3D Reconstruction of Planar Surface Patches: A Direct Solution

Jozsef Molnar, Rui Huang, and Zoltan Kato

Institute of Informatics, University of Szeged, Arpad ter 2, 6720 Szeged, Hungary
kato@inf.u-szeged.hu

Abstract. We propose a novel solution for reconstructing planar surface patches. The theoretical foundation relies on variational calculus, which yields a closed form solution for the normal and distance of a 3D planar surface patch, when an affine transformation is known between the corresponding image region pairs. Although we apply the proposed method to projective cameras, the theoretical derivation itself is not restricted to perspective projection. The method is quantitatively evaluated on a large set of synthetic data as well as on real images of urban scenes, where planar surface reconstruction is often needed. Experimental results confirm that the method provides good reconstructions in real-time.

1 Introduction

Wide baseline multi view stereo has important role in image-based urban scene reconstruction [1]. Classical approaches are either based on sparse point correspondences or dense stereo matching [2]. Then a 3D point cloud is obtained, which is the basis for scene objects' mesh modeling. Recently Poisson surface reconstruction [3] became widely used for this purpose. This method uses point coordinates as well as normal vectors to construct a smooth and detailed polygon mesh. Recently, region-based methods has been gaining more attention, in particular affine invariant detectors [4]. This affine invariance is closely related to the normal of the observed surface patch as we will see in this paper in a general context.

The most frequently used volumetric 3D object representation obtained by space carving [5] or variational level set methods [6] requires bounded objects. The accuracy of the reconstruction is determined by the resolution of the spatial grid used to define the smallest distinguishable elements. These methods would not fit for large open scenes. Multiple depth map [7] is a possible alternative 3D object representation, but it requires complicated registration steps in a later stage assuring the consistency and accuracy. Patch-based scene representation is proved to be efficient [8] and consistent with region-based correspondence-search methods.

The importance of piecewise planar object representation in 3D stereo has been recognized by many researchers. Habbecke and Kobbelt used a small plane, called 'disk', for surface reconstruction [9, 10]. They proved that the normal is a linear function of the camera matrix and homography. By minimizing the difference of the warped images, the surface is reconstructed. In this paper, we give a closed form solution to surface normal and distance. Kannala and Brandt also started from a seed region which is obtained by point detector or blob detector [11]. An affine transformation is then applied to the seed region for further propagation. In our method, we determine planar perspective transformation which provides the surface normal and distance in a closed form. Furukawa proposed using a small patch for better correspondence [12]. The surface is then grown with the expansion of the patches. The piecewise planar stereo method of Sinha *et al.* [13] uses shape from motion to generate an initial point cloud, then a best fitting plane is estimated, and finally an energy optimization problem is solved by graph cut for plane reconstruction. Combining the work by Furukawa and Sinha [12, 13], Kowdle *et al.* introduced learning and active user interaction for large plane objects [14]. Hoang *et al.* also started from a point cloud [15] which was subsequently used for creating a visibility consistent mesh. In our approach, planes are directly reconstructed from image region(s) rather than a point cloud. Fraundorfer *et al.* [16] used MSER regions to establish corresponding regions pairs. Then a homography is calculated using SIFT detector inside the regions. Planar regions are then grown until the reprojection error is small. Zhou *et al.* assumed the whole image is a planar object, and proposed a short sequence SFM framework called TRASAC [17]. The homography is calculated using optical flow. Although the role of planar regions in 3D reconstruction has been noticed by many researchers, the final reconstruction is still obtained via triangulation for most state-of-the-art methods. Planar objects are only used for better correspondences or camera calibration.

In this paper we will develop a direct method to reconstruct whole planar patches using only the camera matrices and an affine or homography map between the image region pairs corresponding to the 3D scene patch. Since we use the correspondence-less approach of Domokos *et al.* [18] to estimate planar homography directly between image regions, our method doesn't require any point correspondences between stereo image pairs. Another important advantage of the proposed method is its real-time performance due to the closed form solution while also maintaining robustness. This opens the way to use our reconstruction algorithm on mobile or embedded devices.

The theoretical derivation of the general formula for 3D plane reconstruction is presented in Section 2, where we also discuss numerical stability of the formulas based on geometric consideration, and a simple recipe is also proposed to avoid unstable situations. Section 3 contains comprehensive numerical test results both for normal and distance calculation using synthetic and real data.

2 Normal and Distance Computation

We now derive a simple, closed form solution to reconstruct the normal and distance of a 3D planar surface patch from a pair of corresponding image regions and the camera matrices. Although differential geometric approaches were used to solve various problems in projective 3D reconstruction, the approach proposed here is unique to the best of our knowledge. For example, [19, 20] are about generic surface normal reconstruction using point-wise orientation- or spatial frequency disparity maps, while our method avoids point correspondences and reconstructs both normal and distance of a planar surface from the induced planar homography between image regions. Unlike [19, 20], which consider only projective camera and uses a parameterization dependent, non-invariant representation; we use a very general camera model and invariant representation.

The notation used in this section follows [21] and is widely used in continuum mechanics and classical differential geometry. For vectors and tensors we use bold letters. We use the symbol "$\cdot$" for dot product, between tensors (in their matrix representation this is the usual matrix-matrix product). A simple sequence of vectors represents their dyadic product. The transpose of a dyad is the reversed sequence of the constituent vectors. A short "dictionary" is provided here for quick reference: $\mathbf{a}^T\mathbf{b} \to \mathbf{a} \cdot \mathbf{b}$, $\mathbf{a}\mathbf{b}^T \to \mathbf{ab}$, $\mathbf{Ab} \to \mathbf{A} \cdot \mathbf{b}$, $\mathbf{b}^T\mathbf{A} \to \mathbf{b} \cdot \mathbf{A}$, $\mathbf{AB} \to \mathbf{A} \cdot \mathbf{B}$, where $\mathbf{a}, \mathbf{b}$ are column vectors and $\mathbf{A}, \mathbf{B}$ are second order tensors represented by two-dimensional matrices. Note that in this notation $(\mathbf{ab})^T = \mathbf{ba}$.

2.1 Basic equations for normal computation

Herein, after briefly summarizing the theoretical backround based on [21], we will show how these results can be applied to compute the normal of a 3D scene plane from corresponding observed image regions. Let us consider the visible part of the scene objects as reasonably smooth surfaces embedded into the ambient 3D space. An image of the scene is a 3D-2D mapping given by two smooth projection functions: $x = x(X, Y, Z)$, $y = y(X, Y, Z)$, with x, y being the image coordinates. Hereafter we don't assume any special form of these coordinate-functions, except their differentiability w.r.t. spatial coordinates X, Y, Z of a world coordinate system given in standard basis $\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$. For the surface representation we use general Gauss-coordinates:

$$\mathbf{S}(u, v) = X(u, v)\,\mathbf{i} + Y(u, v)\,\mathbf{j} + Z(u, v)\,\mathbf{k} \tag{1}$$

If the projected spatial points are on the surface too, the image coordinates depend on the general parameters as well:

$$\begin{array}{l} x(u, v) = x(X(u, v), Y(u, v), Z(u, v)) \\ y(u, v) = y(X(u, v), Y(u, v), Z(u, v)) \end{array} \tag{2}$$

We suppose that the surface point $\mathbf{u_0} = \begin{bmatrix} u_0 \ v_0 \end{bmatrix}^T$ with a neighborhood constituting a small open patch are visible, therefore its mapping to the camera image is a bijection. The differential $d\mathbf{u} = \begin{bmatrix} du \ dv \end{bmatrix}^T$ represents a point shift on the surface with its effect on the image being $d\mathbf{x} \approx \mathbf{J} \cdot d\mathbf{u}$ where $d\mathbf{x} = \begin{bmatrix} dx \ dy \end{bmatrix}^T$ and the Jacobian $\mathbf{J}$ of the mapping is invertible. Now consider a stereo camera pair (distinguishing them with indices i, j). Since $\mathbf{J}$ is invertible, we can establish correspondences between the images having the same point-shift $d\mathbf{u} = \mathbf{J}_i^{-1} \cdot d\mathbf{x}_i$:

$$d\mathbf{x}_j = \mathbf{J}_j \cdot \mathbf{J}_i^{-1} \cdot d\mathbf{x}_i = \mathbf{J}_{ij} \cdot d\mathbf{x}_i \tag{3}$$

where $\mathbf{J}_{ij}$ is the Jacobian of the $\mathbf{x}_i \rightarrow \mathbf{x}_j$ mapping. Considering the derivative of a composite function f:

$$\frac{\partial f}{\partial u} = \frac{\partial X}{\partial u}\frac{\partial f}{\partial X} + \frac{\partial Y}{\partial u}\frac{\partial f}{\partial Y} + \frac{\partial Z}{\partial u}\frac{\partial f}{\partial Z} = \mathbf{S}_u \cdot \nabla f \tag{4}$$

where ∇f is the gradient of f w.r.t. the spatial coordinates, and $\mathbf{S}_u$ is the local basis vector alongside parameter line u. Applying this result to the projection functions, the Jacobians take the following form:

$$\mathbf{J}_k = \begin{bmatrix} \mathbf{S}_u \cdot \nabla x_k \; \mathbf{S}_v \cdot \nabla x_k \\ \mathbf{S}_u \cdot \nabla y_k \; \mathbf{S}_v \cdot \nabla y_k \end{bmatrix}, \quad k = i, j \tag{5}$$

After substitution, the products of the above quantities appear in $\mathbf{J}_{ij}$. For example, the determinant

$$\det(\mathbf{J}_i) = (\mathbf{S}_u \cdot \nabla x_i)(\mathbf{S}_v \cdot \nabla y_k) - (\mathbf{S}_v \cdot \nabla x_i)(\mathbf{S}_u \cdot \nabla y_i) \tag{6}$$

which can be expressed by dyadic products equivalent to the surface normal's cross-tensor as

$$\begin{aligned} \det(\mathbf{J}_i) &= \nabla x_i \cdot (\mathbf{S}_u\mathbf{S}_v\text{-}\mathbf{S}_v\mathbf{S}_u) \cdot \nabla y_i \\ &= -\nabla x_i \cdot [\mathbf{N}]_\times \cdot \nabla y_i = -|\mathbf{N}|\,|\nabla x_i \mathbf{n} \nabla y_i|, \end{aligned} \tag{7}$$

where $\mathbf{N}$ is the surface normal, $\mathbf{n}$ is the unit normal, and $|\nabla x_i \mathbf{n} \nabla y_i|$ is the triple scalar product of the gradients and the normal. Finally, we get [21]

$$\mathbf{J}_{ij} = \frac{1}{|\nabla x_i \mathbf{n} \nabla y_i|} \begin{bmatrix} |\nabla x_j \mathbf{n} \nabla y_i| \; |\nabla x_i \mathbf{n} \nabla x_j| \\ |\nabla y_j \mathbf{n} \nabla y_i| \; |\nabla x_i \mathbf{n} \nabla y_j| \end{bmatrix} \tag{8}$$

The above quantities are all invariant first-order differentials: the gradients of the projections and the surface unit normal vector. Note that (8) is a general formula: neither a special form of projections, nor a specific surface is assumed here, hence it can be applied for any camera type and for any reasonably smooth surface.

The formula derived above can be used for different purposes:

1. an affine transformation can be established between the images of a known surface using known projection functions;
2. if the projections are known and the parameters of the affine mapping acting between corresponding regions of a stereo image pair are estimated, then the normal of the corresponding 3D surface patch can be computed;
3. if the 3D surface normal is known and the affine mapping parameters are estimated, then the gradients of one of the projection functions can be computed.

Case 1) is addressed in [21]. Herein, we will show how to use this formula in case 2) for normal vector computing. Let us write the matrix components -

estimated either directly with affine estimator or taking the derivatives of an estimated homography - with:

$$\mathbf{J}_{ij\,est} = \begin{bmatrix} a_{11} \; a_{12} \\ a_{21} \; a_{22} \end{bmatrix} \tag{9}$$

To eliminate the common denominator we may use ratios, which can be constructed using either row, column, or cross ratios. Without loss of generality, we deduce the equation for the 3D surface normal using cross ratios:

$$\frac{\mathbf{n} \cdot (\nabla y_i \times \nabla x_j)}{\mathbf{n} \cdot (\nabla y_j \times \nabla x_i)} = \frac{a_{11}}{a_{22}}, \; \frac{\mathbf{n} \cdot (\nabla x_j \times \nabla x_i)}{\mathbf{n} \cdot (\nabla y_i \times \nabla y_j)} = \frac{a_{12}}{a_{21}} \tag{10}$$

After rearranging:

$$\begin{array}{c} \mathbf{n} \cdot [a_{22} \left(\nabla y_i \times \nabla x_j\right) - a_{11} \left(\nabla y_j \times \nabla x_i\right)] = 0 \\ \mathbf{n} \cdot [a_{21} \left(\nabla x_j \times \nabla x_i\right) - a_{12} \left(\nabla y_i \times \nabla y_j\right)] = 0 \end{array} \tag{11}$$

Here we have two (known) vectors, both perpendicular to the normal:

$$\begin{array}{c} \mathbf{p} = [a_{22} \left(\nabla y_i \times \nabla x_j\right) - a_{11} \left(\nabla y_j \times \nabla x_i\right)] \\ \mathbf{q} = [a_{21} \left(\nabla x_j \times \nabla x_i\right) - a_{12} \left(\nabla y_i \times \nabla y_j\right)] \end{array} \tag{12}$$

Thus the 3D surface normal can readily be computed as

$$\mathbf{n} = \frac{\mathbf{p} \times \mathbf{q}}{|\mathbf{p} \times \mathbf{q}|}$$

2.2 Specialization to perspective camera

Let us now apply our general results to the case of perspective cameras. The camera matrix of the i-th camera $\mathbf{P}^{(i)}$ is a 3×4 rank 3 matrix with row vectors $\pi_k^{(i)T} = \left[p_{k1}^{(i)} \, p_{k2}^{(i)} \, p_{k3}^{(i)} \, p_{k4}^{(i)} \right], \; k = 1, 2, 3$. Furthermore, spatial coordinates are represented as homogeneous four-vectors $\hat{\mathbf{X}} = \left[X \; Y \; Z \; 1 \right]^T$ and the projection functions become rational functions due to projective division:

$$x_i = \frac{\pi_1^{(i)} \cdot \hat{\mathbf{X}}}{s_i}, \; y_i = \frac{\pi_2^{(i)} \cdot \hat{\mathbf{X}}}{s_i} \tag{13}$$

with $s_i = \pi_3^{(i)} \cdot \hat{\mathbf{X}}$. Using these notations, the gradients become

$$\begin{array}{c} \nabla x_i = \dfrac{1}{s_i} \left[p_{11}^{(i)} - p_{31}^{(i)} x_i \; p_{12}^{(i)} - p_{32}^{(i)} x_i \; p_{13}^{(i)} - p_{33}^{(i)} x_i \right]^T \\ \nabla y_i = \dfrac{1}{s_i} \left[p_{21}^{(i)} - p_{31}^{(i)} y_i \; p_{22}^{(i)} - p_{32}^{(i)} y_i \; p_{23}^{(i)} - p_{33}^{(i)} y_i \right]^T \end{array} \tag{14}$$

Observing that each coefficient composed by cross product has exactly one gradient of projection i and one gradient of projection j, the scaled vectors $\mathbf{P} = s_i s_j \mathbf{p}$ and $\mathbf{Q} = s_i s_j \mathbf{q}$ yields the same result

$$\mathbf{n} = \frac{\mathbf{P} \times \mathbf{Q}}{|\mathbf{P} \times \mathbf{Q}|}$$

with denominators s_i, s_j eliminated.

2.3 Using homography

It is well known from projective geometry that images of a planar surface patch are related by planar homography, which is given by a 3 × 3 matrix realizing the mapping between homogeneous coordinates. It follows that if this matrix is known, one can accurately determine the affine parameters calculating the homography's partial derivatives. Denoting the components of the homography matrix $\mathbf{H}_{ij}$ with h_{kl} $(k,l=1,2,3)$ acting between images i and j, the elements of $\mathbf{J}_{ij}$ become

$$\begin{aligned} a_{11} &= \frac{1}{r}\left(h_{11} - h_{31}x_j\right),\ a_{12} = \frac{1}{r}\left(h_{12} - h_{32}x_j\right) \\ a_{21} &= \frac{1}{r}\left(h_{21} - h_{31}y_j\right),\ a_{22} = \frac{1}{r}\left(h_{22} - h_{32}y_j\right) \end{aligned} \tag{15}$$

with scale factor $r = h_{31}x_i + h_{32}y_i + h_{33}$.

2.4 Discussion

For the sake of simplicity we suppose that our cameras have zero skew (*e.g.* camera with CCD sensor), hence the calibration matrix is

$$\mathbf{K} = \begin{bmatrix} \alpha & 0 & x_0 \\ 0 & \beta & y_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{16}$$

The projection functions expressed in the camera coordinate system are then $x = \alpha\frac{X}{Z} + x_0,\ y = \beta\frac{Y}{Z} + y_0$ and the gradients (scaled by the common multipliers) become

$$\begin{aligned} \frac{1}{\alpha}Z^2\nabla x &= \begin{bmatrix} Z & 0 & -X \end{bmatrix}^T \\ \frac{1}{\beta}Z^2\nabla y &= \begin{bmatrix} 0 & Z & -Y \end{bmatrix}^T \end{aligned} \tag{17}$$

This result shows that ∇x is on the $Y = 0$ plane relative to camera (*i.e.* perpendicular to its momentary "up-down" direction), and perpendicular to the direction of the object's projection onto that plane. Similarly, ∇y is on the $X = 0$ plane (*i.e.* perpendicular to its momentary "left-right" direction), and perpendicular to the direction of the object's projection onto that plane.

Since we use these gradients in cross products, the "perpendicularities on the planes clause" can be lifted and finally we have a very important condition for the cross products involved in (8): they admit parallelism *if and only if the two camera centers and the observed point of interest are on the same line.* Note that in the case of video sequences, this parallax-less condition renders the motion toward the observed object as critical motion. Furthermore, the basic equations (8) include triple scalar products with surface normal involved. The parallelism of any gradient and the normal (the case to be excluded) means that the observed point is imaged as contour point.

Nevertheless, in practice the following algebraic consideration is usually sufficient. As discussed in Section 2.1, the normal can be expressed by three different ratios. Of course, in theory these ratios yield exactly the same normal vector: Clearly, taking row ratios is equivalent to column ratios of the inverse transformation and vice versa; while cross ratios are equivalent for both. In practice, however, affine parameters are subject to noise, inherent to any image processing algorithm, causing slight numerical differences in the normals provided by the 3 ratios. To choose the numerically most stable one, we recommend to follow this three-step procedure:

1. Determine the estimated transformation's Jacobian (see (9)) and choose the two components having the smallest absolute values
2. If these values are both significantly less than the next in order (element having the 3-rd smallest absolute value), then the equations expressed with that particular ratios (i.e. where either $\mathbf{P}$ or $\mathbf{Q}$ are close to zero vector) should be excluded from step 3
3. Choose the expression serving the biggest weighted value for $|\mathbf{P} \times \mathbf{Q}|$

The weight we recommend is based on our numerical experience and not yet theoretically investigated. According to that, expression based on cross ratio seems to be the most reliable and accurate in practice. Therefore if it is not dropped prior to step 3, we recommend its weighting with a greater number than 1 (say 4) as its "effective" magnitude for comparison. If more than two cameras are involved then we can repeat the above procedure to choose the most favorable camera pair.

2.5 Distance Calculation

While for the surface normal, only an affine mapping is needed between the image pairs, knowing the normal and the plane-induced homography allows us to determine the distance from an observed planar patch too. It is well known that a plane-induced homography encapsulates the plane's unit normal and perpendicular distance from the origin [2]. Furthermore, our homography matrix is a homogeneous entity, therefore the ratio of its any two components gives one equation for distance - leading to a highly overdetermined system which can be solved in the least square sense. Within the cameras' relative coordinate system, the world coordinate system can be canonically attached to one of the cameras, and the transformation between normalized points $\mathbf{X}_i = \mathbf{K}_i^{-1}\mathbf{x}_i$ in camera i and $\mathbf{X}_j = \mathbf{K}_j^{-1}\mathbf{x}_j$ in camera j can be described as

$$\mathbf{X}_j = (\mathbf{R} + \frac{1}{d}\mathbf{t}\mathbf{n}) \cdot \mathbf{X}_i \tag{18}$$

where d is the perpendicular distance of the plane to the camera center i, $\mathbf{R}$ and $\mathbf{t}$ are relative rotation and translation of the two camera coordinate frames, and $\mathbf{n}$ is the normal of the 3D plane. Using homogeneous coordinates, the above equation is satisfied up to an arbitrary non-zero scale factor, hence the homography

$\mathbf{H}$ can be expressed as

$$\mathbf{H} \cong d\mathbf{R} + \mathbf{tn} \tag{19}$$

Note that the only unknown of the above equation is d. In order to set the scale of the above relation, the last element of the homography matrix can be fixed to 1 by dividing $\mathbf{H}$ with its last element, assuming it is non-zero. If it would be 0 – which is theoretically possible – then $\mathbf{H}$ would map points to infinity, which is usually excluded by physical constraints in real applications.

When the camera poses are given in an arbitrary world coordinate frame, then relative rotation and translation can be computed as

$$\begin{aligned} \mathbf{R} &= \mathbf{R}_j \cdot \mathbf{R}_i^{-1} \\ \mathbf{t} &= \mathbf{R}_j \cdot (\mathbf{C}_i - \mathbf{C}_j) \end{aligned} \tag{20}$$

where $\mathbf{R}_i$ and $\mathbf{R}_j$ are the orientations while $\mathbf{C}_i$ and $\mathbf{C}_j$ are the positions of camera i and j in the world coordinate system. Furthermore, the surface normal in relative coordinates can be expressed in terms of its world coordinates $\mathbf{n}_w$ as $\mathbf{n} = \mathbf{R}_i \cdot \mathbf{n}_w$, and the distance $d = d_w - \mathbf{n}_w \cdot \mathbf{C}_i$, where d_w is the distance expressed in the world coordinate frame. Finally, when the estimated homography $\mathbf{H}_{i,j}$, mapping the corresponding regions from camera i to j, is given in unnormalized image coordinates, then $\mathbf{H}_{i,j} = \mathbf{K}_j \mathbf{H} \mathbf{K}_i^{-1}$, where $\mathbf{H}$ is from (19). We thus get the following general relation between the homography $\mathbf{H}_{i,j}$, camera and plane relative poses:

$$\mathbf{H}_{i,j} \cong (d_w - \mathbf{n}_w \cdot \mathbf{C}_i) \cdot \mathbf{R}_j \cdot \mathbf{R}_i^{-1} + \mathbf{R}_j \cdot (\mathbf{C}_i - \mathbf{C}_j)(\mathbf{R}_i \cdot \mathbf{n}_w) \tag{21}$$

The only unknown in the above equation is d_w, which can be obtained by minimizing the geometric error of the transferred points over the image regions:

$$\arg\min_{d_w} = \sum_p \|\mathbf{H}_{i,j}\mathbf{p} - \mathbf{A}\mathbf{p}\|^2 \tag{22}$$

where $\mathbf{A}$ is the right hand side of (21). The minimizer of the above expression is easily obtained in a closed form as the position of the zero first order derivative w.r.t. d_w.

3 Experimental Results

The proposed method was tested on an Intel i7 3.4GHz CPU with 8 GB memory. A total of 300 synthetic examples were generated by selecting 15 templates introduced by [18]. The camera intrinsic matrix was derived from a real world camera. The extrinsic parameters were randomly set with the orientation between $-\pi/6$ to $\pi/6$ for each axis, the translation chosen from -20 to 20 in x and y directions, and from -10 to -20 in z directions. The z component was set to be negative so the scene was in front of the camera. The normal of the plane was a random selection with the only assumption that it points out of the image plane.

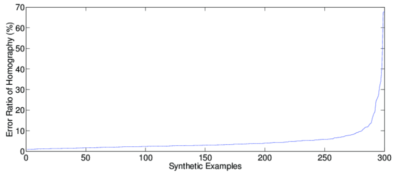


Fig. 1. Homography error for our synthetic dataset (the test cases are sorted on the x-axis).

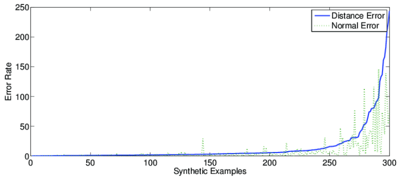


Fig. 2. Distance error and normal error plot for our synthetic dataset (test cases are sorted on the x-axis based on distance error)

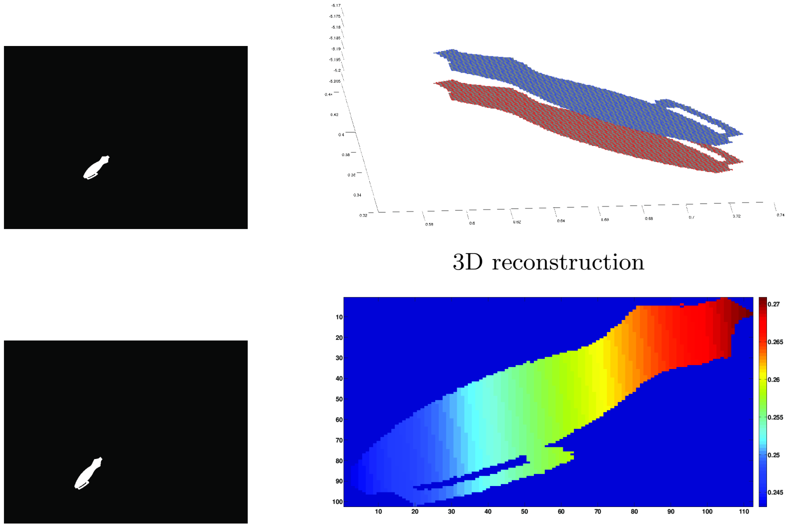
The first step of our algorithm is homography estimation between the corresponding region pairs. For that purpose, we use the correspondence-less method of [18] using the publicly available implementation. For a detailed evaluation of the method, see [18]. For reference, we show the homography error on our synthetic dataset in terms of the percentage of non overlapping area sorted in increasing order in Fig. 1. The registration method has less than 5% error for more than 250 examples. Obviously, this error directly affects the reconstruction error of our method - as we will see later.

Once the planar homography between the corresponding region pair is estimated, we can compute the 3D surface normal and distance using the closed form formulas derived in Section 2. A sample 3D reconstruction for synthetic data is shown in Fig. 3. The red surface is the ground truth surface and the blue one is the recovered surface. We also show the error map of the reconstruction. The color bar gives the index of the distance error in percentage - the error rate was less than 0.3%. The different colours also indicate that the two normals of the two surfaces are not perfectly parallel. Fig. 2 shows the error plots for the whole synthetic dataset. It is clear that distance error plot runs together with the normal error, hence our method provides reliable reconstructions for most test cases, giving low error rates for both surface parameters.

It is important to note that the proposed method can reach real time speed due to the closed form solution of the surface parameters.

3.1 Comparison with classical methods

Herein, we perform an experimental comparison of well known classical plane reconstruction methods and quantitatively demonstrate the performance of our



3D reconstruction

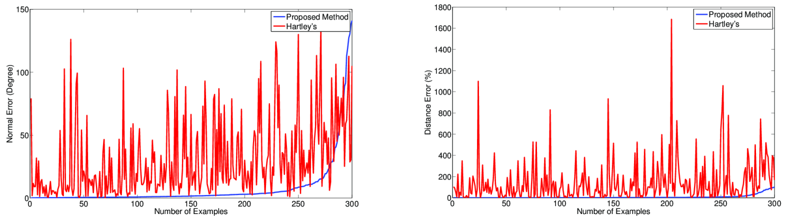
Input image pair

Reconstruction error

Fig. 3. 3D reconstruction of a synthetic image pair

method with respect to these algorithms on our synthetic dataset. The comparison is done first with the plane from homography method described by Hartley and Zisserman [2] for accuracy of the plane parameters, and then with the triangulation method of Fraundorfer *et al.* [16] for reconstruction accuracy. We remark, that Fraundorfer *et al.* originally use Harris corners for point correspondences to estimate homography in [16]. While the accuracy of their homography estimation (around 6%) is comparable to our method, we used the same homography for all methods to guarantee a fair comparison.

In Fig. 4, distance error and normal error plots are shown for the proposed method and the plane from homography direct method [2] (the Matlab code is available from `http://www.robots.ox.ac.uk/~vgg/hzbook/code/codevgg_plane_from_2P_H.m`). The purpose of this experiment is to compare our direct method derived via differential geometric considerations with a classical direct methods derived via projective geometric considerations, as a basis. Of course, in our experiments, we work with the estimated scene plane induced homography, which is theoretically correct but subject to numerical errors (see Fig. 1 for the homography errors). More than 200 examples gave less than 2.5% in distance error and 5 degree in normal error. Indeed, the proposed method performs an order of magnitude better than the classical method. Let us stress again that both methods used exactly the same input, so the results show the (very) different behaviour of these direct formulas in case of realistic image measurements! These experiments show that our direct method can tolerate slight errors in the homography, while the formula obtained via projective geometry is extremely sensitive to the smallest amount of numerical error, as it is also noted in [2].



Normal error plots Distance error plots

	D %	D% [2]	N degree	N degree [2]
Mean	7.5214	145.8588	8.0801	30.6649
Median	1.8717	76.0062	1.5540	17.9270

Fig. 4. Comparative error plots on our synthetic dataset with the plane from homography direct method [2] (test cases are sorted on the x-axis based on the proposed method's error). The table shows distance error D and normal error N statistics.

Would we use a preprocessing to remove the effect of measurement noise as recommended in [2], then both method would become more robust – at the price of an increased computational compexity, of course.

	Distance	Distance [16]
Mean	0.0358	0.2095
Median	0.0350	0.2008

Table 1. Reconstruction accuracy in terms of 3D point distance

Table 1 shows the mean and median 3D distance error of reconstructed points for the proposed method and the triangulation method of Fraundorfer *et al.* [16]. The proposed method performs again and order of magnitude better. In addition, our method recovers the whole surface patch in one step, while [16] gives only a point cloud for matched point pairs.

3.2 Robustness

Noise %	0	1	2	5	10	15	20
x	3.325	5.222	7.861	17.73	33.54	45.23	53.06
y	3.325	5.414	8.81	20.03	36.79	51.79	67.03
z	3.325	4.638	6.997	15.17	29.73	43.72	53.51

Table 2. Normal error w.r.t. rotation error in different axes

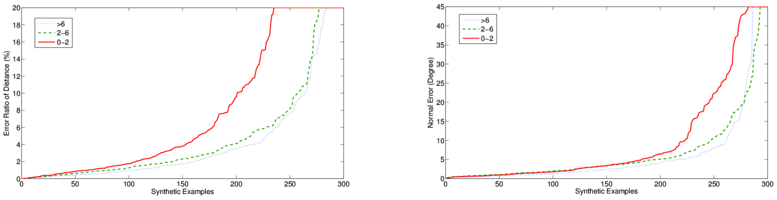
Noise %	0	1	2	5	10	15	20
x	1.706	2.642	3.891	8.753	15.91	19.6	20.69
y	1.706	2.624	4.072	9.495	16.48	20.98	23.92
z	1.706	2.005	3.501	6.703	14.14	19.13	20.55

Table 3. Distance error w.r.t. rotation error in different axes

Noise %	0	1	2	5	10	15	20
	1.706	2.264	3.362	7.109	14.53	22.27	29.75

Table 4. Distance error w.r.t. translation error

The accuracy of the proposed method depends not only on the quality of homography estimation, but also on the camera pose parameters which are used to compute relative rotation and translation as described in Section 2. Obviously, normal estimation is only affected by the rotation matrix, while distance calculation depends on both rotation and translation. To characterize the robustness of our method against errors in these parameters, we added various percent of noise to the original values and quantitatively evaluated the reconstruction error on our synthetic dataset. Table 3 and Table 2 show that normal is slightly more sensitive to this type of error, but its error is still below 10% up to 2% noise. Distance estimation can tolerate up to 5% noise in both rotation and translation.



Distance error w.r.t. different baselines Normal error w.r.t. different baselines

Fig. 5. Error plots w.r.t. different baselines (test cases are sorted on the x-axis based on the error).

Baseline is also an important parameter for 3D reconstruction. Short baseline is often seen in short sequence images such as video. With the distance to the plane set to be around 15 meters, 3 different baselines were tested. The shortest baseline is within $0-2$m, the medium one is between $2-6$m, and large baseline is considered larger than 6m. Fig. 5 shows the error with respect to each baseline range. Of course, shorter baseline has higher error rate, which is a well known fact for stereo reconstruction. In addition, our method is also affected by larger homography error in case of decreasing baseline (see Fig. 6). Nevertheless, the proposed method still have robust performance within a large range of baselines.

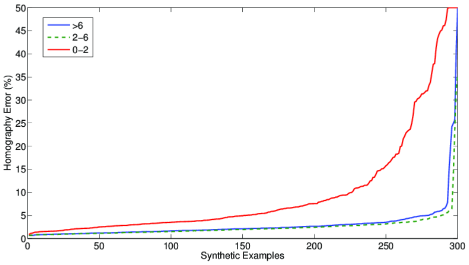


Fig. 6. Homography error w.r.t. different baselines

3.3 Real images

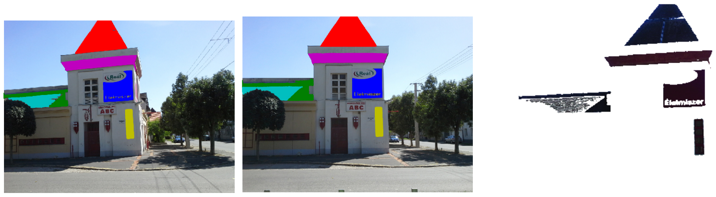


Fig. 7. 3D reconstruction result using MSER regions

Finally, we test our method on real world objects. There are various ways to extract corresponding regions from real image pairs. For example, we can use standard MSER regions - such a reconstruction result is presented in Fig. 7. Another possibility is to extract larger regions corresponding to building facades using *e.g.* color-based clustering such as in Fig. 8. Note that all real examples contain various patch orientations, the color labels denote corresponding image regions.

4 Conclusion

We proposed an efficient 3D reconstruction method, which allows the reconstruction of complete planar surface patches from a homography map between corresponding image regions and calibrated cameras. The theoretical foundation relies on variational calculus, which leads to a closed form solution for the surface normal and distance parameters. Being a direct solution, it runs in real-time which can be particularly useful for mobile and embedded vision systems. Another advantage is that it works without point correspondences by making use of segmented regions. Quantitative experiments on a large synthetic dataset confirm the superior performance w.r.t. classical plane reconstruction algorithms,

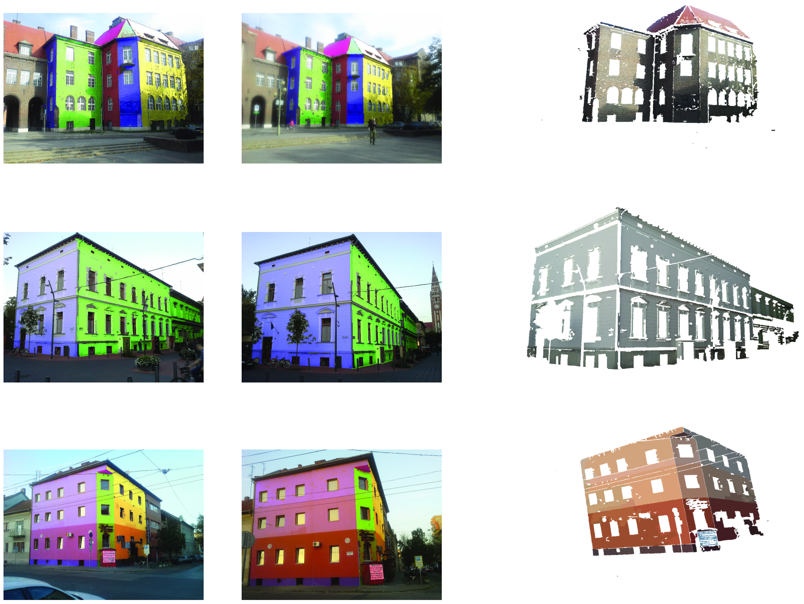


Fig. 8. 3D reconstruction results using regions extracted by color-based clustering

while reconstruction of whole building facades from real images confirm the applicability of our approach for real-life problems. In our future work, the focus will be on reliable planar segmentation methods for urban environments.

Acknowledgement

This research was supported by the European Union and the European Social Fund through project FuturICT.hu (grant no.: TAMOP-4.2.2.C-11/1/KONV-2012-0013).

References

1. Musialski, P., Wonka, P., Aliaga, D.G., Wimmer, M., van Gool, L., Purgathofer, W.: A Survey of Urban Reconstruction. In: EUROGRAPHICS 2012 State of the Art Reports, Eurographics Association (2012) 1–28
2. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Second edn. Cambridge University Press, ISBN: 0521540518 (2004)
3. Kazhdan, M., Bolitho, M., Hoppe, H.: Poisson surface reconstruction. In: Proceedings of the fourth Eurographics symposium on Geometry processing, Eurographics Association (2006) 61–70

4. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Gool, L.V.: A comparison of affine region detectors. Int. J. Comput. Vision **65** (2005) 43–72
5. Laurentini, A.: The visual hull concept for silhouette-based image understanding. IEEE Trans. Pattern Anal. Mach. Intell. **16** (1994) 150–162
6. Faugeras, O., Keriven, R.: Variational principles, surface evolution, pde's, level set methods and the stereo problem. IEEE TRANSACTIONS ON IMAGE PROCESSING **7** (1999) 336–344
7. Fuhrmann, S., Goesele, M.: Fusion of depth maps with multiple scales. ACM Trans. Graph. **30** (2011) 148:1–148:8
8. Furukawa, Y., Ponce, J.: Accurate, dense, and robust multiview stereopsis. IEEE Transactions on Pattern Analysis and Machine Intelligence **32** (2010) 1362–1376
9. Habbecke, M., Kobbelt, L.: Iterative multi-view plane fitting. In: In VMV06. (2006) 73–80
10. Habbecke, M., Kobbelt, L.: A surface-growing approach to multi-view stereo reconstruction. Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on (2007) 1–8
11. Kannala, J., Brandt, S.: Quasi-dense wide baseline matching using match propagation. Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on (2007) 1–8
12. Furukawa, Y., Ponce, J.: Accurate, dense, and robust multi-view stereopsis. In: CVPR. (2007) 1362–1376
13. Sinha, S., Steedly, D., Szeliski, R.: Piecewise planar stereo for image-based rendering. Computer Vision, 2009 IEEE 12th International Conference on (2009) 1881–1888
14. Kowdle, A., Chang, Y.J., Gallagher, A., Chen, T.: Active learning for piecewise planar 3d reconstruction. In: Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition. CVPR '11, Washington, DC, USA, IEEE Computer Society (2011) 929–936
15. Hiep, V.H., Keriven, R., Labatut, P., Pons, J.P.: Towards high-resolution large-scale multi-view stereo. Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on (2009) 1430–1437
16. Fraundorfer, F., Schindler, K., Bischof, H.: Piecewise planar scene reconstruction from sparse correspondences. Image Vision Comput. **24** (2006) 395–406
17. Zhou, Z., Jin, H., Ma, Y.: Robust plane-based structure from motion. In: Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). CVPR '12, Washington, DC, USA, IEEE Computer Society (2012) 1482–1489
18. Domokos, C., Nemeth, J., Kato, Z.: Nonlinear shape registration without correspondences. IEEE Trans. Pattern Anal. Mach. Intell. **34** (2012) 943–958
19. Devernay, F., Faugeras, O.: Computing differential properties of 3-D shapes from stereoscopic images without 3-D models. In: Proceedings of International Conference on Computer Vision and Pattern Recognition. (1994) 208–213
20. Jones, D.G., Malik, J.: Determining three-dimensional shape from orientation and spatial frequency disparities. In Sandini, G., ed.: Proceedings of European Conference on Computer Vision. Volume 588 of Lecture Notes in Computer Science., Springer Berlin Heidelberg (1992) 661–669
21. Molnar, J., Chetverikov, D.: Quadratic transformation for planar mapping of implicit surfaces. Journal of Mathematical Imaging and Vision (2012) 1–9