18^{th} *Computer Vision Winter Workshop*
Walter G. Kropatsch, Fuensanta Torres, Geetha Ramachandran (eds.)
Hernstein, Austria, February 4-6, 2013

Line-based 3D Reconstruction of Wiry Objects

Manuel Hofer, Andreas Wendel, Horst Bischof
Institute for Computer Graphics and Vision
Graz University of Technology, Austria
{hofer,wendel,bischof}@icg.tugraz.at

Abstract. *Man-made environments contain many weakly textured surfaces which are typically poorly modeled in sparse point reconstructions. Most notable, wiry structures such as fences, scaffolds, or power pylons are not contained at all. This paper presents a novel approach for generating line-based 3D models from image sequences. Initially, camera positions are obtained using conventional Structure-from-Motion techniques. In order to avoid explicit matching of 2D line segments in the various views we exploit the epipolar constraints and generate a series of 3D line hypotheses, which are then verified and clustered to obtain the final result. We show that this approach can be used to densify various sparse occupied point clouds of urban scenes in order to obtain a meaningful model of the underlying structure.*

1. Introduction

Generating 3D models from a set of images has become a widely studied field of research over the last few years. The majority of available algorithms is based on point correspondences between multiple views using various local descriptors such as the Scale-Invariant Feature Transform (SIFT) [13] in order to obtain a 3D point cloud while simultaneously estimating the camera parameters. This process is called Structure-from-Motion (SfM). The density of the resulting point cloud highly depends on the amount of texture available in the images. Therefore, point-based SfM may fail in man-made environments with a low amount of distinctive interest points (e.g. urban scenes, indoor scenes). To tackle this issue, many line-based approaches have been presented over the years, due to the fact that especially man-made objects (e.g. buildings) can usually be represented by a set of 3D line segments. Similar to traditional SfM it is usually necessary to match 2D line segments from various views to triangulate a 3D line segment. This can be done using appearance-based similarity measures, e.g. normalized-cross-correlation (NCC) or line descriptors [12, 21], which can be combined with additional geometric constraints [3]. Since the endpoints of matched line segments usually do not correspond to each other due to inexact line segment detection or occlusions, creating 3D line segments from matched 2D lines is much more difficult than traditional point-to-point matching.

Figure 1. Two examples for wiry structures. The left image shows a power pylon and the right image a scaffold in front of a house.

Most of the previous approaches rely on an accurate line matching process between the various views using some appearance-based similarity measures. This usually works fine if the lines are located on a planar surface with constant background, for instance when matching window frames. However, when dealing with wiry structures such as power pylons, bridges or scaffolds (see Figure 1 for some examples), appearance-based matching is hardly possible due to changing surroundings of the line segments in different views (see Figure 2). We present an approach which is especially designed to handle such cases but also performs well on solid objects.

Figure 2. An example were no appearance based line matching can be performed. Note that corresponding line segments have different surroundings in both views (yellow lines).

2. Related Work

In the following we present selected papers from the field of line-based 3D reconstruction. We start with an overview of appearance-based methods which cannot directly be applied to our problem but share some ideas with our approach.

Baillard et al. [1] presented a method which makes use of the epipolar constraint by estimating line correspondences along the epipolar beam. To find the correct match they evaluate the NCC score for candidate lines using patches around the line segments. The estimated 3D line segment is the intersection of the half-planes through the lines of sight of the two endpoints in both views. They further verify their hypotheses by minimizing the reprojection error using the trifocal tensor [6].

Bay et al. [2] use optional region matches in addition to line matches based on color histograms in order to establish an initial set of candidates. They apply a topological filter in order to remove wrong candidates and increase the candidate set by adding unmatched line segments which fit to the topological structure of the already matched hypotheses. They further estimate the epipolar geometry using coplanar subsets of their candidate set. Very accurate results are reported, even for sparsely textured scenes.

In order to generate 3D line models for urban scenes, Schindler et al. [16] proposed an approach which takes vanishing point information into account. They assume that relevant edges are located along mutually orthogonal vanishing directions which reduces the degrees of freedom for 3D line estimation. Their approach delivers pleasant results for urban structures but unfortunately is limited to pictures taken at near-ground level due to their assumptions.

Another approach presented by Kim et al. [8] is based on the intersection context of coplanar line pairs. They match line intersection context features across multiple views using NCC as similarity measure and reject false intersections using coplanarity constraints on the corresponding line segments. The proposed method works well for a wide range of scenarios even when only little texture is available.

Unfortunately, all of these appearance-based approaches usually do not perform well for wiry structures, since they technically do not match the line itself, but rather its surroundings. In our case, explicit matching may be impossible, since the ever changing background is not coplanar with the line and often very far away from the object to be reconstructed. In order to create 3D models without the need of explicit line matching, Jain et al. [10] developed a sweeping based approach which defines the unknown 3D locations of the endpoints of 2D line segments as random variables. They estimate 3D line hypotheses by generating all possible endpoint locations in a certain depth interval (assuming known camera intrisics and extrinsics) and keep the one with the highest score based on the gradient images of many neighboring views. Hence, they create a 3D line for every 2D line in every view. In order to delete outliers and cluster corresponding line segments together, they group 3D line segments which lie close in space and discard all segments which do not have at least one such neighbor. They also perform an optimization based on 2D line connections using loopy belief propagation to enforce connected 3D lines. Even though their approach delivers very accurate results and is very robust against noise and partial occlusions, it is very slow compared to previous approaches.

In our approach we build upon the principles presented in [10] but use a different matching strategy. Instead of using a time consuming sweeping approach we generate hypothetical 3D line segments using epipolar constraints, which drastically limits the number of possible 3D locations for each 2D line segment. We will show that this leads to a significant performance increase while still creating accurate results.

3. Sparse Structure-from-Motion

Given an unordered set $I = \{I^1, ..., I^n\}$ of n images and the corresponding cameras $C = \{C^1, ..., C^n\}$ we want to generate a set of 3D line segments $S = \{S^1, ..., S^k\}$. Since we do not perform explicit line matching and line-based relative

pose estimation the cameras have to be known beforehand. For this purpose we use a point-based SfM system. This limits the application to scenes where interest points can be found, but we have seen that we can usually find enough correct correspondences for an accurate relative pose estimation in the background of wiry structures.

We follow the approach of Wendel et al. [20] and Irschara et al. [9] which enables us to perform sparse SfM for unordered image sets. The three necessary processing steps are feature extraction, feature matching, and geometry estimation. In the first step we extract SIFT [13] features from all images. SIFT has been shown to work well in general scenes [14], but it also works surprisingly well in scenarios with wiry structures. The reason is that matches are obtained either in the background, or in the foreground in case of a homogeneous background such as sky. Afterwards, we match the resulting keypoint descriptors between all possible image pairs and perform a geometric verification procedure using the Five-Point algorithm [15]. In order to eliminate possible outliers we use RANSAC [5] for robust estimation. The resulting pairwise reconstructions are then merged to obtain a sparse reconstruction of the scene. Finally, bundle adjustment [17] is applied to minimize the global reprojection error over all measurements. See [19] and [7] for further details.

As a result we know the relative positions of all cameras C in a common coordinate frame, and we can thus proceed to the task of 3D line segment estimation.

4. Reconstruction of 3D Line Segments

Our algorithm consists of three steps: 2D line segment detection extracts line segments from each input image, 3D line segment hypotheses generation tries to estimate the 3D position of these segments, and finally 3D line grouping and outlier removal merges corresponding segments from different views and removes incorrect estimates. In the following sections these steps will be explained in detail.

4.1. 2D Line Segment Detection

In order to generate triangulated 3D line segments from a set of images, we first have to apply a line segment detection algorithm onto our input images. We employ the Line Segment Detector (LSD) [18] algorithm to extract all relevant line segments with as few incorrect detections as possible. The authors report their algorithm to be significantly faster than previous methods while producing very accurate results. Their approach is based on the grouping of points with a high gradient and similar level line angle, followed by a least squares line fit. All detections are validated using the Helmholtz principle [4] which proves to be very effective for the general case. Figure 3 shows the detected line segments for a power pylon image.

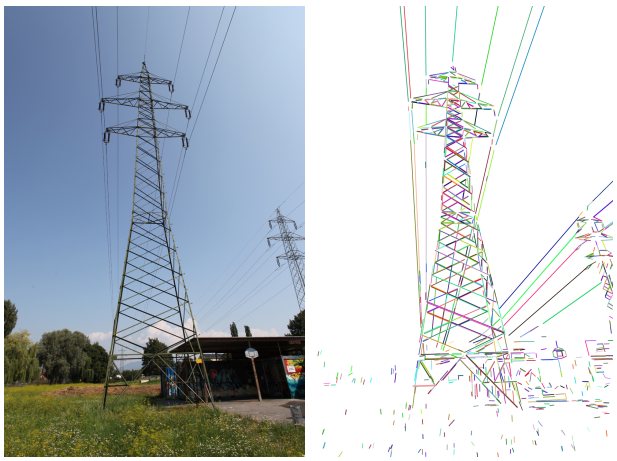

Figure 3. Line Segment Detection. The line segments extracted using the *LSD* [18] algorithm are visualized in pseudo-colors. The underlying wiry structure is represented very well, except for a few outliers due to noisy gradients, cause for instance by grass.

4.2. 3D Line Segment Hypotheses Generation

Assuming no false detections in the previous step, every 2D line segment from image I^i corresponds to a 3D line segment in world space. Since we can not perform an explicit appearance-based matching procedure and triangulation, we have to estimate the correct 3D location of each segment in a different way.

As we know the projection matrix P^i of the camera, we are able to compute the *epipolar geometry* between I^i and some other view I^j. Using the epipolar lines e_p and e_q defined by the two endpoints p and q of a certain line segment l in view i, we can limit the possible matches for l to those line segments whose endpoints lie on e_p and e_q respectively. In practice it is unlikely that we will find an exact match with both endpoints being located exactly on the epipolar lines, e.g. due to imprecise line detection or occlusions. Therefore we extend all candidate segments which overlap with the region between the two epipolar lines to infinity (from line segments to actual lines) and intersect them with e_p and e_q in order to generate hypothetical matches. This enables us to

find correct matches even if the current line segment is shorter or longer in I^j (see Figure 4). For every hypothesis we create a 3D line segment L by triangulating the two corresponding endpoint pairs from the two views I^i and I^j.

Since we usually have more than one hypothesis for each 2D line segment (because the epipolar lines do not provide enough information to perform exact matching), we have to determine which one is correct. Therefore we adopt a gradient based scoring approach similar to [11, 10]. We then backproject each 3D line segment L into all neighboring views $N(I^i)$ of I^i with a camera center closer than a certain distance d_c and an absolute viewing angle difference smaller than d_{ang} to the current camera C^i. For each camera we compute a set of measurement points M along and perpendicular to the backprojected line, and compute the image gradient-based score

$$s(L) = \frac{1}{|N(I^i)|} \sum_{I \in N(I^i)} \sum_{x \in M(I)} \frac{\|\nabla I(x)\|}{|M(I)|} e^{-(\frac{\lambda \cdot dist(x,L)}{2 \cdot dist_{max}(L)})^2} \quad (1)$$

for every 3D line segment L, where $\nabla I(x)$ denotes the image gradient at position x, $dist(x, L)$ is the perpendicular Euclidean distance to the backprojected line in the current image I and $dist_{max}(L)$ denotes the maximum distance based on the configuration of the measurement points. Assuming that line segments correspond to high gradient areas in images, this method ensures that we choose the hypothesis which fits best to the image data. Using this formula we give more weight to measurement points which are closer to the backprojected line, and less weight to those perpendicular to it depending on the distance. An illustration is given in Figure 5.

After computing the score for each hypothesis we choose the one with maximum score, denoted as L_{best}, which is then added to our 3D line segment hypotheses set H. Since we generate 3D line segments for all views individually, we end up with a quite large hypotheses set which has to be pruned. Figure 6 shows an example for a 3D line model before grouping and outlier removal.

4.3. 3D Line Grouping and Outlier Removal

It is possible that the correct matches for 2D line segments are not among the candidates, for instance because the line segments are not redetected in any neighboring view, and therefore we have to remove possible outliers. The outlier removal process goes hand in hand with the line grouping step which has to be performed in order to remove multiple detections. Since we match and triangulate the 2D line segments individually for every view, the same 3D line might be generated in multiple views. Assuming a correct matching procedure, all the hypotheses in H which correspond to the same 3D line should be located close in space. Hence, a line clustering algorithm is performed in order to generate the final 3D line model.

In order to remove incorrectly triangulated 3D line segments and cluster corresponding segments, we adopt the idea of spatial proximity based grouping from [10]. First, we order the hypotheses set H by

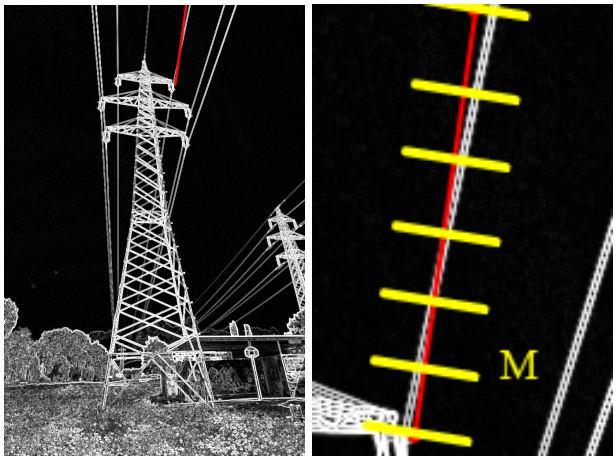

Figure 5. The left image shows the gradient magnitudes from a power pylon image with a backprojected 3D line hypothesis shown in red. The right image shows a close-up of the line segment with the set of measurement points M illustrated as yellow lines. The weighted sum of the gradient magnitudes over all measurement points is computed and then divided by the number of points in order to compute the score for this view. The average score over all neighboring views is then used to evaluate the best hypothesis (see Equation 1).

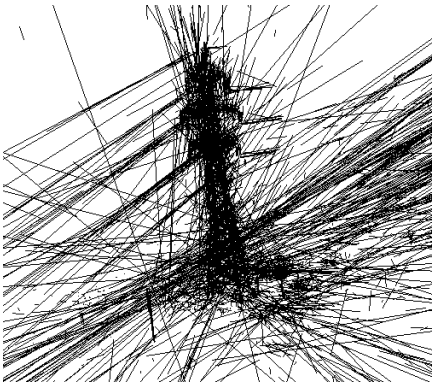

Figure 6. 3D line segment hypotheses before the grouping and outlier removal procedure. Our approach generates 71538 segments from 106 views. Note that there is a large number of outliers due to incorrect matches, but the power plyon which appears in the imagery is clearly recognizable.

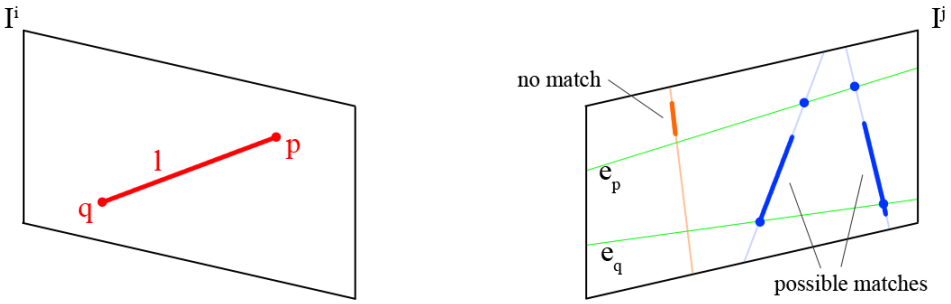

Figure 4. We match the line segment L in view I^i with line segments from view I^j using its epipolar lines e_p and e_q. The blue line segments are possible candidate matches because of their overlap with the region between the two epipolar lines. The endpoints of the hypothetical line segments used for triangulation are shown as blue dots. The orange line segment does not overlap with the epipolar lines and is therefore not considered to be a possible match.

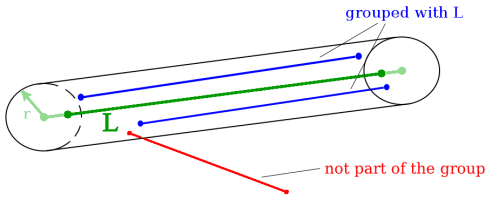

Figure 7. To group corresponding 3D line segments together, the true segment L (green) is expanded by 10% in each direction. All other line segments with both endpoints within a cylinder of radius r, defined by the new endpoints, are considered to be in the same group as L (blue lines). The red line does not belong to the group, because one of its endpoints is outside the cylinder.

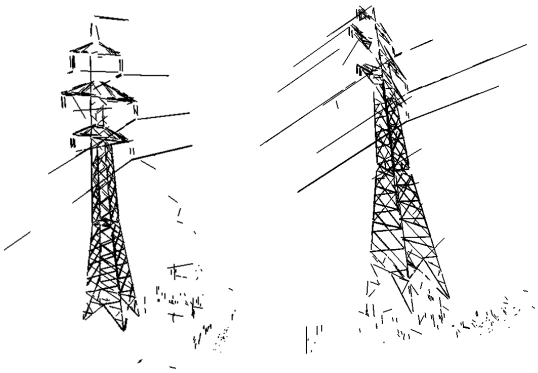

Figure 8. After the grouping procedure most of the outliers have been successfully removed resulting in an accurate 3D model (1381 line segments).

score in descending order to start grouping with lines which are best aligned with the image gradients. For each line $L_m \in H$ we define a cylinder of a fixed radius r by expanding the central axis of the line segment by 10% in both directions. We then try to find all line segments $L_n, n \neq m$ where both endpoints are located within the cylinder (Figure 7).

If the final line group (including L_m) has at least h_{min} members we consider it to be valid and exclude all line segments in the group from further grouping, otherwise L_m is removed from H and we continue with the next best hypothesis.

After the clustering step, each group is replaced with one single line segment for our final 3D line segment set S. To define this line we first compute the center of gravity of all line segment endpoints from the group. Afterwards we perform a singular value decomposition of the scatter matrix containing all endpoints and take the eigenvector corresponding to the maximum eigenvalue as new line direction. We now project all endpoints onto the new line and add the line segment defined by the two outmost points to S. Figure 8 illustrates the outcome of the grouping procedure. Note that compared to Figure 6 most of the outliers have been successfully removed.

5. Experiments

In the previous section we have already shown a resulting 3D model of a power pylon. In this section we want to present additional results and finally compare our algorithm to [10] using one of their test-cases.

5.1. Parameter Selection

The various steps of our approach require a set of parameters in order to generate pleasant results. Most of them are valid for a large number of scenarios and therefore do not need to be especially tuned.

The line segment detection algorithm (LSD [18]) does not need parameters. In order to eliminate outliers and speed up the computation we reject 2D line segments smaller than 1% of the diagonal length of the image in pixels, which is usually sufficient to capture the underlying structure of our images.

During hypotheses generation we need to deter-

mine which views are considered neighboring views $N(I^i)$ for the current view I^i (see Section 4.2). The maximum viewing angle difference $d_{ang} = 50°$ and the maximum distance between the camera centers $d_c = 30$ for all our experiments. In order for the second parameter to make any sense we need to know the scale of our 3D model. In our experiments the result achieved during preprocessing (camera estimation) is transformed to a metric scale ($1 \equiv 1m$), using either a marker with known size [11] or manual user interaction. Assuming equidistant camera centers we usually have a large number of views available for scoring. In order to increase the performance (since scoring has to be done for every 2D line segment in every view) we limit the number of neighboring views to 20.

For the scoring procedure we choose the set of measurement points (M) in a way that the distance between the points on the backprojected line is 5 pixels. The number of perpendicular points is set to 5 in each direction (with a distance of 1 pixel), meaning that $dist_{max} = 5$. The parameter λ is set to 10 (see Equation 1).

The parameters for the grouping procedure are the only ones which have to be estimated for each testcase individually. For most scenarios setting $r = 0.05$ and $h_{min} = 3$ yields good results, meaning that the grouping radius is $5cm$ and every 3D line segment has to be correctly estimated in at least 3 different views.

5.2. Results

In traditional point-based SfM it is often the case that the resulting point cloud is sparsely distributed due to the lack of distinctive features especially for man-made structures. Many of the keypoints may be rather located on the background instead of the object. Nevertheless, background features can be used for relative pose estimation and therefore our line matching algorithm can be applied in order to densify the 3D model.

Figure 9 shows an example 3D point cloud of a house surrounded by a scaffold, and Figure 10 shows a model of a staircase. As we can see, the point clouds are rather sparsely occupied and the visible objects are difficult to determine for the viewer. Adding 3D line segments clearly improves the result and allows the viewer to identify the underlying structure.

Our algorithm is designed to handle wiry structures. Nevertheless, it is not limited to such scenarios and can also handle solid objects. In order to compare our approach to [10], we reconstructed their *Timber-frame house* sequence[1] using our algorithm. The sequence consists of 240 synthetic images. Figure 11 shows exemplar views from the sequence along with our 3D reconstruction and the result from Jain et al. [10], colored using the Hausdorff distance as similarity measure (for densely sampled points along the lines) to the ground truth model. Table 1 shows the root mean square (RMS) error for both reconstructions compared to the CAD model.

As we can see, both algorithms manage to recon-

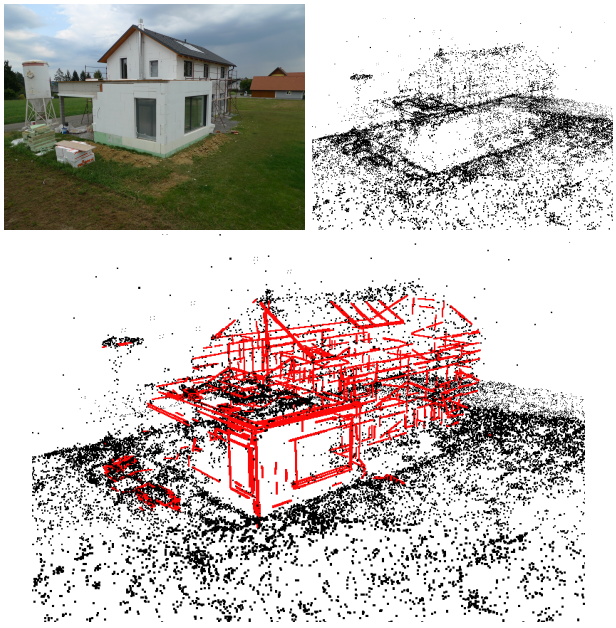

Figure 9. The top images show an example view from a house sequence (93 images) and a SfM point cloud. The bottom image shows the densified 3D model with the reconstructed line segments.

Figure 10. The left image shows an example view from a staircase sequence (14 images). The right image shows the densified 3D model with the reconstructed line segments.

[1] http://www.mpi-inf.mpg.de/resources/LineReconstruction/

Method	min error	max error	RMS error
Jain et al.	0.000	0.019	0.0036
Ours	0.000	0.023	0.0013

Table 1. The comparison to the method by Jain et al. [10] revealed that our method performs better in terms of the RMS error but their method has a slightly lower maximal error.

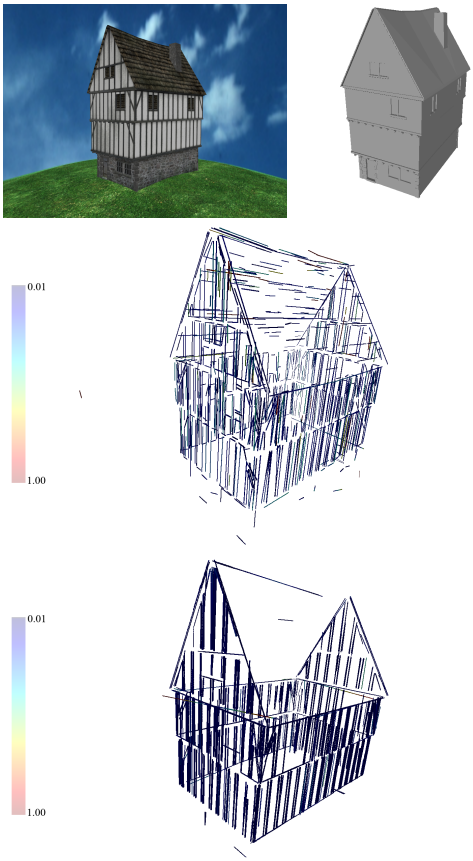

Figure 11. The top images show an example view from the synthetic Timber-frame house sequence (240 images) along with the ground truth CAD model. The middle image is the reconstruction achieved by [10], the bottom image is our reconstruction. The color reveals the errors compared to the CAD model (from 0.01 to 1.00). Best viewed digitally and in color.

struct the building in a qualitatively accurate way. Our approach performs better in terms of the RMS error, while Jain et al. are able to reconstruct a few more lines, especially on the roof. Even though the resulting models are similar, the computational time differs highly. The authors report that their algorithm often needs several hours to deliver the result, while our method is able to reconstruct this sequence in 7.5 min using all 240 images and not only a subset of 72 as [10].

5.3. Performance Evaluation

Since we have to evaluate many possible matches for each 2D line segment (to avoid appearance-based matching) our algorithm is more time consuming than traditional line-matching approaches. Nevertheless, we manage to generate accurate results for the general case in reasonable time. Table 2 shows a performance evaluation for the three test sequences presented in this paper. All experiments were performed on a desktop PC equipped with an Intel Core2 4×2.66 GHz processor. Note that for our own sequences (Pylon, House and Staircase) the image size was significantly larger than for the Timber-frame house, which explains the difference in speed.

6. Conclusion

We have presented a novel approach for the purpose of generating 3D line models without explicit appearance-based matching. The proposed algorithm performs well for wiry structures as well as solid objects. In contrast to a previous approach by Jain et al. [10], we exploit epipolar constraints to speed-up the computation while still creating accurate results. We have shown that for scenes with few keypoints located on the actual foreground object, 3D line segments can be used to densify the resulting model. This is of particular importance for urban scenes and man-made structures which often provide few distinctive feature points.

While our approach is able to generate 3D line segments even when a 2D segment is not exactly redetected in any other view (due to the matching strategy based on epipolar lines), it usually generates a large set of outliers. These outliers have to be removed in a computationally expensive grouping step, which may take a lot of time depending on the number of hypotheses. Therefore, our future work will be to formulate the matching procedure in a probabilistic way to allow online hypotheses generation, in order to further improve the performance.

Acknowledgements

This work has been supported by the Austrian Research Promotion Agency (FFG) project FIT-IT Pegasus (825841/10397) and OMICRON electronics GmbH.

References

[1] C. Baillard, C. Schmid, A. Zisserman, and A. Fitzgibbon. Automatic line matching and 3d re-

Sequence	#img	Resolution	#lines raw	#lines final	LSD	Matching	Grouping	Total
Pylon	106	5616×3744	71538	1381	17.2	15.5	34.1	66.8
House	93	3048×2736	30967	1262	13.8	3.2	6.7	23.7
Staircase	14	3072×2304	16517	265	2.7	0.8	1.1	4.6
Timber-f. H.	240	1280×960	29927	2113	4.2	2.5	0.8	7.5

Table 2. This table shows a performance evaluation for the test sequences used in this paper. All times are in minutes (last four columns).

construction of buildings from multiple views. In *ISPRS Conference on Automatic Extraction of GIS Objects from Digital Imagery*, volume 32, 1999. 2

[2] H. Bay, V. Ferrari, and L. Van Gool. Wide-baseline stereo matching with line segments. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, 2005. 2

[3] F. Bin, W. Fuchao, and H. Zhanyi. Robust line matching through linepoint invariants. *Pattern Recognition*, 45(2), 2012. 1

[4] A. Desolneux, L. Moisan, and J.-M. Morel. Meaningful alignments. *International Journal of Computer Vision*, 40, 2000. 3

[5] M. Fischler and R. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communication Association and Computing Machine*, 24(6), 1981. 3

[6] R. Hartley. A linear method for reconstruction from lines and points. In *Fifth International Conference on Computer Vision*, 1995. 2

[7] C. Hoppe, M. Klopschitz, M. Rumpler, A. Wendel, S. Kluckner, H. Bischof, and G. Reitmayr. Online feedback for structure-from-motion image acquisition. In *In Proceedings of the British Machine Vision Conference (BMVC)*, 2012. 3

[8] K. Hyunwoo and L. Sukhan. A novel line matching method based on intersection context. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2010. 2

[9] A. Irschara, V. Kaufmann, M. Klopschitz, H. Bischof, and F. Leberl. Towards fully automatic photogrammetric reconstruction using digital images taken from uavs. In *Proceedings International Society for Photogrammetry and Remote Sensing Symposium, 100 Years ISPRS - Advancing Remote Sensing Science*, 2010. 3

[10] A. Jain, C. Kurz, T. Thormaehlen, and H. Seidel. Exploiting global connectivity constraints for reconstruction of 3d line segments from images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. 2, 4, 5, 6, 7

[11] T. Kempter, A. Wendel, and H. Bischof. Online model-based multi-scale pose estimation. In *Proceedings of the Computer Vision Winter Workshop (CVWW)*, 2012. 4, 6

[12] B. Khaleghi, M. Baklouti, and F. Karray. SILT: Scale-invariant line transform. In *IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, 2009. 1

[13] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60, 2004. 1, 3

[14] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 27(10), 2005. 3

[15] D. Nister. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6), 2004. 3

[16] G. Schindler, P. Krishnamurthy, and F. Dellaert. Line-based structure from motion for urban environments. In *Third International Symposium on 3D Data Processing, Visualization, and Transmission*, 2006. 2

[17] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment a modern synthesis. *Vision Algorithms: Theory and Practice*, 2000. 3

[18] R. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall. LSD: A fast line segment detector with a false detection control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(4), 2010. 3, 5

[19] A. Wendel, C. Hoppe, H. Bischof, and F. Leberl. Automatic fusion of partial reconstructions. In *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2012. 3

[20] A. Wendel, A. Irschara, and H. Bischof. Automatic alignment of 3d reconstructions using a digital surface model. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), Workshop on Aerial Video Processing*, 2011. 3

[21] W. Zhiheng, W. Fuchao, and H. Zhanyi. MSLD: A robust descriptor for line matching. *Pattern Recognition*, 42(5), 2009. 1