, but appear perspectively distorted to the 'correct' orientation in the original image. If a line-pair has more than one orientation labels, rectangles are computed for each of the orientations so that every perspectively distorted rectangle has correct extents in the image and has a unique orientation label.

Previous literature on multi-planar segmentation uses pixels, super-pixels or lines as basic unit for bottom-up segmentation. The use of rectangles, rather

than any other method of plane segmentation, is key to our segmentation algorithm. Given the orthogonality constraint our line-pairs must follow, rectangles arise naturally and allow us to compute region hypotheses. Moreover, since two sides of a rectangle are supported by its generating line-pair, the opposite sides form useful 'hallucinated' lines, which are often line segments that have not been identified by the line detector, or are occluded by image clutter. Lastly, frequent overlap between rectangles is useful in a consensus based region grouping approach as explained below.

Since each rectangle supports a plane orientation, some overlapping rectangles have conflicting orientation labels. We remove these conflicts by dropping some rectangles. The idea is to discard as few rectangles as possible, so we iteratively remove the most 'troublesome' rectangles in two steps. Step 1: since each rectangle contains many line-pairs, some of them have an orientation label different from the label assigned to the rectangle. We first compute the 'inlier percentage' for each rectangle, i.e. the number of line-pairs having the same plane orientation divided by the total number of line-pairs inside that rectangle. Here, a line-pair is considered inside a rectangle if the intersection point of the line-pair lies in the rectangle. This step yields a normalized measure of correctness of a rectangle.

Step 2: For each rectangle, we define its 'conflict score' as the sum of the inlier percentages of conflicting rectangles it overlaps with. High conflict score for a rectangle means that it conflicts with many good rectangles and should be removed. If we remove a rectangle then the conflict scores of the conflicting rectangles will also change. We iteratively remove the rectangle with the highest conflict score and then update the conflict scores for all the rectangles. This greedy conflict removal process is repeated until there are no more conflicts left. After conflict removal, overlapping rectangles must belong to the same plane orientation label, and each pixel in the image gets at most one label. Some pixels may not be labeled at all if they lie in a region away from any rectified line-pair, such as pixels in the sky. This implicitly results in removal of clutter and sky regions.

The planar segmentation process is illustrated in Figure 1. Lines are detected on the original image (Figure 1a) and two plane orientation hypotheses are generated through RANSAC. The inlier line-pairs of each orientation are shown in Figure 1b marked by circles at their intersection points. The rectangles induced by these inliers result in the distorted rectangles shown in Figure 1c (left). The color of each rectangle indicates its plane orientation label. Note that significant overlaps exist between rectangles of different colors before conflict removal. However, after conflict removal, the remaining overlapping rectangles do not have different colors as Figure 1c (middle) demonstrates. Also, parallel but different planes still have the same labels as shown by two red planes. In order to separate physical planar regions, we make sure that only overlapping rectangles have the same orientation by using graph based connected components algorithm. This results in physically contiguous regions getting unique labels as shown in Figure

1c on the right. These groups of rectangles are treated as planar segments in the subsequent steps of the algorithm.

5.3 Articulation Lines

At this stage we have planar segments containing line segments and their plane normals. In order to do 3D reconstruction we must ascertain whether a pair of planar segments is connected to each other and if yes, then identify the articulation line between the pair. Although knowing one image point common to both planes is enough to constrain their relative depths but we identify the articulation lines because two planes always intersect at a line in 3D. We assume that the articulation line has either been detected or hallucinated in our rectangle generation process. This assumption significantly reduces the search space for articulations while allowing for a general polygonal plane boundary — a significantly relaxed model for plane boundaries as compared to earlier literature.

For a selected pair of planes and all the candidate line segments, detected or hallucinated, we first filter out the lines which are geometrically inconsistent with the normals of these two planes. We compute the plane relative depths using one of the end-points of a given line segment. The 3D intersection line of these two planes is computed and projected in the image. If the line segment is consistent with the plane normals, its second end-point must also lie on the projected articulation line. Therefore, we filter out the line segments which make an angle larger than 5° with the projected articulation line. It results in a set of geometrically plausible articulation lines for the pair of planes.

In order to identify the best articulation line from a set of plausible lines, we apply two heuristics: 1) a good line should separate the the two plane segments well and, 2) should not be too far away from either segment. After picking a few best separating lines, we compute the minimum distance to both the plane segments and pick the line with the least distance as the *best articulating line* for this pair of planes. This process is performed for all the plane-pairs to identify the best possible line for every plane-pair.

Some plane-pairs might not have any plausible articulation lines because they are not adjacent while others may be assigned incorrect lines accidentally through this process. It is reasonable to assume that the correct articulating line will not be too far away from either of the segments in a physically adjacent pair. We use this observation in two ways. First, we remove plane articulations from consideration if it is too far away from either of the corresponding plane segments. Second, we form a plane adjacency graph where two nodes are connected if they have a valid articulating line and the weight of the edge is the maximum distance of that articulating line from the plane segments. We find the *Minimum Spanning Tree* (MST) of the largest connected component in this graph which results in a minimal set of best constraints. We throw away the smaller connected components that are not connected to this tree because the relative depth of two disconnected plane sets cannot be reconstructed by our algorithm in Section 4.

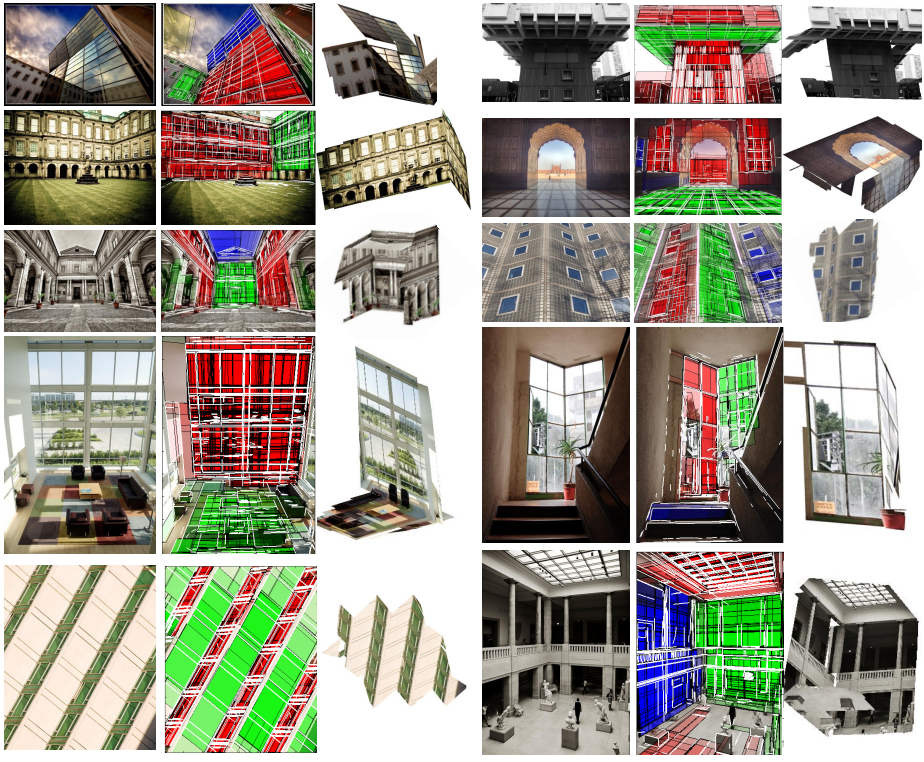

Fig. 6. More results of single-view reconstruction using angle regularity. The columns 2 and 5 show detected lines and the colors indicate segmentation labels. The columns 3 and 6 show 3D structure texture-mapped from a novel view-point.

5.4 Structure Recovery and Refinement of Plane Extents

Given that articulation lines and plane normals are known, we can now compute the 3D structure through the formulation in Section 4. After reconstruction, as a final step, we determine the extents of the planes by considering the union of all rectangles belonging to that plane and the corresponding articulation line. This is because the articulation line may not lie within the connected component of the rectangles supporting that plane.

6 Results

While results for 2D rectification, reconstruction and segmentation have been illustrated earlier, here we show the results of the overall automatic system. No specific parameter tuning was performed, other than the scale parameter that is required for good line detection by the Line Segment Detector (LSD) code [17]. Focal length was provided through the EXIF data for all photographs but it was automatically computed for the sketch in Figure 7.

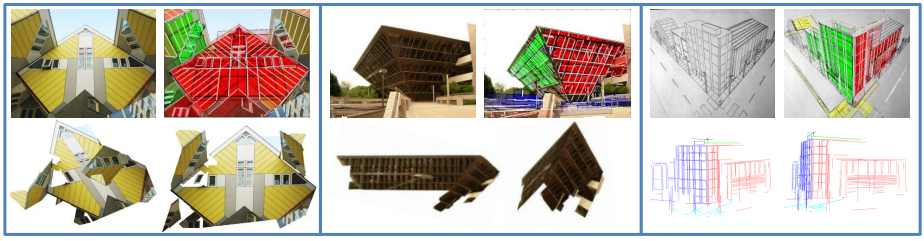

Fig. 7. Illustrative results for Shape from Angle Regularity on challenging images: Note that none of these images would be reconstructed by Pop-up or Manhattan models.

The results in Figure 6 show several diverse but typical scenarios including indoor, outdoor, Manhattan, non-Manhattan and Popup scenes as well as scenes with significant clutter and sky regions. These results demonstrate the general applicability of our approach for everyday man-made scenes. Some of them would not have been reconstructed by existing approaches because of their restrictive world assumptions, because of the ground plane not being visible, planes not meeting at orthogonal angles, and atypical camera orientation.

Figure 7 shows segmentation and reconstruction on very challenging images. These images do not follow the geometric models of any of the state-of-the-art automatic SVR algorithms. The first image shows a very awkward structure whose details have been correctly segmented and reconstructed by our approach. The second example defies vertical-walls assumption required by earlier SVR algorithms, and also contains contains sky and clutter which has been correctly filtered out. The third example is that of a line sketch. In this case, three parameter search was carried out to recover focal length, and correct structure recovery is illustrated.

Typical failure cases are illustrated in Figure 8. Our algorithm will not generate the correct depth of two planes that do not have an intermediate set of connecting plane visible between them. Other failure reasons include not finding enough line-pair constraints on a plane, and incorrect grouping of constraints between planes.

The results shown here are for illustrative purposes. More examples, dataset and code is available on our project page: `http://cvlab.lums.edu.pk/SfAR`.

References

1. Lee, D., Hebert, M., Kanade, T.: Geometric reasoning for single image structure recovery. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE (2009) 2136–2143
2. Hoiem, D., Efros, A., Hebert, M.: Automatic photo pop-up. ACM Transactions on Graphics (TOG) **24** (2005) 577–584
3. Barinova, O., Lempitsky, V., Tretiak, E., Kohli, P.: Geometric Image Parsing in Man-Made Environments. Computer Vision–ECCV 2010 (2010) 57–70

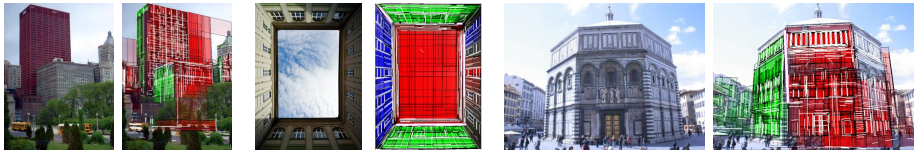

Fig. 8. Typical failure cases of segmentation. First is a failure of segmentation due to discontinuity between planes. Second is an example in which constraints on one plane have been incorrectly grouped with multiple adjacent planes in earlier iterations of RANSAC. Last example shows a failure of RANSAC grouping due to insufficient distinguishing line-pairs on two planes.

4. Horry, Y., Anjyo, K., Arai, K.: Tour into the picture: using a spidery mesh interface to make animation from a single image. In: Proceedings of the 24th annual conference on Computer graphics and interactive techniques, ACM Press/Addison-Wesley Publishing Co. (1997) 225–232
5. Liebowitz, D., Criminisi, A., Zisserman, A.: Creating architectural models from images. In: Computer Graphics Forum. Volume 18. (1999) 39–50
6. Sturm, P., Maybank, S.: A method for interactive 3d reconstruction of piecewise planar objects from single images. In: In British Machine Vision Conference. (1999)
7. Kang, H., Pyo, S., Anjyo, K., Shin, S.: Tour into the picture using a vanishing line and its extension to panoramic images. In: Computer Graphics Forum. Volume 20., Wiley Online Library (2001) 132–141
8. Barinova, O., Konushin, V., Yakubenko, A., Lee, K., Lim, H., Konushin, A.: Fast Automatic Single-View 3-d Reconstruction of Urban Scenes. In: Proceedings of the 10th European Conf. on Computer Vision: Part II, Springer-Verlag (2008) 100–113
9. Hoiem, D., Efros, A., Hebert, M.: Geometric context from a single image. In: Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on. Volume 1., IEEE (2005) 654–661
10. Delage, E., Lee, H., Ng, A.: A dynamic bayesian network model for autonomous 3d reconstruction from a single indoor image. In: Computer Vision and Pattern Recognition, 2006. Volume 2., IEEE (2006) 2418–2428
11. Saxena, A., Sun, M., Ng, A.: Make3d: learning 3d scene structure from a single still image. IEEE transactions on pattern analysis and machine intelligence (2008) 824–840
12. Zhang, Z., Liang, X., Ganesh, A., Ma, Y.: Tilt: transform invariant low-rank textures. Computer Vision–ACCV 2010 (2011) 314–328
13. Kanade, T.: A theory of Origami world. Artificial Intelligence **13** (1980) 279–311
14. Yu, S., Zhang, H., Malik, J.: Inferring spatial layout from a single image via depth-ordered grouping. In: Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on, IEEE (2008) 1–7
15. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Second edn. Cambridge University Press, ISBN: 0521540518 (2004)
16. Fischler, M., Bolles, R.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24** (1981) 381–395
17. von Gioi, R., Jakubowicz, J., Morel, J., Randall, G.: LSD: A Fast Line Segment Detector with a False Detection Control. Pattern Analysis and Machine Intelligence, IEEE Transactions on **32** (2010) 722–732