Incremental Line-based 3D Reconstruction using Geometric Constraints

Manuel Hofer
hofer@icg.tugraz.at
Andreas Wendel
wendel@icg.tugraz.at
Horst Bischof
bischof@icg.tugraz.at

Institute for Computer Graphics and Vision
Graz University of Technology
Austria

Abstract

Generating accurate 3D models for man-made environments can be a challenging task due to the presence of texture-less objects or wiry structures. Since traditional point-based 3D reconstruction approaches may fail to integrate these structures into the resulting point cloud, a different feature representation is necessary. We present a novel approach which uses point features for camera estimation and additional line segments for 3D reconstruction. To avoid appearance-based line matching, we use purely geometric constraints for hypothesis generation and verification. Therefore, the proposed method is able to reconstruct both wiry structures as well as solid objects. The algorithm is designed to generate incremental results using online Structure-from-Motion and line-based 3D modelling in parallel. We show that the proposed method outperforms previous descriptor-less line matching approaches in terms of run-time while delivering accurate results.

1 Introduction

Generating accurate 3D models of man-made objects and urban scenery from an image sequence is a challenging task. Traditional Structure-from-Motion (SfM) approaches may fail because of the high amount of untextured objects and wiry structures present. These objects are usually poorly represented in the resulting point clouds, which are obtained by matching local point features across multiple views using local descriptors, such as the Scale-Invariant Feature Transform (SIFT) [10]. The main problems are the low descriptor discriminability of visually similar points, and the changing surroundings for feature points located on wiry structures. Therefore, a different kind of image feature is necessary. Since most man-made objects can be approximated by line segments, line-based 3D reconstruction techniques can be used as an alternative. While common approaches usually deliver accurate results for a wide range of urban scenery, they cannot be directly applied to wiry structures. This is because of explicit 2D line segment matching, which is usually necessary to estimate 3D line segments. To match two line segments from different views, an appearance-based similarity measure has to be applied (e.g. normalized-cross-correlation) or line descriptors such as MSLD [15, 16] or LEHF [4] have to be used. Since the resulting matching scores are based

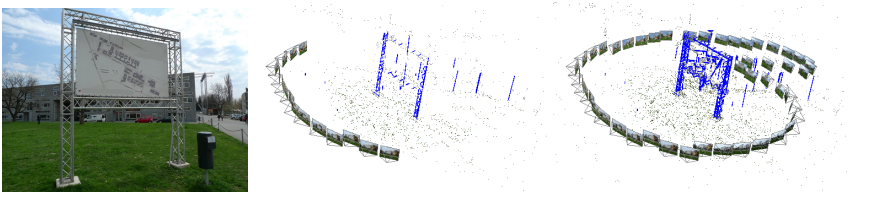


Figure 1: The incremental 3D line modelling procedure is illustrated for the *Sign* sequence. The images show the result after 15, and the full 48 images, together with the sparse 3D point cloud. As we can see, the density of the reconstruction improves significantly when more images are available. The reconstruction including SfM took 6 minutes.

on the surroundings of a line segment rather than the line segment itself, explicit matching would fail for wiry structures due to the changing background, as well as for repeated structures due to the low discriminability between similar segments.

To overcome such limitations, methods which do not rely on appearance-based line segment matching can be applied [5, 8]. Those algorithms, which assume known cameras, are usually based on generating a large set of possible 3D line segment hypotheses, gradient based scoring, and spatial clustering. For this purpose the scale of the reconstruction has to be known, which can be difficult to estimate unless markers or GPS information are available. Moreover, hypotheses generation and grouping are very time consuming steps, which have to be computed offline after all cameras are available and oriented correctly.

We propose a novel line-based 3D modelling approach, which extends the principles presented in [5, 8] by incremental hypotheses clustering and geometric verification steps, without the need of time consuming scoring in the image space. Furthermore, we show how the spatial grouping radius can be derived from a distance threshold in the image space, to achieve scale invariance. We demonstrate how fusing this approach with an incremental point-based SfM [6] leads to an online 3D reconstruction method, which is able to cover wiry- and repeated structures, as well as solid objects. Figure 1 illustrates the incremental reconstruction process. Evaluation showed that this approach outperforms previous approaches in terms of performance, while still delivering accurate 3D models.

2 Related Work

Over the years many line-based 3D reconstruction approaches have been presented. As stated above, most of them are based on matching 2D line segments over multiple images using some appearance-based similarity measure. Additionally, geometric constraints can be used to verify sets of matched line segments [11, 14] along with the information provided by the intersection context [7]. Even though these methods deliver accurate results for various scenarios, they cannot handle wiry objects such as power pylons, street lights, or traffic signs. In order to reconstruct these structures as well, we need to apply an approach without the usage of similarity based line segment matching.

In the approach by Jain *et al.* [8] a line-based 3D model is generated using an image sequence with known cameras, without any kind of explicit line matching. Therefore, the depths of the endpoints of detected line segments in image space are seen as random variables. In order to estimate probabilities for certain depth values, they use backprojection of the resulting 3D line hypotheses, and gradient based scoring in multiple views. By exploiting global connectivity constraints and spatial clustering they estimate the final locations of

the 3D line segments, and remove outliers for which no valid cluster can be formed. Therefore, information about the scale of the reconstruction is necessary to define an appropriate clustering radius. Their evaluations showed accurate results for various scenarios. Unfortunately, the algorithm is slow due to the extensive evaluation of all possible depth values in a certain sweeping range. Hofer *et al*. [5] proposed a modified version where the possible 3D endpoint locations are limited to a certain range. This is achieved by intersecting the epipolar lines of the line segment endpoints in neighboring views with overlapping line segments. Assuming that the same line segment is visible in multiple views, there is a high probability that the correct match is among these hypothetical correspondences. The authors showed that these modifications lead to a massive performance increase while still delivering accurate results. Nevertheless, the reconstruction procedure is still comparably slow and assumes that the cameras as well as the reconstruction scale are known beforehand. Therefore, both methods are not directly adaptable to perform SfM in an online manner.

Existing real-time SfM approaches involving line segments are usually related to Simultaneous Localization and Mapping (SLAM) [1]. SLAM approaches are designed to perform pose estimation and sparse 3D reconstruction simultaneously in previously unknown environments. While traditional SLAM methods are based on interest point features, it has been shown that incorporating line segments produces more accurate results for urban environments [3, 4, 9]. Even though these approaches are in a way related to the proposed method, they focus more on fast localisation than on generating accurate and complete 3D line models of the scene. Additionally, most of these algorithms require explicit line matching and are therefore not out-of-the-box usable for wiry structures.

In Section 3 we will explain our proposed incremental line-based 3D reconstruction method, and how we integrate it into an existing online SfM approach [6], for on-the-fly reconstruction of man-made scenes. In Section 4 we evaluate the accuracy of our approach using a synthetic dataset, and present additional real-world examples. We will demonstrate that our method outperforms [5, 8] in terms of performance, without lowering the accuracy of the results.

3 Incremental Line-based 3D Reconstruction

We want to achieve two things: First, we want to be able to perform on-the-fly 3D reconstruction using line segments. Second, we want to be able to handle different types of urban scenery, including wiry structures and texture-less objects. For the first requirement, we have to use incremental SfM principles, where each new image is integrated into an initial reconstruction based on at least two views, using absolute pose estimation. While pose estimation could be done using line segments alone (e.g. [2]), this would require line segments to be matched before camera information is available, using appearance-features. This contradicts with the second requirement, because we cannot rely on accurate similarity based line matching when wiry structures are involved. To fullfill both requirements, we fuse point-based online SfM [6] for camera estimation, and an incremental descriptor-less line-based 3D reconstruction approach for refinement.

For pose estimation we directly apply the algorithm by Hoppe *et al*. [6]. Their approach is divided into an alignment part, where new cameras are incrementally integrated into an existing reconstruction, and a structure expansion part, where the 3D point cloud is densified using SIFT [10] feature matches. When a new view is correctly aligned with the existing geometry, bundle adjustment [12] is performed in order to maintain global consistency. The algorithm provides incremental camera poses along with 3D world points and the corresponding visibility information.

For every new pose, we perform incremental line-based 3D reconstruction. After line segment detection using the LSD algorithm [13], we integrate the new image into the existing reconstruction. To avoid appearance-based line matching, we compute a potentially large set of possible matches among all previously processed similar views using weak epipolar constraints, for each newly detected line segment. Since we perform online SfM, we often cannot decide right away which potential match is the best, due to the lack of redundancy in a certain part of the scene. Therefore, we keep all possible matches for each line segment and perform hypotheses merging during the matching procedure, by applying purely geometric constraints. This allows us to easily adapt our 3D model when new information is available, or when parameters have been changed during runtime. To decide whether two hypotheses should be merged or not, we use spatial clustering in world- and image space. While [5, 8] assume a certain constant grouping radius in world space, which means that the reconstruction scale has to be known beforehand, we achieve scale invariance by deriving a dynamic spatial grouping radius from the image space. This allows us to reconstruct a wide range of man-made scenery, without the need of using markers or GPS information for scale estimation. To achieve close to real-time performance, we avoid time consuming scoring in the image space by using cluster size and visibility information to compute the inlier set after each new image. Since the number of potential hypotheses could be very large, unpromising hypotheses have to be removed periodically. In the following sections we give a detailed overview over all the relevant computing steps involved.

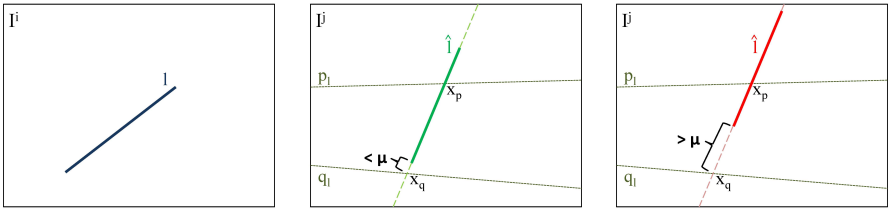
3.1 Initial Hypotheses Generation

To perform incremental SfM we need to have an initial geometry involving at least two views. Therefore, given two images I_1 and I_2 and their respective sets of 2D line segments L_1 and L_2, we create an initial set of 3D line segment hypotheses H by computing all possible line segment matches between the two images.

To limit the number of potential matches, we exploit epipolar constraints using the corresponding cameras C_1 and C_2 (estimated using [6]). For each line segment $l \in L_1$ we compute the epipolar lines p_l and q_l for the two endpoints, in image I_2. Each line segment $\hat{l} \in L_2$ is then extended to a full line and intersected with the epipolar lines to obtain two intersection points x_p and x_q. We consider a line segment $\hat{l}$ to be a potential match for l if it fulfills both of the following constraints: For at least one of the endpoints of the line segment $\hat{l}$, the distance to the nearest intersection point has to be smaller than μ (*endpoint similarity constraint*). The second endpoint of $\hat{l}$ has to be closer to the other intersection point (*orientation constraint*). For an illustration of the matching procedure see Figure 2.

The parameter $\mu = 0.1 \cdot min(\|x_p - x_q\|, \|\hat{l}\|)$ represents a 10% tolerance threshold with respect to the length of $\hat{l}$ and the distance between the intersection points. The *endpoint similarity constraint* states that even under occlusions, corresponding line segments need at least one similar endpoint with respect to the epipolar geometry. Allowing a certain tolerance is necessary in order to compensate imperfect pose estimation or variance in the line segment detection step. The *orientation constraint* states that if there is one corresponding pair of end- and intersection point, the line segment has to overlap the region between the epipolar lines as much as possible. This constraint is weaker than the first one, but ensures that we have a limited set of potential matches.

For each putative match $\{l, \hat{l}\}$ we compute a 3D line segment $K_{l,\hat{l}}$ by triangulating the corresponding 2D line segments from I_1 and I_2. Each match results in a new hypothesis $h \in H$, which consists of an estimated 3D line segment K_h, the camera set $C(h) = \{C_1, C_2\}$,



(a) (b) (c)

Figure 2: (a) One example line segment l in image I^i. (b) The green line segment is a potential match for l in image I^j, according to the matching constraints. (c) The line segment shown in red is not a potential match for l, because the first constraint is violated.

and a corresponding 2D line set $L(h) = \{l, \hat{l}\}$, and a set of triangulated 3D line segments $K(h) = \{K_{l,\hat{l}}\}$. For a new hypothesis, $K_h = K_{l,\hat{l}}$. Additionally, each hypothesis has a score $s(h)$ and a corresponding camera $C^*(h)$ defined as

$$s(h) = 1 - min\left\{\left|\left\langle \frac{\overrightarrow{K_h}}{\|\overrightarrow{K_h}\|}, \frac{\overrightarrow{C_i}}{\|\overrightarrow{C_i}\|} \right\rangle\right|\right\}, \quad C^*(h) = \underset{C_i}{\operatorname{argmax}}(s(h)), \quad C_i \in C(h) \tag{1}$$

where $\overrightarrow{K_h}$ is the directional vector of K_h, and $\overrightarrow{C_i}$ denotes the camera ray of camera C_i, and $\langle \cdot, \cdot \rangle$ is the inner product. This means that the score is high for hypotheses with a corresponding camera where the camera ray and the 3D line vector have an angle close to 90°. This prevents that hypotheses which are estimated in a way that the 3D line segment points in the same direction as the camera ray are defined as valid. Such hypotheses appear very small in the corresponding images and are therefore prone to become degenerate by false matches.

3.2 Dynamic Grouping Radius Estimation

So far, we have shown how new hypotheses can be generated from 2D line segment matches between two views. To perform incremental hypothesis merging for further incoming images, we need to define a spatial grouping radius r_{space}. This parameter defines how close in space two hypotheses need to be for merging. Unlike [5, 8] we do not want to define a constant radius beforehand, because we usually do not know the scale of our reconstruction. To be scale invariant we derive r_{space} from the image space dynamically. Therefore, we define a maximum uncertainty σ in the image space, which denotes a tolerance threshold for possibly imprecise line segment detection and camera estimation. This means that two similar line segments should be merged if the distance between them is smaller than σ.

To bring this value to 3D space, we first compute a specific grouping radius $r_{space}(h)$ for each hypothesis $h \in H$. Therefore, we project the 3D line segment K_h back into the two images which created this hypothesis. We then shift the resulting 2D line segments k_1 and k_2 in the same orthogonal direction by σ, and obtain $\hat{k_1}$ and $\hat{k_2}$. Afterwards, we compute the epipolar lines for the endpoints of $\hat{k_1}$ and intersect them with the line defined by $\hat{k_2}$. We triangulate the corresponding end- and intersection points, and obtain a shifted 3D line segment $\hat{K_h}$. The radius $r_{space}(h)$ is defined as the maximum distance between the endpoints of $\hat{K_h}$, and the infinite line passing through K_h.

To be robust against imprecise triangulation, we compute a characteristic grouping radius $r_{space}(C_i)$ for each view. For this purpose we use the median of all values $r_{space}(h)$, for which

the hypothesis h references the view i. For the initial image set, both views have the same grouping radius, because all existing hypothesis have to reference both of them. During the incremental reconstruction procedure, each hypothesis h stores its radius, but uses the value $r_{space}(C^*(h))$ associated with its characteristic camera, for the merging step. This allows us to adapt the system to severe viewpoint changes, when the target object might appear much larger or much smaller than before. We will show that this procedure leads to accurate results, without the need of parameter tuning.

3.3 Incremental Update

When a new image I_i is available we integrate it into our existing reconstruction, based on the current set of previously computed images $S = \{I_1, ..., I_{i-1}\}$. Since the number of images which are already integrated in our geometry could be very large, we have to determine which of them are visually similar to the new image. Hence we have to define a set $N(I_i)$ of neighboring views for I_i, which consists of all views I_j where the angle between the camera rays $\angle(\overrightarrow{C_i}, \overrightarrow{C_j}) < \alpha$, and there is at least one 3D world point (obtained by the sparse SfM) which is visible in both images. The angle threshold α is set to 45° for all our experiments. We found that even if larger angles would result in a better triangulation quality, we often cannot find appropriate matches using a larger angle threshold, due to line-segment dissimilarities under severe viewpoint changes. Since $|N(I_i)|$ could be very large, we usually only use the M nearest neighbors, based on the angle difference. This limitation does not alter the results significantly, while the runtime per image gets approximately constant. The default value for M is 10 (see Section 4 for an evaluation). Once we have determined $N(I_i)$, we update our hypotheses set H using the new information available.

As above, we compute all possible matches for each line segment $l \in L_i$, with the images in $N(I_i)$. Unlike before, we do not want to create a new hypothesis for each match, because some of these matches might correspond to an already existing hypothesis. Therefore, for each possible correspondence $\{l, \hat{l}\}$ we try to add l to an existing hypothesis, which references $\hat{l}$. We create a triangulated 3D line segment $K_{l,\hat{l}}$ and compute the spatial distances to all candidate hypotheses. If we find a hypothesis h for which the spatial distance is below $r_{space}(C^*(h))$, we perform an additional verification step before l is added to h. We project the triangulated segment $K_{l,\hat{l}}$ into all referenced views in $C(h)$, and compute the distance to the corresponding line segments. If it is below σ for all segments, we consider l to be a part of h.

To update the hypothesis h, we add C_i to $C(h)$, l to $L(h)$, and $K_{l,\hat{l}}$ to $K(h)$. We also re-compute the score $s(h)$ and update the corresponding view $C^*(h)$. The estimated 3D line segment K_h has to be adapted as well, incorporating the newly triangulated line segment. Therefore, we estimate the singular value decomposition of the scatter matrix containing all endpoints of all line segments in $K(h)$. The new line direction d is then defined as the eigenvector corresponding to the maximum eigenvalue. We compute the center of gravity G of all endpoints, and define a 3D line using G and the direction d. Finally, we project all supporting 3D line segments onto the line defined by G and d, and set K_h to be the part of the newly computed line which is overlapped by more than half of the projected segments. This ensures that the characteristic segment will be estimated correctly, even if some degenerated line segments have been matched to this hypothesis.

For each possible match $\{l, \hat{l}\}$, for which we cannot find an existing hypothesis to be added to, we create a new hypothesis in the same way as during initialization (see Section 3.1). After all possible matches for every line segment in L_i have been evaluated, we have to

re-estimate the characteristic grouping radius for each view, if new hypotheses have emerged during the matching procedure (see Section 3.2). This can be done very efficiently, since we do not recompute the characteristic grouping radius for existing hypotheses.

3.4 Hypothesis Verification and Outlier Removal

After all line segments have been matched, we compute the current inlier set. Therefore, we sort the hypotheses $h \in H$ descending by the number of supporting line segments ($= |L(h)|$). If two hypotheses have the same number of line segments, we order them according to their reprojection error. To compute the current inlier set, we define three possible states for a hypothesis: *active* (inlier), *neutral* (new hypothesis), or *inactive* (outlier).

We compute the current inlier set by iterating over the sorted hypotheses set. If a hypothesis h is *active* it is already an inlier from previous evaluation. If it is *neutral*, we have to make the decision whether it is an inlier or an outlier. For h to be an inlier the following criteria have to be fulfilled: the number of supporting line segments has to be at least λ, and the score $s(h)$ has to be higher than 0.5. If this holds, the hypothesis is set to *active* and all other hypotheses related to any of the segments referenced by h, are set to *inactive*. This ensures that each 2D line segment can only contribute to one 3D line segment in the inlier set. If the validity criteria are not satisfied, the hypothesis is set to *inactive* and considered an outlier. All *inactive* hypotheses are skipped during the iteration. After processing, all *active* hypotheses form the current result and can be visualized. The default value for λ is 4 (see Section 4 for an evaluation).

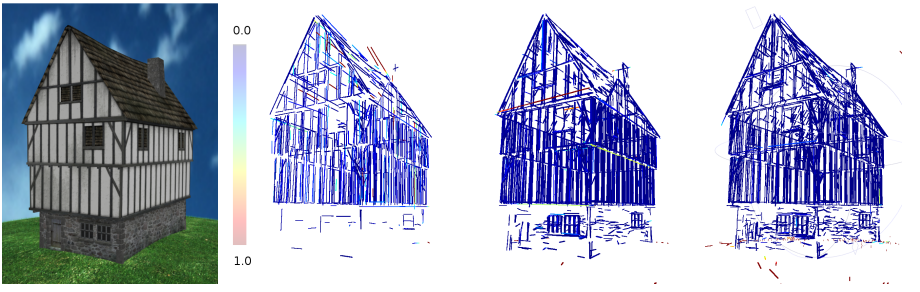
Compared to [5, 8] we do not need to compute a gradient based score in the image space to validate our hypotheses, which is a very time consuming task. Our method is purely based on geometric constraints and validation through clustering. Furthermore, the incremental grouping procedure prevents evaluation of a very large set of hypotheses at the end of the algorithm. However, using our proposed method might as well produce a huge hypotheses set for large image sequences, which would ultimately lead to a performance breakdown. Hence, we need to remove unpromising hypotheses from time to time. To achieve this, we evaluate the number of supporting line segments in a hypothesis h compared to the number of views, which have been matched with score maximizing view $C^*(h)$. If there are less than λ line segments that agree on h, and $C^*(h)$ has been matched with at least $2 \cdot \lambda$ views, then hypothesis h is permanently removed from the hypothesis set. This ensures that the number of hypotheses remains approximately constant, with respect to the actual number of line segments in the scene. To ensure that *inactive* hypotheses have the possibility to become valid again, we reset all *inactive* hypotheses to *neutral* when a new image is being processed.

One thing that has to be additionally considered is bundle adjustment [17]. In order to maintain a global consistency in the reconstruction, this is particularly important for loop closing when an object is surrounded by the camera. Since bundle adjustment might severely change the camera parameters during the reconstruction process, we have to adapt our estimated hypotheses accordingly. For this purpose we re-estimate and re-validate all existing hypotheses after the periodically performed bundle procedure.

4 Evaluation

To evaluate the accuracy of the proposed method, we use the *TimberFrame*[1] sequence from [8]. For this synthetic sequence a ground truth CAD model is available. Even though a natural scene would be a much better demonstration of the capabilities of the algorithm, it is

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



(a) (b) (c) (d)

Figure 3: (a) An example image from the *TimberFrame* sequence by [8] (240 images total). (b) Reconstruction by [8] (runtime of several hours mentioned). (c) Reconstruction by [5] (45 minutes without SfM). (d) Our reconstruction (12 minutes including SfM, $M = 10$, $\lambda = 4$). The results are colored to illustrate the error compared to the ground truth CAD model (values cropped from 0.0 to 1.0).

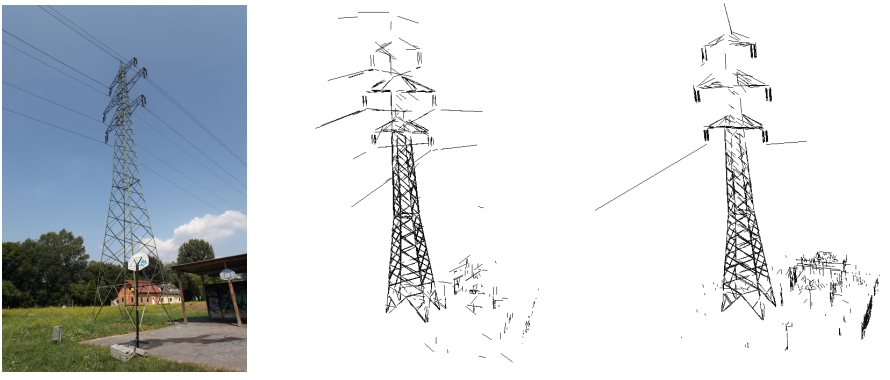
very hard to find or produce correctly labelled datasets. Usually there are simply too many line segments in a real world scene, and whether some structure should be defined as a line segment or not can be ambiguous even for the human eye.

Figure 3 illustrates our result for the *TimberFrame* sequence compared to [5, 8]. To evaluate the accuracy of the reconstruction methods, we estimated the Hausdorff distance between densely sampled points along the line segments and the ground truth CAD model, and computed the mean and root mean square (RMS) error. All three methods deliver accurate results with little variation in the error metrics. The highest accuracy is achieved by [5] ($RMS = 0.094, mean = 0.051$), followed by our method ($RMS = 0.196, mean = 0.065$), and [8] ($RMS = 0.291, mean = 0.162$). While the quality of the results is comparable among all three approaches, our online approach is significantly faster. We manage to obtain the result in 12 minutes, which is more than three times as fast as [5], even though we also perform camera estimation and they assume the cameras to be known. We do not know the exact runtime of [8], but the authors report several hours in their paper.

Figure 4 shows results for a wiry structure, using the *Pylon* sequence by [5]. As we can see, our approach has even less outliers due to the improved scoring approach and the automatic grouping radius selection. The performance increase is even more significant for this testcase.

All experiments in this paper have been performed on a desktop PC equipped with an *Intel Core i5,* 4×3.4 *GHz* CPU and an *nVidia geForce GTX560* graphics card, used for SIFT feature extraction. In order to increase the performance, all images are scaled down from 10 megapixel to FullHD resolution for 2D line segment detection and incremental reconstruction (but not for pose estimation and SIFT feature extraction). Evaluation showed that this decreases the runtime of line segment detection from approximately 5 seconds per image to less than 1 second, while still delivering almost the same results. Since we only consider line segments with a minimum length of 1% of the image diagonal, the number of detected segments remains approximately constant.

We have introduced a number of parameters in Section 3. While the majority of them remains unchanged for all scenarios, the uncertainty σ may have to be adapted since it depends on the size of the target object in the image. Even though this seems to be hard to predict, extensive evaluation showed that setting σ to 1 pixel (for FullHD images) leads to



(a) (b) (c)

Figure 4: (a) An example image from the *Pylon* sequence by [5] (106 images). (b) Reconstruction obtained by [5] (67 minutes, lines only). (c) Our reconstruction, $M = 10$, $\lambda = 4$ (9 minutes, including SfM).

accurate results for the general case. For some sequences it might be beneficial to increase the number of nearest neighbors M as well. Since we always use the M nearest visual neighbors with the smallest view angle difference, it might occur that we get poorly triangulated hypotheses if all neighboring views have similar camera rays. If few images of the scene are available, the minimum number of hypothesis participants λ could be decreased to densify the result. Accordingly, if we have a large set of images λ could be increased to improve outlier robustness. Figure 5 illustrates the effect of parameter alteration for the *Pylon* sequence. As we can see, increasing M leads to improved results at the expense of performance, while decreasing λ introduces outliers.

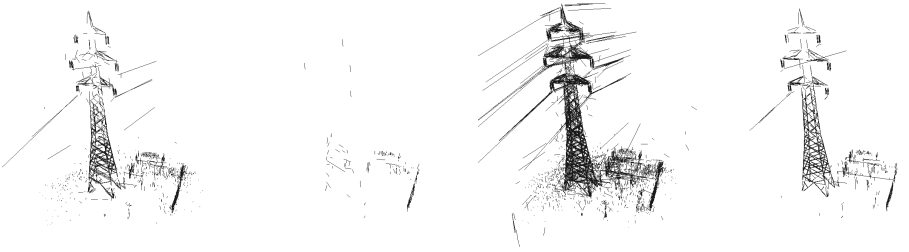
5 Conclusion

We have presented a novel approach to perform on-the-fly line-based 3D reconstruction. We have shown that it is possible to extend time consuming offline algorithms to work online, without sacrificing the accuracy of the results. Our algorithm is able to handle a wide range of man-made scenery, even wiry structures are correctly reconstructed. Additionally, being able to choose a grouping radius in image space rather than in world space (with probably unknown scale) eliminates excessive parameter tuning, and allows to create fast results.

While the performance is massively increased compared to offline algorithms, the approach is not yet real-time capable. The bottleneck is the matching procedure, since for each line segment a large number of possible matches have to be evaluated depending on the objects in the scene. To further improve the performance, we plan to incorporate additional matching criteria and to evaluate possible matches in parallel using the GPU.

Acknowledgements

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



(a) $M = 5, \lambda = 3$ (7) (b) $M = 5, \lambda = 5$ (7) (c) $M = \infty, \lambda = 3$ (10) (d) $M = \infty, \lambda = 5$ (11)

Figure 5: Example results using modified parameters M and λ. As we can see, alteration of M changes the density of the result and has a high influence on the runtime, since larger values generally mean more possible matches. Decreasing λ also increases the density but simultaneously introduces a large number of outliers, while increasing λ leads to a much sparser reconstruction. The numbers in brackets are the runtimes in minutes.

References

[1] A.J. Davison. Real-time simultaneous localisation and mapping with a single camera. *International Conference on Computer Vision*, 2002.

[2] A. Elqursh and A. Elgammal. Line-based relative pose estimation. In *Conference on Computer Vision and Pattern Recognition*, 2011.

[3] A.P. Gee and W. Mayol-Cuevas. Real-time model-based SLAM using line segments. *Advances in Visual Computing*, 2006.

[4] K. Hirose and H. Saito. Fast line description for line-based SLAM. *British Machine Vision Conference*, 2012.

[5] M. Hofer, A. Wendel, and H. Bischof. Line-based 3D reconstruction of wiry objects. *Computer Vision Winter Workshop*, 2013.

[6] C. Hoppe, M. Klopschitz, M. Rumpler, A. Wendel, S. Kluckner, H. Bischof, and G. Reitmayr. Online feedback for structure-from-motion image acquisition. *British Machine Vision Conference*, 2012.

[7] K. Hyunwoo and L. Sukhan. A novel line matching method based on intersection context. *International Conference on Robotics and Automation*, 2010.

[8] A. Jain, C. Kurz, T. Thormaehlen, and H. Seidel. Exploiting global connectivity constraints for reconstruction of 3D line segments from images. *Conference on Computer Vision and Pattern Recognition*, 2010.

[9] T. Lemaire and S. Lacroix. Monocular-vision based SLAM using line segments. *International Conference on Robotics and Automation*, 2007.

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

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

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

[13] R.G. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall. LSD: A fast line segment detector with a false detection control. *Transactions on Pattern Analysis and Machine Intelligence*, 2010.

[14] L. Zhang and R. Koch. Line matching using appearance similarities and geometric constraints. In *Pattern Recognition*, Lecture Notes in Computer Science, 2012.

[15] Y. Zhang, H. Yang, and X. Liu. A line matching method based on local and global appearance. *International Congress on Image and Signal Processing*, 2011.

[16] W. Zhiheng, W. Fuchao, and H. Zhanyi. MSLD: A robust descriptor for line matching. *Pattern Recognition*, 2009.