Wide-Baseline Stereo Matching with Line Segments*

Herbert Bay[1], Vittorio Ferrari[1], Luc Van Gool[1,2]
[1]Computer Vision Laboratory (BIWI), ETH Zuerich, Switzerland
[2]ESAT-PSI, University of Leuven, Belgium
{bay,ferrari,vangool}@vision.ee.ethz.ch

Abstract

We present a new method for matching line segments between two uncalibrated wide-baseline images. Most current techniques for wide-baseline matching are based on viewpoint invariant regions. Those methods work well with highly textured scenes, but fail with poorly textured ones. We show that such scenes can be successfully matched using line segments. Moreover, since line segments and regions provide complementary information, their combined matching allows to deal with a broader range of scenes. We generate an initial set of line segment correspondences, and then iteratively increase their number by adding matches consistent with the topological structure of the current ones. Finally, a coplanar grouping stage allows to estimate the fundamental matrix even from line segments only.

1. Introduction

In the last few years, various approaches for wide-baseline stereo matching have been proposed [1, 4, 9, 10, 17]. Most of them use local, viewpoint invariant regions. These regions are extracted independently from each image, characterized by a viewpoint invariant descriptor, and finally matched. Those methods are robust to large viewpoint and scale changes. However, the number of produced matches depends on the amount of visible distinctive texture contained in both views. Images of poorly textured scenes provide only a few matches, and sometimes none at all.

The class of scenes addressed in this paper are typically man-made environments with homogeneous surfaces, like blank walls in architectural interiors. Despite lacking texture, these scenes often contain line segments which can be used as additional features. Line segments convey an important amount of geometrical and topological information about the constitution of the scene, whereas regions capture the details of its local appearance. Using both types of features allows to exploit their complementary information, making wide-baseline stereo matching more reliable and generally applicable.

We propose a novel matcher for straight line segments, based on their appearance and their topological layout. It is capable of reliably finding correspondences among line segments between two views taken from substantially different viewpoints. It works in a completely uncalibrated setting, and hence does not assume epipolar geometry to be known beforehand. Moreover, we show the benefits of matching line segments *and* regions in a combined fashion. Finally, we propose a technique to obtain epipolar geometry from a mixture of line segments and region matches, and even from line segment correspondences alone.

Line segment matching is a challenging task. One of the reasons is that we cannot directly exploit the epipolar geometry as a global geometric constraint. Moreover, line segments have little distinctive appearance, and the location of the endpoints is often inaccurate.

Until now, only a few methods for automatic line segment matching for wide-baseline stereo exist. Schmid and Zisserman [14] perform guided line matching using a plane sweep algorithm. However, their approach requires the prior knowledge of the epipolar geometry. Lourakis et al. [6] use the '2 lines + 2 points' projective invariant for images of planar surfaces, and therefore their method is limited to such scenes. Several other approaches restricted to small baseline cases have been proposed (e.g. [13]).

Our approach exploits the appearance similarity of pairs of line segments, and the topological relations between all line segments. If region matches are available, they are automatically integrated in the topological configuration, and exploited in combination.

The scheme is composed of six stages:

1. A **viewpoint-invariant region matcher** is applied in order to get a set of initial region matches. We use the one proposed by [17]. This first stage is optional: our scheme can work also without region matches.

2. The **line segments are extracted** using the Canny edge detector [2]. Edges are split at points with high

*The authors acknowledge support by the ETH project Blue-C-II.

curvature. Line segments are then fitted to the split edges using orthogonal regression.

3. **Color histogram based matching** is performed in order to get initial candidate matches. The line segments in both views are compared based on the histograms of the neighboring color profiles (section 2).
4. A **topological filter** is applied to remove wrong candidate matches. The filter can work with line segment matches alone, or with a mixture of line segment and region matches (section 3).
5. **More matches** are found by iteratively reintroducing unmatched line segments which respect the topological structure of the current set of matches (section 4). This novel mechanism can substantially increase the number of correct matches.
6. The **epipolar geometry** is estimated after an intermediate step of coplanar grouping using homographies (section 5). The intersections of every set of potentially coplanar line segments are treated as point matches, and used to estimate the fundamental matrix. If available, region matches contribute to the estimation too.

2. Appearance-based matching

This section describes our strategy for obtaining an initial set of candidate matches based on the appearance similarity of the line segments (or 'segments', for short).

Because of the unknown motion range and epipolar geometry, segments cannot be compared using the classic correlation windows or corrected correlation patches [14]. However, if a segment lies on the intersection of two planes or at the border between two differently colored areas, then the color neighborhood on both sides of the segment undergoes only slight changes, even if the camera motion is important.

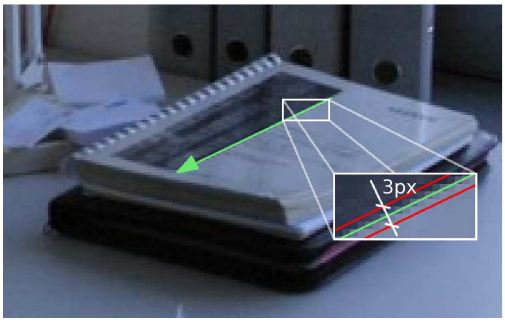

Figure 1. *The lines are directed so that the darker neighborhood is on the right side. The color profiles are separated by 3 pixels.*

Let Ψ^R and Ψ^L be the color profiles along the right and along the left side of a directed segment. The direction of the segment is determined by the mean intensity along the profiles: the brighter profile is set to be on the left (figure 1). This simple criteria allows to determine the direction consistently over both views for the vast majority of cases.

The color profiles are computed along two stripes parallel to the segment, one on either side. The separation between those stripes has to be as small as possible, but not smaller than the standard deviation of the Gaussian filter used for the Canny edge detection (we have to avoid that they contain the edge segment itself). We chose the distance of 3 pixels (see figure 1) in all our experiments. The color intensity values for the profiles are computed using fast B-Spline interpolation as proposed by Unser et al. [18].

A direct comparison of the profiles is not possible because of the inaccurate localization of the endpoints along the segment, and the projective distortion of the profiles. Therefore, we use two color histograms of Ψ^R and Ψ^L as descriptor of a line segment, as they are more robust to these factors (although not invariant either). We define a color histogram $\mathbf{h}$ of a profile $\Psi^{R|L}$ as the vector $[h_1, \ldots, h_M]$ in which each bin h_m contains the number of pixels of $\Psi^{R|L}$ having a certain color m, normalized by the length N of the segment.

$$h_m = \frac{1}{N} \sum_{i=1}^{N} \begin{array}{ll} 1 & \text{if } \Psi^{R|L}(i) = m \\ 0 & \text{otherwise} \end{array} \quad (1)$$

Color histograms considering all possible colors are very big, which would make the computational time for the matching stage tremendously high. Furthermore, they are sensitive to illumination changes as well as to noise. Thus, we reduce the number of colors, by applying a color space quantization using a predefined palette of colors. Let C be a color space and $P = \{c_1, c_2, \ldots, c_i, \ldots, c_n \,|\, c_i \in C, n \ll \| C \|\}$ the quantization space. The quantizer Q is the function that maps every color in C to an element, also called bin, in P.

$$Q : C \rightarrow P$$

The choice of the quantization space is crucial for the quality of the candidate matches. We chose the approach proposed by Smith and Chang [16] who partitioned the HSV color space into 166 bins, placing more importance on the Hue channel (H) than on Saturation (S) and Value (V). The Hue channel is the most useful, and is invariant to changes in illumination intensity. However, in order to distinguish between grayish looking colors, which are frequent in man-made environments, the Saturation and Value channel should also be used to some extent. Therefore, we divide the channels into 18 bins for Hue, 3 for Saturation, and 3 for Value. Additionally, four grey levels including black and white are defined. This choice of the quantization space brings some degree of invariance to brightness changes, without losing the important information of grey levels.

The segments are matched by measuring the similarity of their profile histograms, independently for each side. There are various histogram similarity measures with different properties (e.g. Minkowski, Kulback-Leibler, Earth Mover's Distance). When using color similarity measures within the histogram dissimilarity computation, a quadratic metric is suitable. In [12], such a metric is proposed; it considers Euclidean distances between colors and is much faster to compute than the Earth Mover's distance. This is an important issue, because we have to calculate hundreds of histogram similarities for every segment.

The distance between histograms $\mathbf{h_1}$ and $\mathbf{h_2}$ is given by

$$d_{1,2} = (\mathbf{h_1} - \mathbf{h_2})^\top \mathbf{A} (\mathbf{h_1} - \mathbf{h_2}) \tag{2}$$

where $\mathbf{A} = [a_{i,j}]$ is a 166×166 matrix, and its elements $a_{i,j}$ denote the Euclidean distance between the bins c_i and c_j of the palette P in the color space C. In our case C is the conical representation of the HSV color space. Therefore, the distance $a_{i,j}$ between two colors $p_i = [h_i,\, s_i,\, v_i]$ and $p_j = [h_j,\, s_j,\, v_j]$, in the quantized HSV color space is

$$a_{i,j} = \frac{1}{\sqrt{2}} \left((v_i s_i \cos(h_i) - v_j s_j \cos(h_j))^2 + (v_i s_i \sin(h_i) - v_j s_j \sin(h_j))^2 + (v_i - v_j)^2 \right)^{1/2} \tag{3}$$

The dissimilarity of two segments is expressed by the square root of the mean of the histogram dissimilarities for both sides.

For the matching, the segments in the first image I_1 are compared to all segments in the second image I_2. A segment of I_2 is considered as a candidate match for a segment of I_1 if their dissimilarity is below 0.25. Notice that a segment can have more than one candidate match, therefore we call them *soft matches*. When there are many of them, only the 3 with the lowest dissimilarity are kept.

The reasons for choosing soft matches instead of the classic 1-to-1 approach are manifold. First of all, segments tend to be only weakly distinctive in their appearance, thus several non-corresponding segments tend to resemble each other. Secondly, important changes in the viewing conditions may result in a loss of similarity of the correct match (scale changes, specularities, background changes, etc.). Finally, real scenes often contain repeated or visually similar elements. Keeping more than just the most similar segment significantly reduces the chances of missing the correct match.

3. Topological filter

This section describes a mismatch filter based on the semi-local spatial arrangement of the features (segments and regions) in two views. This is an extension of the topological filter proposed by Ferrari et al. [3], who used similar techniques only with region triplets. Instead, our filter can handle sets of matches containing only segments, or both segments and regions at the same time. This filtering stage substantially reduces the number of mismatches, which are produced by the initial matching stage as a consequence of the weak appearance distinctiveness of segments.

3.1. Sidedness Constraint

The filter is based on two forms of the *sidedness constraint*, each used in one of two successive stages (figure 2).

The first form states that for a triplet of feature matches, the center of a feature $\mathbf{m_v^1}$ should lie on the same side of the directed line $\mathbf{l_v}$ going from the center $\mathbf{m_v^2}$ of the second feature to the center $\mathbf{m_v^3}$ of the third feature, in both views $v \in \{1, 2\}$:

$$\mathrm{side}(\mathbf{l_1},\, \mathbf{m_1^1}) = \mathrm{side}(\mathbf{l_2},\, \mathbf{m_2^1}). \tag{4}$$

with

$$\mathrm{side}(\mathbf{l_v},\, \mathbf{m_v^1}) = \mathrm{sign}\left(\mathbf{l_v} \mathbf{m_v^1}\right) = \mathrm{sign}\left(\left(\mathbf{m_v^2} \times \mathbf{m_v^3}\right) \mathbf{m_v^1}\right) \tag{5}$$

This constraint holds always if the three features are coplanar, and in the vast majority of cases if they are not [3]. The filter is designed to favor features which respect the constraint often, while tolerating modest amounts of violations.

In the first stage of the filter, the constraint is tested for all feature triplets (see next subsection). Regions and line segments are used here in a homogeneous manner, as they both contribute with their centers. The center of a segment is defined as its midpoint.

The second stage of the filter uses another form of the sidedness constraint. For a *pair* of feature matches, the first being a line segment, the center of the second feature must be on the same side of the (directed) line segment in both views. Notice that we previously assigned a direction to each line segment (section 2), so it directly takes up the role of $\mathbf{l_v}$ as in the first form of the constraint. This second stage takes advantage of the fact that a line segment suffice to define a directed line, so the constraint can be verified already for pairs, rather than triplets. Moreover, this second stage uses the whole information provided by the segment.

3.2. Algorithm

A feature triplet or pair including a mismatch is more likely to violate the sidedness constraint. When this happens, it does not tell yet *which* feature is a mismatch. Therefore, for every feature we compute a violation score which counts the number of violated constraints for all unordered triplets/pairs including it. A mismatched feature is typically involved in more violations than a correctly matched one, thus a match with a high violation score has a higher chance to be incorrect.

The filter algorithm is similar for both stages, which are applied one after the other. Given a set of N feature matches $\mathbf{m_v^i}$ between views $v \in \{1, 2\}$, with S the number of segment matches, do:

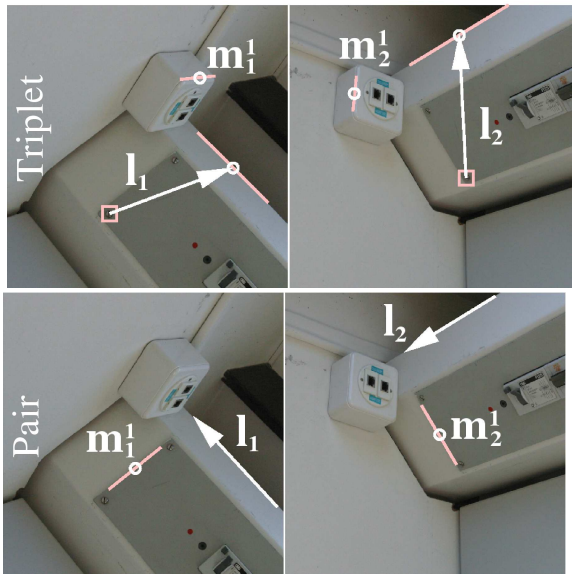

Figure 2. *Top: a triplet of features (two segments, with midpoints 'o', and a region center, marked with a square).* $\mathbf{m}^1$ *lies on the same side of the directed line* $\mathbf{l}$ *in both views. Bottom: a pair of segments, where one takes the role of the directed line* $\mathbf{l}$*.*

1. For every feature match $(\mathbf{m_1^i}, \mathbf{m_2^i})$, test all sidedness constraints in which it is involved.

2. For each feature, we define a *violation score* V_i by counting the number of violated constraints, divided by the total number in which it is involved. This total number depends on the current stage of the filter. For the triplet stage this is $(N-1)(N-2)/2$. For the pair stage, there are three cases: $N-1$ when the feature is a line-segment and plays the role of the directed line $\mathbf{l_v}$, or S when the feature is a region, or $S-1$ when it is a line-segment playing the second role in the pair. This latter distinction in necessary because of the two different roles a line-segment plays in the second stage of the filter: it can provide the directed line, or the center-point whose sidedness is tested.

3. Find the feature match with the highest violation score V_{max}. If $V_{max} > 0.15$, this match is considered incorrect and removed, then the algorithm re-iterates from step 1. If $V_{max} \leq 0.15$, or if all matches have been removed, the algorithm terminates.

After applying the topological filter, there might still be some soft matches left. For further processing, we keep for each feature only the single match with the lowest sum of appearance dissimilarity and topological violation score.

3.3. Motivation

It is important to point out the reasons behind our choice for this filtering stage. First of all, we cannot compute the fundamental matrix directly from segment correspondences, and therefore cannot exploit the epipolar constraint as a filter (as opposed to what is done in the region matching literature, e.g. [9, 10, 17]). The topological filter instead can effectively use the information within the segment matches to discriminate between correct and incorrect ones.

The second main reason lies in the fact that the proposed filter is insensitive to the exact localization of the features, because the number of violated sidedness constraints varies slowly and smoothly for a feature departing from its ideal location. This is particularly important in our context, where the midpoint of a segment is often inaccurately localized, because the endpoints are not precisely determined along the segment's direction. Moreover, since the perspective projection of the midpoint of a segment in 3D space is not congruent with the midpoint of the projected segment, the midpoints of two long matched segments can sometimes represent only an approximative point correspondence.

4. Finding more matches

While Ferrari et al. [3] use the sidedness constraint only to filter mismatches, we exploit it also for adding matches.

During the topological filtering, some correct matches are erroneously rejected. Additionally, there might be correct correspondences which have been missed by the initial matcher, because of their high similarity with other segments. Thanks to the novel technique presented in this section, such matches can be found and added to the current set. We apply this to find more segment matches, but it can easily be extended to regions.

The algorithm is iterative and starts from the set of matches after applying the topological filter. At each iteration, new line segment matches are added if they respect the topological structure of the current set (i.e. if they have a low violation score). In the next iteration, the updated matches set serves as reference to calculate the violation scores of potential new matches.

1. For every unmatched segment in I_1, calculate the violation score of each possible candidate match in I_2, computed with respect to the current configuration of matches. As in section 2, we consider as possible candidates all segments with an appearance dissimilarity below 0.25. It is advantageous to consider also segments of I_2 which are already matched, because another segment may have a lower violation score. The three candidate matches with the lowest violation score are stored in a waiting room.

2. All segments in the waiting room are added to the current set of matches

3. The two stages of the topological filter are applied on this extended set of matches. The matches which are

rejected are left out of any further consideration. In the case that some segments still have more than one candidate match, these are eliminated by keeping only the one with the lowest sum of appearance dissimilarity and topological violation score.

4. If the current configuration is the same as in the beginning of this iteration, the algorithm terminates, otherwise it iterates to point 1.

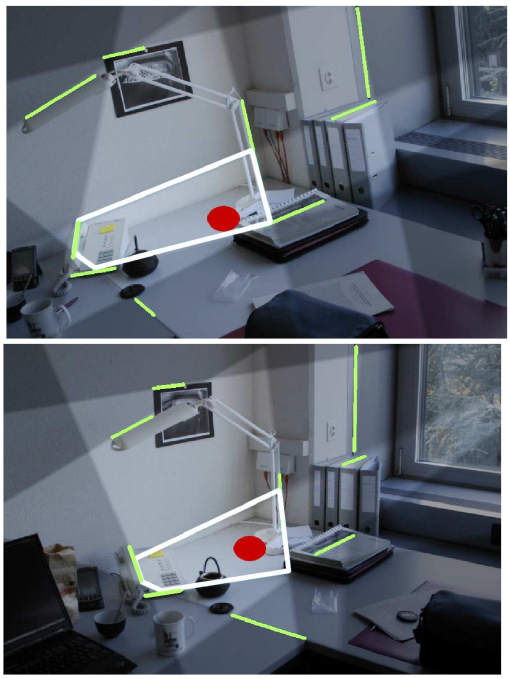

Figure 3. *Candidate matches within the highlighted zone of the bottom image violate no sidedness constraints for the point in the top image.*

This method typically substantially increases the number of correct matches.

5. Epipolar geometry estimation

This section presents a new scheme for estimating the epipolar geometry from segment matches. If region matches are available, their information is also included.

As we cannot directly estimate the epipolar geometry from line correspondences, we go through an intermediate coplanar grouping stage. We assume that the scene contains at least two planes. The idea of our method is to group coplanar segments using homographies. We then compute the intersection points of all pairs of segments within a group, and treat them as point correspondences. This provides many matches which are likely to be correct, and do not depend on the accuracy of the estimated homography. Moreover, these intersection points are exact correspondences, in contrast to the midpoints used in the previous sections. We finally estimate the fundamental matrix from these point matches using standard RANSAC [5].

The advantage of our method is that we use homographies only for grouping potentially coplanar segments, and *not* for directly estimating the epipolar geometry as described in [15]. Estimating the epipolar geometry directly from homographies has been shown to be less stable than the estimation from point correspondences [8]. Moreover, our strategy uses the whole information provided by all line segments within a group, rather than just what is expressed by the homography matrix. This brings robustness: even when some group accidentally contains a few non-coplanar, or mismatched segments, only a minority of incorrect point correspondences are generated, and the subsequent RANSAC stage can then proceed successfully.

5.1. Homography estimation

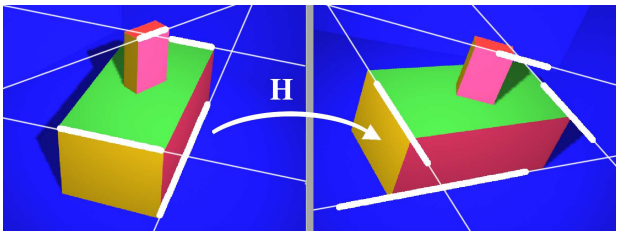

Figure 4. *Left: four non-coplanar segment matches (thick). Right: their projection via the homography H estimated from the correspondence among their support lines (thin). The segments are not in correspondence with their projection, but they 'slide' along the support lines instead.*

Groups of coplanar segments and the associated homographies are computed by the following algorithm:

1. We repeat points 2 and 3 for K subsets Ω of four randomly selected segments. The number of subsets is calculated adaptively as described in [5]

2. A homography $\mathbf{H}$ is fit to the support lines of the segments of Ω, using SVD. In order to check whether the segments of Ω are really coplanar, we project the associated profiles $\Psi_1^{R|L}$ from the first image to the second by $\mathbf{H}$, and compare them by normalized cross-correlation (NCC). More precisely, we require that the NCC between each profile and its projection in the other image is above 0.5. The test is applied symmetrically, by projecting the profiles from the second image to the first by $\mathbf{H^{-1}}$. This test proved very effective in rejecting non-coplanar quadruplets of segments (figure 4). If no Ω fulfilling this test is found within K trials, the algorithm stops.

3. Now that a valid quadruplet is found, we try to include more segments in the group. The symmetric transfer distance d_{transfer} is computed for every segment

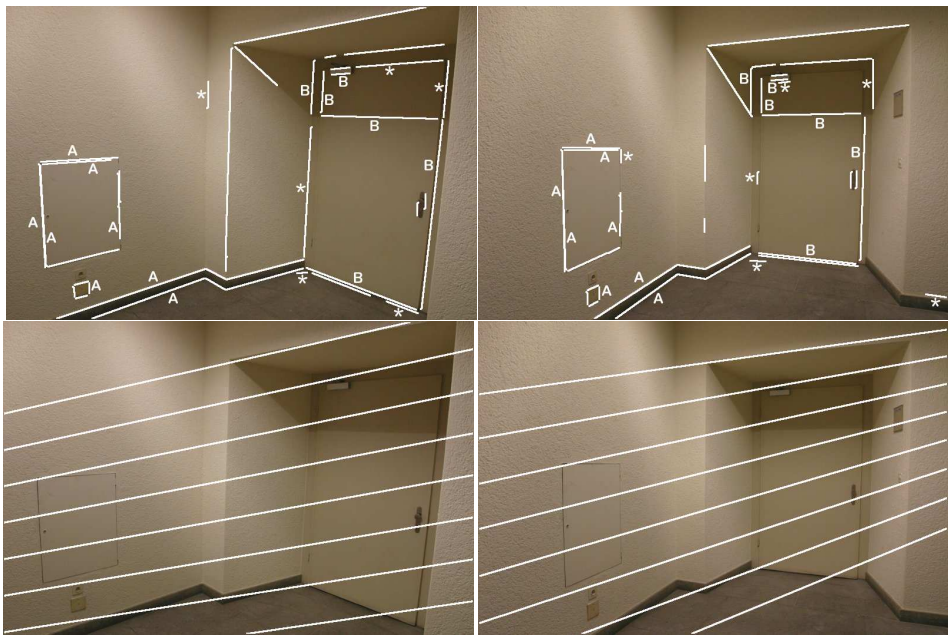

Figure 5. *Corridor scene. Top: 41 segment matches, including 6 mismatches (marked '*'). The two coplanar groups are marked 'A' and 'B'. Bottom: estimated epipolar geometry.*

match $(\mathbf{m_1}, \mathbf{m_2})$

$$d^2_{\text{transfer}} = \max\ d_\perp(\mathbf{m_1^s}, \mathbf{H^{-1}l_2})^2,\ d_\perp(\mathbf{m_1^e}, \mathbf{H^{-1}l_2})^2\ + \max\ d_\perp(\mathbf{m_2^s}, \mathbf{Hl_1})^2,\ d_\perp(\mathbf{m_2^e}, \mathbf{Hl_1})^2\ , \quad (6)$$

where $\mathbf{l_v}$ is the support line of the segment in view v, and $\mathbf{m_v^s}$, $\mathbf{m_v^e}$ are the endpoints. $d_\perp$ is the distance between a point and a line.

All matches with $d_{\text{transfer}} < 5$ pixels, which also pass the NCC test above, are put in a coplanar group.

4. After K trials, we select the group with the largest number of segments. In case of ties, we prefer the group with the lowest sum of symmetric transfer distances. The segments of the selected group are removed from the current set of matches, and the whole algorithm is repeated in order to find the next group of coplanar segments.

The algorithm is repeated until the stopping condition in step 2 is met. In practice, the procedure returns only a few groups (rarely more than 4), because non-coplanar quadruplets are likely to be rejected during step 2. Figure 5 shows an example grouping.

5.2. Fundamental Matrix estimation

After producing groups of coplanar segments, we now estimate the fundamental matrix. For each group, we compute the intersection points of all pairs of segments within it, therefore producing point correspondences. All correspondences arising from all groups are put in the same *estimation set*. If available, the center of region correspondences are also added to this set. Finally, the Fundamental Matrix is estimated from all the point matches in the estimation set using the standard RANSAC algorithm [5]. Figure 5 shows an example where the fundamental matrix has been successfully computed based on line segments only.

6. Results and Conclusion

We report results on 3 example scenes, imaged under wide-baseline conditions. All experiments have been made with the same parameters. The average computation time for the matcher was 8 seconds on a modest workstation (Pentium 4 at 1.6GHz, line detection not included).

The first scene depicts the corner of a corridor, and has almost no texture (figure 5 top). Trying to match regions with the approach of [17] fails, as only 6 matches are found, out of which just 3 are correct. Our approach instead, using line segments only, finds 21 matches (16 correct) already before the stage of adding more matches (section 4). After the latter was performed, the number increases to 41 (35 correct), showing how this stage can substantially boost the number of correct matches, while keeping the number of mismatches low. Based on these 41 segment matches, the algorithm we proposed in section 5 successfully estimates the epipolar geometry (figure 5 bottom).

Figure 6 presents another textureless scene, consisting of a corner of an office room's ceiling. Beside the viewpoint change, an additional challenge is posed by the strong in-plane camera rotation. Our method, with only line segments, produces 24 correspondences (21 correct) spread over the whole images, and determines the fundamental ma-

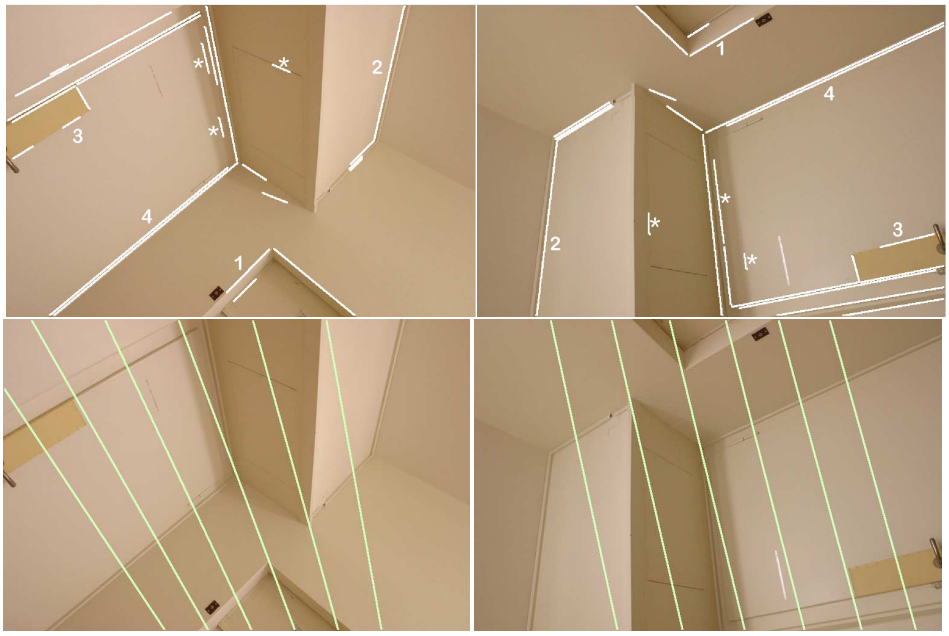

Figure 6. *Corner scene: Top: 21 correct matched segments, and 3 mismatched ones (marked '*'). The numbers indicate a few correct matches, to help the reader. Bottom: estimated epipolar geometry.*

trix (figure 6 bottom).

Figure 7 shows the last scene. The viewpoint change can be appreciated by looking at the deformation of the blackboard. The region matcher yields 26 matches (11 correct) on the left wall with the blackboard, but only 3 matches (2 correct) on the cupboard doors on the right. When estimated only from these region matches, the fundamental matrix is only valid for the left wall (figure 7 top). By running our method, using segments *and* regions, we obtain 50 segment matches (37 correct, spread over both planes), and 18 region matches (12 correct), thereby clearly reducing the number of mismatched regions. Moreover, the method finds the correct fundamental matrix (figure 7 bottom).

For comparison, we matched all image pairs also with three state-of-the-art invariant region detectors [7, 9, 10] and described the regions with the SIFT descriptor [7]. The latter performed best in a recent comparative study [11]. The matching is carried out as follows: a region of the first image is matched to the region in the second image with the closest descriptor, if it is closer than 0.6 times the distance to the second closest descriptor[1]. This is a common robust matching strategy [1, 7, 11]. As table 1 shows, MSER and Harris-Affine find nearly no matches, while the DoG detector provides a few correct matches. Nevertheless, their number is still unsatisfactory, in contrast to the number of line segment matches produced by our method. This shows that

[1]We have used the official detector implementations `http://www.robots.ox.ac.uk/~vgg/research/affine/`. The matching was performed using David Lowe's software from `http://www.cs.ubc.ca/~lowe/keypoints/`, Version 3.

Scene	MSER	DoG	Harris-Affine
Corridor	2/2	8/14	0/2
Corner	1/1	5/10	1/1
Classroom (cupboard)	0/0	0/0	1/2
Classroom (rest)	14/14	6/8	3/3

Table 1. *Results for different region detectors. MSER: Maximally Stable Extremal Regions [9], DoG: Difference of Gaussians [7] and Harris-Affine [10]. Entries report the number of correct matches over the number of correct matches. Notice how there is a relatively high number of correct matches only on the textured blackboard.*

our technique is much better suited for textureless scenes.

In conclusion, we have presented a new wide-baseline stereo matcher capable of working with line segments alone, or combined with affine invariant regions. It can robustly estimate the fundamental matrix from a mixture of both kind of correspondences, or even from segment matches only. The experiments support the approach and show that it can find a good number of line segment correspondences also on untextured scenes, imaged under general wide-baseline conditions, and without any prior knowledge about the scene or camera positions (e.g.: in contrast to [14]). Moreover, the system can effectively exploit the complementary information provided by segments and regions, and hence can handle a wider range of scenes than possible with either cue alone.

One potential improvement could be to apply the stage of adding more matches also for regions, or to extend it to take advantage of the epipolar constraint once the fundamental

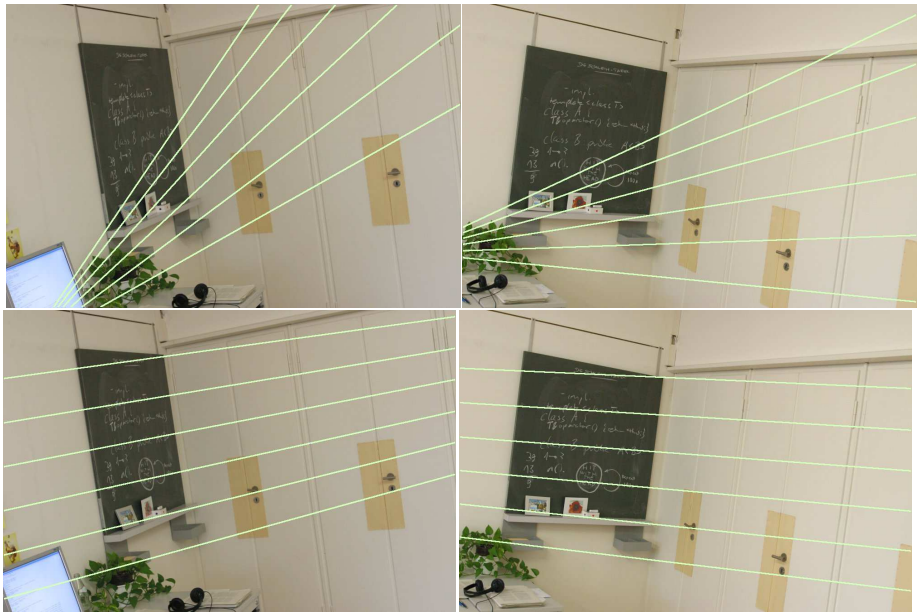

Figure 7. *Classroom scene. Top: the epipolar geometry estimated from region matches is incorrect. Bottom: correct epipolar geometry, estimated including line segments.*

matrix has been computed. Moreover, the color quantizer could be chosen based on an initial global analysis of the color distribution in the two images. This would increase the quality of the initial set of candidate matches.

References

[1] A. Baumberg. Reliable feature matching across widely separated views. In *CVPR*, pages 774–781, 2000.

[2] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, November 1986.

[3] V. Ferrari, T. Tuytelaars, and L. V. Gool. Wide-baseline muliple-view correspondences. In *CVPR*, pages 718–725, 2003.

[4] T. Goedemé, T. Tuytelaars, and L. V. Gool. Fast wide baseline matching with constrained camera position. In *CVPR*, pages 24–29, 2004.

[5] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[6] M. Lourakis, S. Halkidis, and S. Orphanoudakis. Matching disparate views of planar surfaces using projective invariants. In *British Machine Vision Conference*, pages 94–104, 1998.

[7] D. G. Lowe. Distinctive image features from scale-invariant keypoints, cascade filtering approach. *International Journal of Computer Vision*, 60(2):91–110, January 2004.

[8] Q. T. Luong and O. D. Faugeras. Determining the fundamental matrix with planes: Instability and new algorithms. In *CVPR*, pages 489–494, 1993.

[9] J. Matas, O. Chum, U. Martin, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *British Machine Vision Conference*, pages 384–393, 2002.

[10] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *European Conference on Computer Vision*, pages 128–142, 2002.

[11] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In *CVPR*, volume 2, pages 257–263, June 2003.

[12] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin. The QBIC project: querying images by content using color, texture and shape. In *SPIE International Symposium on Electronic Imaging: Science and Technology, Conf. 1908, Storage and Retrieval for Image and Video Databases*, 1993.

[13] O. A. Pellejero, C. Sagüés, and J. J. Guerrero. Automatic computation of the fundamental matrix from matched lines. In *10th Conference of the Spanish Association for Artificial Intelligence CAEPIA*, pages 197–206, 2003.

[14] C. Schmid and A. Zisserman. Automatic line matching across views. In *CVPR*, pages 666–671, 1997.

[15] D. Sinclair, A. Blake, S. Smith, and C. Rothwell. Planar region detection and motion recovery. *Image and Vision Computing*, 11(4):229–234, 1993.

[16] J. Smith and S. Chang. Single color extraction and image query. In *Proceedings of IEEE International Conference on Image Processing*, pages 528–531, 1995.

[17] T. Tuytelaars and L. Van Gool. Wide baseline stereo based on local, affinely invariant regions. In *British Machine Vision Conference*, pages 412–422, 2000.

[18] M. Unser, A. Aldroubi, and M. Eden. Fast B-spline transforms for continuous image representation and interpolation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(3):277–285, March 1991.