Electronic Letters on Computer Vision and Image Analysis 11(1):28-40, 2012

Two View Line-Based Motion and Structure Estimation for Planar Scenes

Saleh Mosaddegh*, David Fofi* and Pascal Vasseur+

* *Le2i, Université de Bourgogne, UMR CNRS 6306, Le Creusot, France*
+ *LITIS, Université de Rouen*
+ *MIS, Université de Picardie Jules Verne*

Received 22nd Jun 2011; accepted 27th Jul 2012

Abstract

We present an algorithm for reconstruction of piece-wise planar scenes from only two views and based on minimum line correspondences. We first recover camera rotation by matching vanishing points based on the methods already exist in the literature and then recover the camera translation by searching among a family of hypothesized planes passing through one line. Unlike algorithms based on line segments, the presented algorithm does not require an overlap between two line segments or more that one line correspondence across more than two views to recover the translation and achieves the goal by exploiting photometric constraints of the surface around the line. Experimental results on real images prove the functionality of the algorithm.

Key Words: Structure and Motion, piece-wise planar scene, line correspondence, two views.

1 Introduction and related work

Generally speaking, motion estimation is the second main stage of Structure from Motion or SfM process after the feature matching and before the final reconstruction. Wide baseline motion estimation from point correspondences between two views has been the subject of much investigation and even though this is still a very active field of research, many fast, simple and efficient methods have been proposed in such studies. On contrary, motion estimation from line correspondences between two wide baseline views has not received much attention. The reasons for this are manifold. First, line segments are more difficult to match. Besides the traditional challenges in point matching, these difficulties proceed from some other different reasons such as the inaccuracy of the endpoint extraction, fragmented segments, the poor geometric disambiguity constraint, lack of significant photometric information in the local neighborhood of segments and no global geometric constraint such as the epipolar constraint. As a result, up to now, only a few methods are reported in the literature for automatic wide baseline line segment matching [1, 2, 3, 4, 5, 6]. Secondly, classical methods such as [15, 8] which use supporting lines (geometric abstraction of straight line segments), need many line correspondences across at least three images while for only two views, one can find several works in literature

Correspondence to: <saleh.mosaddegh@u-bourgogne.fr>

Recommended for acceptance by <Giorgio Fumera>
ELCVIA ISSN:1577-5097
Published by Computer Vision Center / Universitat Autònoma de Barcelona, Barcelona, Spain

in which the impossibility of motion determination from the line correspondences between only two views are mentioned [16, 8, 11].

However, using lines is advantageous because lines are easier to extract and have less localization error than other features of interest such as points. Moreover, their detection is very reliable according to their orientation. Most keypoints hardly capture geometrical and structural information of the scene since they are not localized at edges while lines are directly connected to geometrical information about the scene. We are especially interested in the case where the motion of the camera is a large baseline motion. For such scenario, classical methods are of minor applicability mainly because finding enough number of line matches between several views is a very difficult task if not impossible. For such cases, algorithms which can work with two images become a better choice. Instead of relying on many correspondences, we present a method which takes only one line correspondence over only two views as input, and suggest a two stage algorithm which uses lines in man-made scenes to recover rotation and then estimates the translation.

To our knowledge, the algorithm introduced by Zhang [17] is, so far, the only work on motion estimation based on only two views of only line segments. The algorithm tries to recover the motion using the epipolar geometry by maximizing the total overlap of line segments in correspondence as an objective function with no closed-form solution hence a five-dimensional motion space (three for rotation and two for the translation) has to be sampled. Unlike Zhang's algorithm, in our method only one line correspondence is enough and it is not necessary that two matched line segments overlap. Besides, Zhang's algorithm has this main drawback that it needs relatively large set of correspondences of line segments randomly distributed and oriented in the scene. Though the number of extracted line segments from an image of a constructed scene can be relatively high, matching them is a very difficult task especially if the motion has a long baseline. Moreover, in constructed scenes, the assumption of randomly oriented lines may not hold since the majority of the lines are pointing in the same direction as one of three main Manhattan directions. The only assumptions we use is that the line belongs to a planar surface with enough strong discriminative texture. Thanks to the presence of many parallel lines in man-made scenes, the rotation can be computed from other methods based on matching vanishing points [10, 9]. The assumption of the textured surface is also very likely to happen in constructed scenes. To our best of knowledge, this is the first attempt to estimate large baseline motion using lines and photometric constraints between two images.

2 The methodology

Using both geometric and photometric constraints of the line and its image neighborhood and reconstructing the 3D surface around the line is the key idea of our method. For each view, a set of planar facets passing through the 3D line in space are hypothesized and the plane hypothesis which shows higher similarity between two views is chosen to be the best reconstruction of the surface (see figure 1). The camera translation is simultaneously recovered during the construction of the surface.

For the rest of this text, we assume that except two end points of line segments which are expressed in image plane frame, the rest of values and parameters are expressed in the first camera coordinate system which is chosen to be aligned with the world coordinate system with no loss of generality. The intersection of two planes passing through each segment and the origin of the related camera results in l, the direction of the 3D line:

$$l = \frac{n \times R^{-1}n'}{\|n \times R^{-1}n'\|} \tag{1}$$

where n and n' are known images of the 3D line on the image planes (*i.e.* normal vector of the plane passing through the 3D line and the origin of the camera) and R is the camera rotation which can be computed based on matching vanishing points (see below).

The fully reconstruction of the line, however , is not possible since at least the ratio of the distances of the line to the origins of the cameras should be known. T, the translation from first camera to the second camera

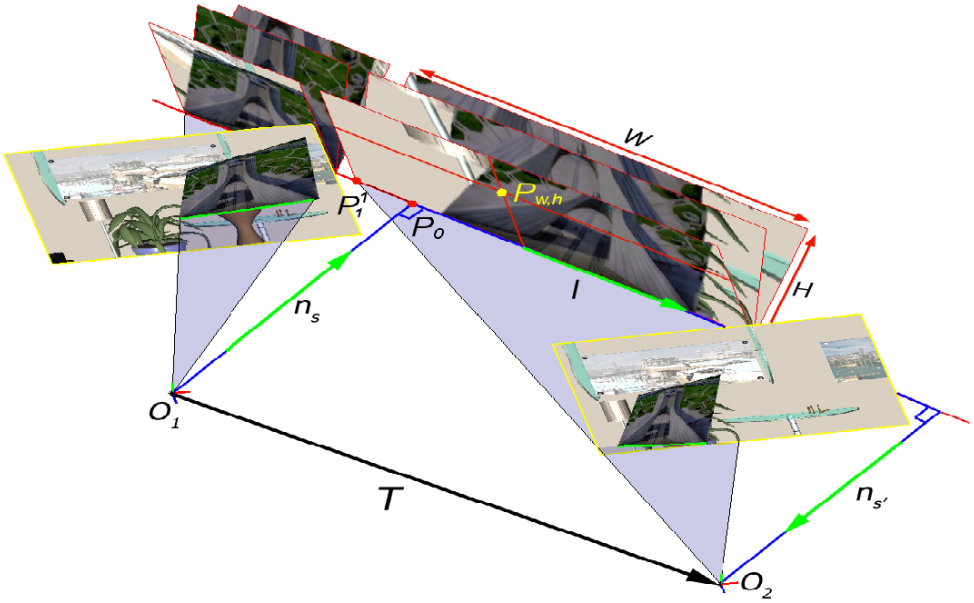


Figure 1: Geometry of proposed method. On each hypothesized plane (three planes are shown here), one rectangular mesh is built on the same side of the 3D line from each camera image.

(we are using t and capital T to distinguish between the normalized and un-normalized translation vectors respectively) can be decoupled in three vectors and be expressed by (see Fig. 1):

$$T = n_s s + l s_l - n_{s'} s' \; , n_s = \frac{l \times n}{\|l \times n\|} \; , n_{s'} = \frac{l \times R^{-1} n'}{\|l \times R^{-1} n'\|} \tag{2}$$

where s and s' are the unknown distances between 3D line and the origins of the cameras and s_l is an unknown scalar value. Therefore in order to determine the translation of the system we need to find these three values. Since the final solution will always be up to a positive scale, we can set one of these values to a fixed value (we chose s). Now the problem is simplified to estimating the other two scalars. To estimate these two values we need to use the photometric information around the line. However any comparison between the areas around the images of the line in two views is meaningless due to perspective distortion. To overcome this difficulty we reconstruct the surface and therefore the texture on it from each of images by sweeping a hypothesized plane through space which pass through 3D line and find the plane which best matches two reconstructed surfaces. In the section 2.2, we drive the necessary formulas for the parametrized reconstruction of the 3D surface and its texture seen by each camera, referred to as the "mesh image" hereinafter.

2.1 Recovering R

For our method to work, we first need to recover the rotation between two views. For obtaining the rotation directly, one may use an inertial measurement unit (IMU). These sensors are now in smart phones, in cameras and on most of the robotic platforms (see for example [18] or [19]). For estimating the rotation after extracting and matching vanishing points, one can choose among many algorithms available for the perspective images (cf. [10] for a review on some generic methods and their pros and cons). In this work, we use the method of [10] in order to extract and match some vanishing points. Having these vanishing point correspondences , the relative rotation between two views can be computed using the simple linear method of [9]. Assume V_1 and V_2 (and their corresponding V_1' and V_2') are unit vectors corresponding to two vanishing directions. R can be decomposed into a rotation axis N and an angle θ which can be recovered as follows:

- if $V_i' = V_i$, $i \in [1, 2] \Rightarrow R = I$.

- if $\begin{cases} V_i' = V_i & i,j \in [1,2] \\ V_j' \neq V_j & i \neq j \end{cases} \Rightarrow \begin{cases} N = V_i \\ cos\theta = \frac{V_j'.V_j - (V_j.N)^2}{1-(V_j.N)^2} \\ sin\theta = \frac{V_j'.(N \times V_j)}{||N \times V_j||^2} \end{cases}$

- if $V_i' \neq V_i$, $i \in [1,2] \Rightarrow \begin{cases} N = \frac{(V_i - V_i') \times (V_j - V_j')}{||(V_i - V_i') \times (V_j - V_j')||} \\ cos\theta = \frac{V_j'.V_j - (V_j.N)^2}{1-(V_j.N)^2} \\ sin\theta = \frac{V_j'.(N \times V_j)}{||N \times V_j||^2} \end{cases}$

where I is the 3×3 identity matrix. Finally, using Rodrigues' formula, rotation matrix R can be computed as:

$$R = I + sin\theta[N]_\times + (1 - cos\theta)[N]_\times^2$$

where $[N]_\times$ is the skew-symmetric matrix corresponding to vector N.

2.2 Building 3D rectangular meshes and mesh images

For each image, the steps of building a 3D rectangular mesh with the orientation n_m on one side of the 3D line are as follows:

Let p_i^j denote the coordinates of i^{th} end point of the line segment in the image plane of the j^{th}camera and P_i^j denote its corresponding Cartesian coordinates on the 3D line with respect to the world coordinate system (see Fig.1). After some calculations, P_i^j can be expressed by (here T, as a superscript, means transpose):

$$P_i^j = \frac{n_m^T \left(P_0 - O_j\right) \Omega}{(n_m^T \Omega)} + O_j \ , \tag{3}$$

$$\Omega = R_j \left(p_i^j + \begin{bmatrix} -u_j & -v_j & f_j \end{bmatrix}^T \right) \begin{matrix} i = 1,2 \\ j = 1,2 \end{matrix} \tag{4}$$

where (u, v), O and f are principal point, focal point and focal length of the cameras, respectively. R_j are camera rotations with respect to the world coordinate system. Since first camera coordinate system is aligned with the world coordinate system, we have $O_1 = [0\ 0\ 0]^T$, $R_1 = I$ (*Identity matrix*) , $R_2 = R$ and $O_2 = T$ (since O_2 is now the translation vector between two camera positions). The 3D point P_0 is a point on the 3D line, and it can be chosen to be the closest point of the line to the origin of the first camera: $P_0 = n_s s$. For each image, $P_{w,h}^j$, the 3D coordinates of the mesh at location (w, h) is:

$$P_{w,h}^j = P_1^j + wl + h\frac{l \times n_m}{\|l \times n_m\|} \ , \ w = 0 : W^j \ , \ h = 0 : H^j \tag{5}$$

Mesh resolution should be carefully selected since it depends on the scale of whole reconstruction which in return is determined by the value chosen for s. W^j should be chosen proportional to the distance between P_1^j and P_2^j in order to simplify the registering of two meshes at later steps of the method:

$$W^j = round(k \left\| P_1^j - P_2^j \right\|) \tag{6}$$

where k is a positive value. The higher the k is, the higher the resolution of the mesh is and therefore the reconstruction of the surface is more accurate in the expense of higher computational time.

H^j, the height of the mesh must be high enough to include neighboring texture around the segment, otherwise during the registration, the similarity measuring function can fail to match mesh images.

After some calculations, $p_{w,h}^j$, the corresponding image pixel for each mesh vertex $P_{w,h}^j$ can be expressed by:

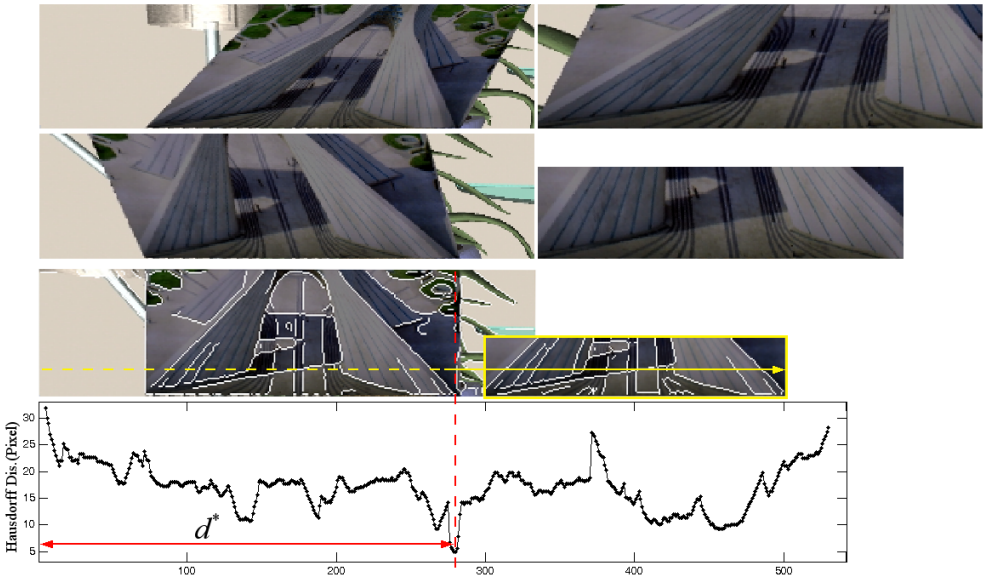


Figure 2: Two top rows: 2D mesh images where orientation of the surface and depth ratio are different from the ground truth. Two bottom rows: As the second mesh image sweeps the first mesh image, the Hausdorff distance between points of overlapping parts is computed (point sets are shown by white pixels). The best estimation of the surface orientation and s' occurs to have the lowest Hausdorff distance. The location where this smallest value occurs, d^*, is used to register two meshes in 3D and therefore recovering s_l.

$$p^j_{w,h} = [\Delta_x + u_j \ \Delta_y + v_j]^T \ , \ \begin{matrix} w = 0 : W^j \\ h = 0 : H^j \end{matrix} \ , \ j = 1, 2 \tag{7}$$

$$\Delta = R^{-1} \left(\frac{f_j}{r_3^T \left(P^j_{w,h} - O_j \right)} \left(P^j_{w,h} - O_j \right) \right) \tag{8}$$

where r_3 is the third column vector of rotation matrix R. Fig. 2 shows examples of what mesh images look like as the parameters change. Only for ground truth parameters, the surface texture in both mesh images are identical. Note that one of the strips is extended up to the image border, therefore, the algorithm works as long as a part of the surface attached to the line is seen by both cameras and it is not necessary that the segments overlap.

2.3 Estimation of surface orientation, depth ratio and T

After setting s to an arbitrary value and choosing the mesh resolution accordingly, the best values of scalars s' and s_l and also the orientation of the surface which minimizes the Hausdorff Distance between two point sets (extracted from mesh images using edge detectors such as Canny) must then be estimated in order to recover T (up to a scale factor) from the equation 2. Refer to the section 4 for a justification on using Hausdorff Distance as the similarity measuring function. This needs a brute force algorithm for searching among all possible values for these variables which is computationally expensive. Fortunately, the problem can be formulated in a simpler way and computational burden can be greatly reduced by observing the following facts from the geometry of the proposed method:

- s_l can be set to zero since its value has no effect on retrieving the mesh images (*i.e.* image pairs of Fig. 2 do not change as value s_l changes). However its real value is necessary for computing T and it will be recovered through registering two meshes in 3D (Eq. 9).

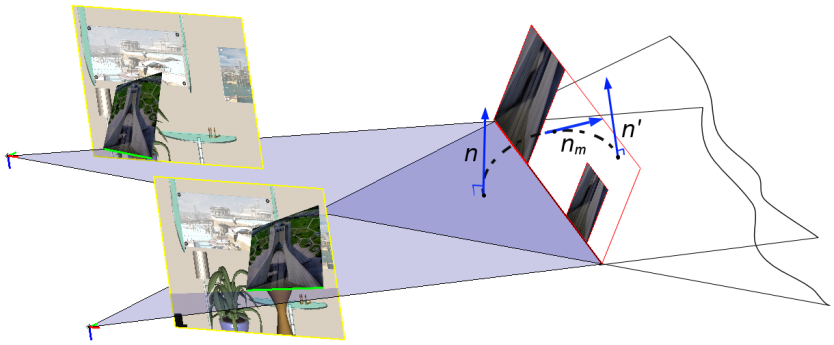


Figure 3: The searching space for the surface ground truth orientation. Instead of searching all potential surface orientations, a smaller set of likely surface normals varying between n and n' should be considered.

- Changing depth values s and s' has a zooming effect on their mesh images (*i.e.* doubling the distance between camera origin and the 3D line results in a twice larger mesh image). This suggests that in order to reduce computational time, instead of reconstructing from scratch, the second mesh image can be built just once by setting s' to its maximum expected value and then the effect of reducing s' can be simulated by resizing the second mesh image using interpolation. We limit the possible depth ratios $\frac{s}{s'}$ to the range $[\frac{1}{3}\ 3]$ in order to restrict the search and chose 20 equi-spaced points on this interval. In practice, this ratio is not likely to exceed this range.

- The searching space for the surface ground truth orientation, n_m, can also be reduced by taking into account that not all orientations of the hypothesized plane can generate proper image on the same side of the line segment. For details see Fig. 3.

Pseudo-code algorithm 1 summarizes our method for simultaneous estimation of surface orientation and depth ratio. The combinatorial functions $index(min(M))$ return the index corresponding to the smallest value in the matrix M.

Algorithm 1 The proposed motion and structure algorithm.

1: Given two images, extract their line segments and, manually or by using an automatic method, match a line segment between two views belong to the surface under reconstruction;
2: Estimate R using vanishing points (cf. section 2.1);
3: $s \leftarrow$ An arbitrary positive value;
4: **for** each $n_m(i)$ between n and n' **do**
5: **for** each $\frac{s}{s'}(j)$ between $\frac{1}{3}$ and 3 **do**
6: - Construct two mesh images and extract edge points;
7: - $M(i,j) =$ MIN(Hausd. Dis. between two point sets);
8: **end for**
9: **end for**
10: $(i^*, j^*) \leftarrow$ index(min(M));
11: **return** $n_m(i^*)$ and $\frac{s}{s'}(j^*)$;

The surface orientation n_m and the depth ratios $\frac{s}{s'}$ corresponding to the smallest distance in the matrix M inside nested loop are the best estimations of these two parameters. Note that the nested loop is very fast since not only the template and image to be searched are small, but also the whole image is not searched. Since it is not clear that which side of the line is planar, we run the algorithm for both sides and chose the side which gives a better reconstruction (*i.e.* lowest Hausdorff distance). If the Hausdorff distances for both sides are enough small, then there is a high chance that either the line is the intersection of two planar surfaces or it is located inside the surface instead of its boundaries (in the later case two computed orientations match).

After estimating n_m and s', computing s_l is straightforward. Let denote $d^\star$ as the distance at which Hausdorff distance between two mesh images corresponding to the best estimation of s' and the surface orientation occurs to be minimum as shown in Fig. 2. After some simple geometric considerations, s_l can be expressed by:

$$s_l = P_0 - P_1^1 + d^\star l + \left(P_1^2 - P_0 - W^2 l\right)\left(s/s'\right) \tag{9}$$

Knowing all the three scalars, T can be computed by equation 2.

3 More than one line correspondences

Theoretically, one line correspondence on a textured surface is enough to estimate the translation. While it is noteworthy that the approach works from only a single correspondence, in practice one can usually determine multiple good correspondences and combining the estimates from all available correspondences can help to verify and refine the accuracy of the estimated translation as well as reconstructing more planar surfaces of the scene. Assume T_i and T_j are two such estimations and $\frac{s_i}{s'_i}$ and $\frac{s_j}{s'_j}$ are corresponding depth ratios. It is clear that:

$$\frac{T_i}{\|T_i\|} = \frac{T_j}{\|T_j\|}\ ,\ \frac{\|T_i\|}{\|T_j\|} = \frac{s_i}{s_j} = \frac{s'_i}{s'_j} \tag{10}$$

The first constraint implies that the direction of the translation is identical for each line correspondence. Even though translation vectors computed by each line correspondence have different magnitude (due to different scales) but they should have the same direction. This constraint can provide a method for verifying the other estimated translation directions. One can also improve the accuracy of the final translation vector by computing the vector which has the minimum deviation from all the estimations. The second constraint relates the overall scale between two line (surface) reconstructions and can be used to reconstruct all planar surfaces of the scene in one uniform framework (refer to the Fig. 8 for an example of application).

4 Discussion

The bottleneck of our proposed method is finding the location of one of the mesh images in the other one. This is simply well-known template matching problem for which there are numerous number of methods available in literature with different cons and pros. These methods are usually different in the level of invariability to various deformation such as translation, rotation, scale, affine or perspective. Due to the nature of our problem in this stage, we are looking for the most simplest similarity measuring function which should not be invariant to any deformation except translation (to escape any false positive matches since we want the function to show a high similarity only when the surface reconstruction is corresponding to the ground truth). One choice is ZNCC* which is very simple but as it is shown in [14], it increases the sensitivity of the algorithm to the error in the estimation of rotation since this function is very sensitive to the pixel displacement which is inevitable during forming mesh images. Therefore we employed a generalized Hausdorff Distance which has been shown to work well in comparing images and it is computationally efficient when the template undergoes a simple translation [13]. It also can deal with individual pixel displacement errors while remains sensitive to overall mesh images deformation.

It worth mentioning that, in order to compare the surrounding of the line segments in two views, we also approached the problem in a similar way to what plane-sweeping methods do for feature matching and depth recovery using homography induced by each plane hypothesis [12] and we ended up estimating 3 unknowns (instead of 2 for our proposed method) plus a higher computational cost. One also may notice that our work is very different from plane-sweeping approaches which use the already known motions between several views

*Zero-mean Normalized Cross Correlation

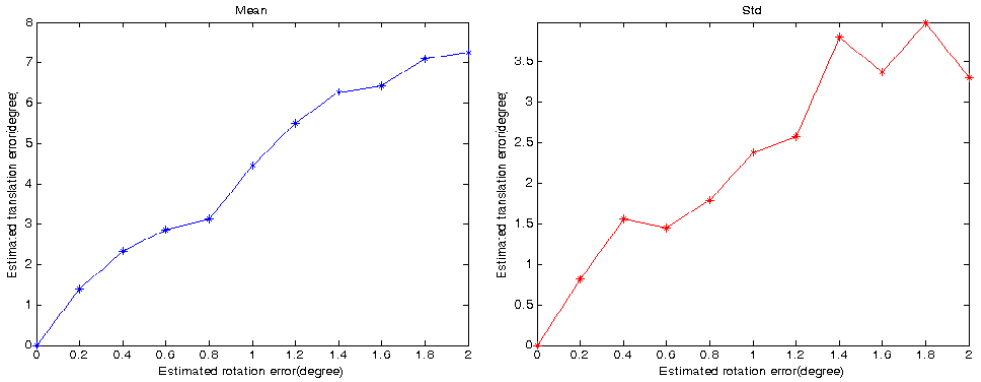


Figure 4: The mean/std error for the estimation of the translation direction.

of a scene for 3D reconstruction while our method uses lines first to recover the motion from only two views and at the same time to reconstruct the surfaces of the scene.

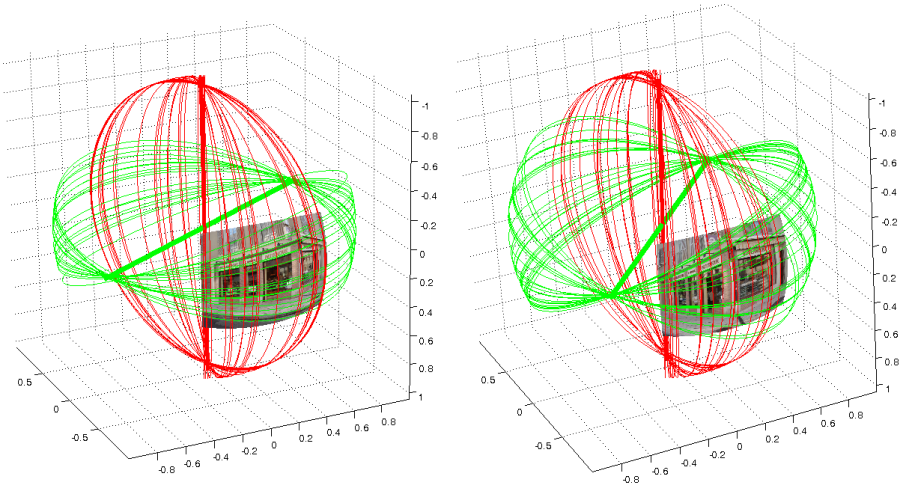
We consider our approach as a simple and efficient method for dense reconstruction of a planar object or scene from two images taken with considerably long baseline motion and with minimum need for the user interaction compared to other methods. Using methods such as Zhang's method [17] or multiple-view methods, we are only able to reconstruct 3D line segments and not any surfaces, therefore our method has the advantage of dense reconstruction. Also note that since we reconstruct a line and part of the surface attached to it, any other line coplanar with the surface is also reconstructed and we do not need to match and reconstruct these lines anymore (for example each wall of the bakery and the pavement attached to it (Fig. 8) are reconstructed using only one line from their intersection and we do not need to deal with the rest of the lines on these two surfaces). On the contrary, these coplanar lines and also parallel lines can decrease the performance of the methods which rely on only geometric information, for example Zhang's algorithm which is highly dependent on the random distribution of the line segments in the scene.

4.1 Sensitivity to the error in estimating R

While reconstructing two meshes, the estimated rotation between two views plays a critical role. Generally speaking, the error in the estimation of the rotation can have considerable effect on the estimation of the translation and this is always the case when estimating the displacement by a decoupling approach. Line extraction stage can also introduce some error in the direction and location of the segments on the image plane. In order to find the extend of the sensitivity of the recovered translation to these errors, we have carried out *10* set of *100* experiments with simulated image pair of Fig.1 and instead of identity matrix, we set R to a rotation matrix with an arbitrary rotation axis and a rotation angle around this axis which increases between each set by *0.2* degrees (*i.e.* in the 10^{th} set of experiments, there is a *2* degrees rotation between two views). The results of the experiments are shown in Fig. 4.

In general, the introduced error increases as the angle of rotation increases but this is not true for all experiments. This reflects the fact that the angles close to *90* degrees between axis of rotation and the 3D line may cause greater error in the estimation of the translation than the increase in the angle of rotation. This is true because in this case, the highest deformation on the mesh images occurs which can easily affect the performance of the similarity measuring function in correctly registering the two mesh images. Therefore a better similarity measuring function is advised if the error in the estimation of the rotation is expected to be high.

(a)



(b)

Figure 5: (a) Two real images used to estimate the motion of the camera. (b) Unitary spheres after projecting the images onto them and extracting vertical and horizontal vanishing points. For a better visualization only a small percentage of lines are shown.

5 Experimental results

To prove the accuracy of the method, we tried our algorithm on simulated images such as images of Fig. 1 in which there is no rotation between two views. Note, however, that there are still few sources of error such as the error in the position and direction of extracted line segments. Despite these errors, the translation was estimated almost without error (the angle between estimated direction of the translation and ground truth was *0.08* degrees).

We have tested our algorithm with success on several real image pairs taken from urban scenes. Fig. 5(a) shows two real images taken from a bakery which is a representative example of a mainly piecewise planar structure. The camera has a generic motion between two views. The position and rotation of the second camera with respect to the first one was obtained through a careful setup and use of a gyroscope:

$R = [-0.0073, -0.3049, -0.0036]$, $t = [0.9318, -0.0123, 0.3629]$,

where the translation t is normalized and the rotation R is represented by a 3D Rodrigues' vector (whose direction is that of the rotation axis and whose norm is equal to the rotation angle in radian).

We applied our algorithm and Zhang's algorithm, for comparison, on this data. For latter one, we extracted and matched 85 lines between two views manually. The minimum samplings required for this method to converge were 90 sampling of translation space and 728 sampling of rotation space. Only 1 of 10 best samples converged to the good solution. This may reflect the fact that the computed overlap between segments is not

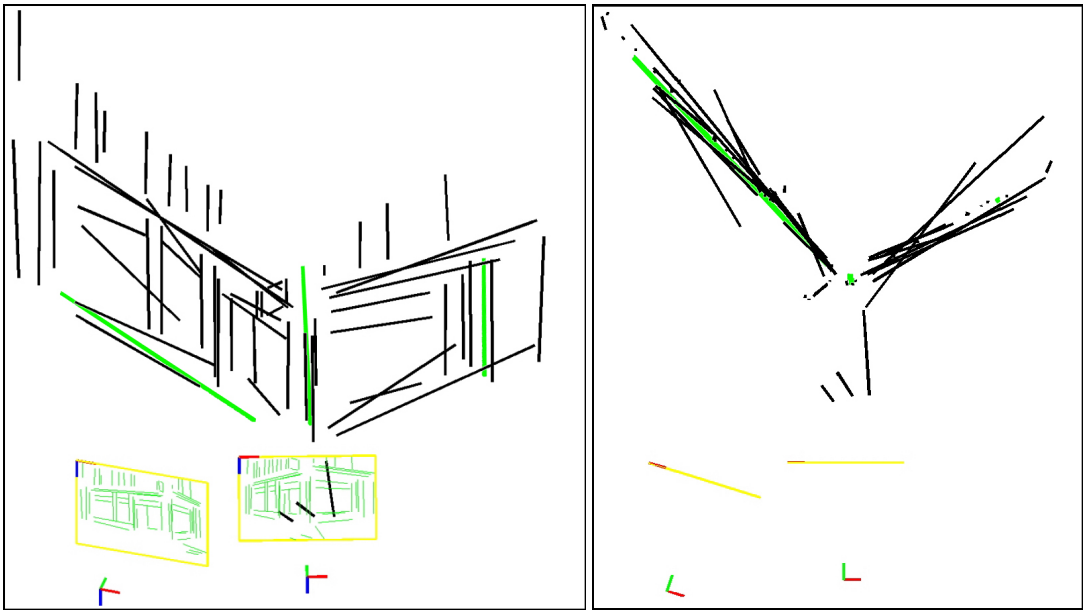


Figure 6: 3D reconstruction of the bakery by the structure from motion technique described in [17]. The right column corresponds to the top view.

correct when there are segments that are almost parallel to their epipolar lines. This is the case for many horizontal segments of this example. The result of the best solution is shown in Fig. 6. The error in the translation direction is $1.0847°$. The error in the rotation angle is $0.3087°$ and the error in the rotation axis is $1.9147°$. The motion is well estimated but, as it can be seen from the figure, the reconstruction is not very accurate.

In our algorithm, we use all lines which can automatically be extracted from each image (without need for matching them) to recover the rotation, only one line correspondence to recover the translation and more two line correspondences to verify and refine the results. To estimate the rotation between two views, we used the approach suggested in [10] . Fig. 5(b) shows projection of two images on unitary sphere. Dominant directions (vertical and one horizontal) of lines in the scene were used to estimate the rotation of the camera sufficiently accurate. The error in the rotation angle is $1.103°$ and the error in the rotation axis is $2.074°$.

For estimating the translation, we chose 30 equi-spaced points on the searching interval for the surface orientation. The output of our algorithm for reconstruction of 3 scene surfaces and 3 estimated translation vectors associated to each surfaces is shown in Fig. 7.

All three surfaces plotted in one uniform framework are shown in Fig. 8. The worst estimated translation error among three is $4.7°$. The estimated motion is comparable with the Zhang's algorithm and as it can be seen from Fig.8, the immediate result of the algorithm is a more useful reconstruction of the scene. In order to test the accuracy of the reconstruction, we took a few concrete distance and angle measurements (shown with blue arrays) and compared them with the result of the reconstruction. The worst estimated angle error between planes is less than $3°$ and the difference in the distance between landmarks is not exceeding 2.4%. For comparison, in Fig. 6, we also plotted (in green color), the three lines segments reconstructed by our algorithm.

Careful reader may notice that the accuracy of motion estimation and reconstruction of the proposed algorithm should not be compared with multiple-view-based methods such as [15, 8, 7] which are generally more accurate but need extraction and matching of many line(point) correspondences between more than two views, a very difficult task especially if the motion has a long baseline. For piece-wise planar scenes, our algorithm outperforms such generic methods in the sense of speed and accuracy of results are also comparable.

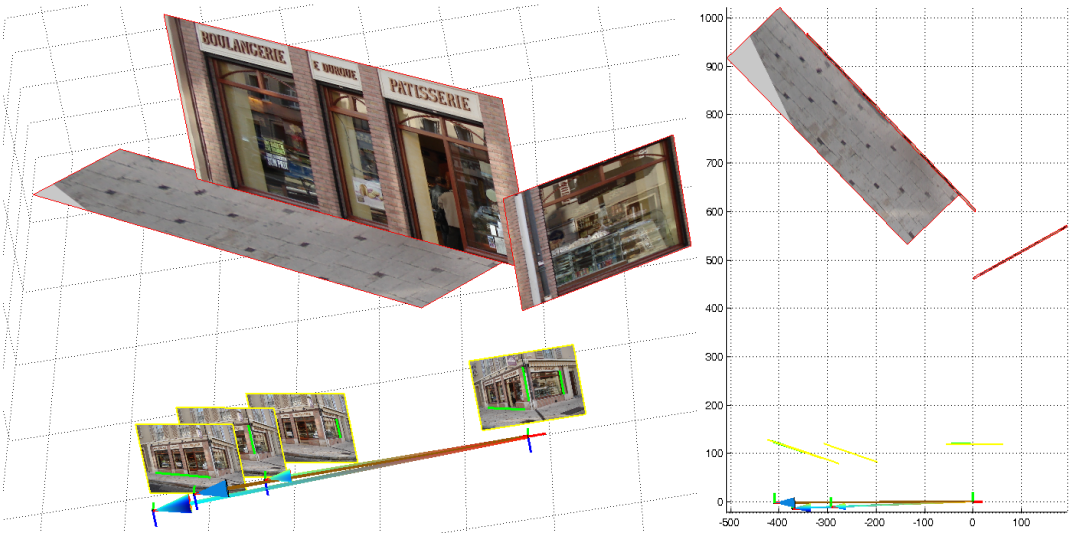


Figure 7: Reconstruction of the pavement and two walls of the bakery and 3 estimated translation vectors related to each surfaces. Scale of each surface reconstruction is different from others. The right image corresponds to the top view.

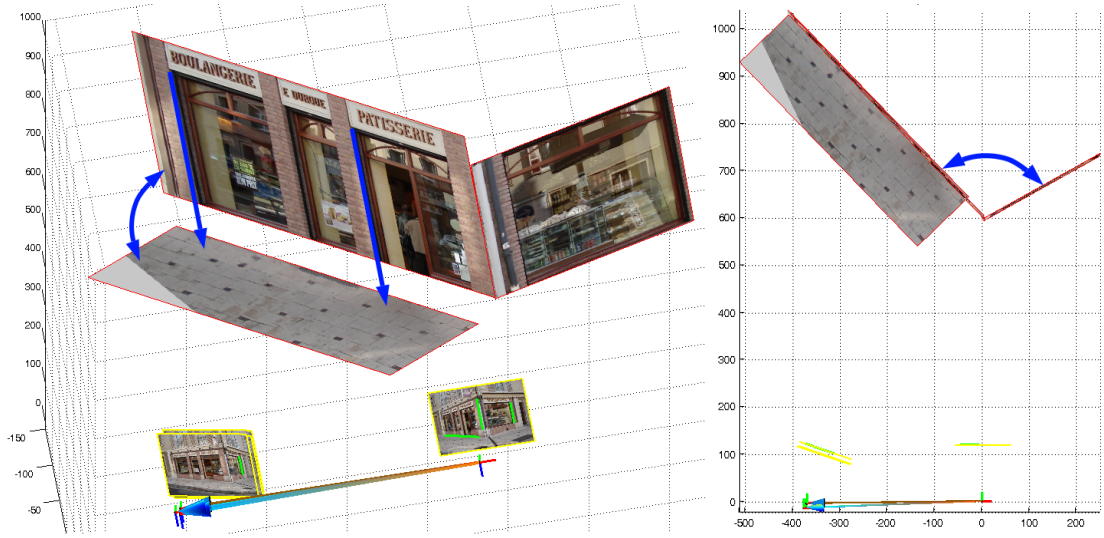


Figure 8: Reconstruction of the bakery and 3 estimated translation vectors in one uniform framework.

6 Summary

We proposed an efficient algorithm especially suitable for piece-wise planar scenes, proving that architecture modeling can be made very simple by exploiting the available information from such senses such as vanishing points and the planar nature of the surfaces. It consists of the following steps:

- Extracting all line segments from left and right image.
- Recovering the rotation through matching vanishing points. This can be done automatically and there is no need for matching individual lines.
- Reconstructing one line and the planar surface attached to it and at the same time recovering the translation (here user needs to feed one line correspondence belonging to a planar surface to the algorithm).
- In previous step, one line and the surface attached to it are already reconstructed. Interestingly enough, now this step can be repeated to incrementally reconstruct more lines and surfaces of the scene and at the same time verifying and improving the accuracy of estimated translation.

We do not establish many feature correspondences (only one line correspondence), nor do we estimate the optical flow or normal flow (in fact such methods do not work for wide baseline motion) but we rely on image intensities of the flat surface. Our method does not have multiple solution ambiguity and it guaranties one solution as long as the surfaces are well textured.

References

[1] L. Wang, U. Neumann, S. You, "Wide-baseline image matching using Line Signatures", *ICCV*, 1311-1318, 2009.

[2] Z.H. Wang, F.C. Wu, Z.Y. Hu, "MSLD: A robust descriptor for line matching", *PR*, 941-953, 2009.

[3] H. Bay, V. Ferrari, L.J. Van Gool, "Wide-Baseline Stereo Matching with Line Segments", *CVPR*, 329-336, 2005.

[4] T. Goedeme, T. Tuytelaars, L.J. Van Gool, "Fast wide baseline matching for visual navigation", *CVPR*, 24-29, 2004.

[5] D. Tell, S. Carlsson, "Wide Baseline Point Matching using Affine Invariants Computed from Intensity Profiles", *ECCV*, 814-828, 2000.

[6] C. Schmid, A. Zisserman, "Automatic Line Matching Across Views", *CVPR*, 666-671, 1997.

[7] C. Baillard, A. Zisserman, "Automatic reconstruction of piecewise planar models from multiple views", *CVPR*, 559-565, 1999.

[8] A.E. Bartoli, P.F. Sturm, "Structure-from-motion using lines: Representation, triangulation, and bundle adjustment", *CVIU* , 100(3): 416-441, 2005.

[9] J. C. Bazin, C. Demonceaux, P. Vasseur, I. S. Kweon, "Motion estimation by decoupling rotation and translation in catadioptric vision", *CVIU*, 114:254-273, 2010.

[10] J.C. Bazin, I.S. Kweon, C. Demonceaux, P. Vasseur, "Spherical region-based matching of vanishing points in catadioptric images", *OMNIVIS*, xx-yy, 2008.

[11] Olivier Faugeras, *Three-dimensional computer vision : a geometric viewpoint*, MIT Press, Cambridge, Mass., 2001.

[12] D. Gallup, J.M. Frahm, P. Mordohai, Q.X. Yang, M. Pollefeys, "Real-time plane-sweeping stereo with multiple sweeping directions", *CVPR*, 1-8, 2007.

[13] D.P. Huttenlocher, G.A. Klanderman, W.J. Rucklidge, "Comparing images using the hausdorff distance under translation" , *CVPR*, 654-656, 1992.

[14] S. Mosaddegh, D. Fofi, P. Vasseur, "Simultaneous ego-motion estimation and reconstruction of piecewise planar scenes from two views", *Technical Report 10-01*, Le2i, UMR CNRS 5158, Universite de Bourgogne, 2010.

[15] C.J. Taylor, D.J. Kriegman, "Structure and motion from line segments in multiple images", *T-PAMI*, 17:1021-1032, 1995.

[16] J. Weng, TS Huang, N. Ahuja, "Motion and structure from line correspondences; closed-formsolution, uniqueness, and optimization", *T-PAMI*, 14(3):318-336, 1992.

[17] Z.Y. Zhang, "Estimating motion and structure from correspondences of line segments between two perspective images", *ICCV*, 257-262, 1995.

[18] F. Fraundorfer, P. Tanskanen, M. Pollefeys, "A minimal case solution to the calibrated relative pose problem for the case of two known orientation angles", *ECCV*, 269-282, 2010.

[19] O. Naroditsky, X.S. Zhou, J. Gallier, S. I. Roumeliotis, K. Daniilidis, "Two Efficient Solutions for Visual Odometry Using Directional Correspondence", *T-PAMI*, 818-824, 2012.