THE ACCURACY OF SCENE RECONSTRUCTION FROM IR IMAGES BASED ON KNOWN CAMERA POSITIONS

An Evaluation with the Aid of LiDAR Data

Stefan Lang, Marcus Hebel and Michael Kirchhof
FGAN-FOM Research Institute for Optronics and Pattern Recognition, Gutleuthausstr. 1, D-76275 Ettlingen, Germany

Keywords: Infrared, LiDAR, evaluation, structure from motion, sensor data fusion.

Abstract: A novel approach for the evaluation of a 3D scene reconstruction based on LiDAR data is presented. A system for structure computation from aerial infrared imagery is described which uses known pose and position information of the sensor. Detected 2D image features are tracked and triangulated afterwards. Each estimated 3D point is assessed by means of its covariance matrix which is associated with the respective uncertainty. Finally a non-linear optimization (Gauss-Newton iteration) of 3D points yields the resulting point cloud.
The obtained results are evaluated with the aid of LiDAR data. For that purpose we quantify the error of a reconstructed scene by means of a 3D point cloud acquired by a laser scanner. The evaluation procedure takes into account that the main uncertainty of a Structure from Motion (SfM) system is in direction of the line of sight. Results of both the SfM system and the evaluation are presented.

1 INTRODUCTION

Structure from Motion (SfM) is still an important issue in Computer Vision. Over the past years there has been done a lot of work in this area and sophisticated algorithms have been developed for many applications. In case of an airborne system, as considered in this work, the extracted 3D information can for example be used for modelling buildings, for detecting and tracking vehicles or for supporting the navigation. However in case of both free motion and a fast moving sensor like a camera mounted on a helicopter, new problems emerge because of motion blur. Additionally, if an infrared sensor is used, the acquired images contain much more noise and are of lower resolution compared to video images.

In this paper a newly developed system for automatic reconstruction of a scene from aerial infrared imagery is described. For more accuracy in the reconstruction and to overcome the drawbacks of feature tracking in IR images, pose information measured by an IMU (Inertial Measurement Unit) are used.

Normally reconstruction results of SfM systems given as point cloud as well as dense reconstructed models look very impressive but no quantitative measures of their correctness are available. One of the main contribution of our work is the novel approach to the evaluation of the scene reconstruction with the aid of a LiDAR (Light Detecting And Ranging) system, taking into account that the main uncertainty of an SfM system is in direction of the line of sight. The laser scanner employed provides very precise range data of the scene, which are well suited as basis for the evaluation of our SfM system. Both the IR sensor and the laser scanner use the same INS (Inertial Navigation System) and therefore have the same motion errors. The main error is called ”drift” which occurs because of dead reckoning. However, the laser scanner can be used as ground truth for the assessment of the SfM system anyway if no absolute evaluation is needed. That is the case for our evaluation of the accuracy of an SfM system with known motion information.

The rest of this paper is organized as follows. An overview of related work is given in section 2. In section 3 the sensors used are briefly described. The developed algorithms and details of the SfM system are presented in section 4. As main part of this paper, section 5 describes the evaluation in depth. Section 6 closes the paper with a summary and future work.

2 RELATED WORK

This section is divided into two parts according to the structure of this paper. First, existing systems for scene reconstruction are described in relation to

Lang S., Hebel M. and Kirchhof M. (2008).
THE ACCURACY OF SCENE RECONSTRUCTION FROM IR IMAGES BASED ON KNOWN CAMERA POSITIONS - An Evaluation with the Aid of LiDAR Data.
In *Proceedings of the Third International Conference on Computer Vision Theory and Applications*, pages 439-446
DOI: 10.5220/0001074404390446

our system. Second, registration and evaluation algorithms for 3D point clouds as well as the alignment of 2D images onto range information acquired by a LiDAR system are examined.

In the field of Structure from Motion there are many different approaches to reconstruct a scene from a given video sequence. Nistér (2001) describes a system which processes the whole chain from images taken by an uncalibrated camera to a dense reconstruction of the scene.

An advanced system, working with multiple cameras mounted on a ground vehicle is presented in (Akbarzadeh et al., 2006). An urban scene is reconstructed from long video sequences of high resolution. The collected GPS and IMU data are only used to geo-register the reconstructed models. Their system is divided into two parts. The first part works online and generates a sparse 3D point cloud. In the second part the computational expensive dense reconstruction for the whole video sequence is performed.

In (Nistér et al., 2006) visual odometry is compared to pose and position estimation based on DGPS (Differential GPS) and IMU measures. It is shown that it is possible to use ego motion estimated from video for an accurate image based navigation. But – as in the other systems described above – video images of high resolution are used instead of IR images as considered in our work.

To evaluate an extracted 3D point cloud there are several different possibilities. A very good survey of the field of registration and fusion of point clouds is given in (Rodrigues et al., 2002).

For the registration of point clouds ICP (Iterative Closest Point), as introduced by Besl and McKay (1992), has been established as standard approach. However in the case of one dense and one sparse point cloud as in our case, the standard algorithm has to be adjusted. In (Pottmann et al., 2004) an approach is suggested in which a point cloud is aligned to a surface. They introduce an alternative concept to the ICP algorithm which relies on local quadratic approximants to the squared distance function of the surface.

Alignment of video onto 3D point clouds is the topic of (Zhao et al., 2005). They propose a method to align a point cloud extracted from video onto a point cloud obtained by a 3D sensor like a laser scanner. After that a 3D model of the scene is built, which differs from our objective to perform an evaluation.

3 EXPERIMENTAL SETUP

As sensor platform a helicopter is used. The different sensors are installed in a pivot-mounted sensor carrier on the right side of the helicopter. The following sensors are used:

IR Camera. An AIM 640QMW is used to acquire mid-wavelength (3-5μm) infrared light. The lens has a focal length of 28mm and a field of view of $30.7^\circ \times 23.5^\circ$.

LiDAR. The Riegl Laser Q560 is a 2D scanning device which illuminates in azimuth and elevation with short laser pulses. The distance is calculated based on the time of flight of a pulse. It covers almost the same field of view as the IR camera.

INS. The INS (Inertial Navigation System) is an Applanix POS AV system which is specially designed for airborne usage. It consist of an IMU and a GPS system. The measured pose and position are Kalman-filtered to smooth out errors in the GPS.

The range resolution of the LiDAR system is about 0.02m according to the specifications given by the manufacturer. The absolute accuracy specifications of the Applanix system state the following accuracies (RMS): position 4-6m, velocity 0.05m/s, roll and pitch 0.03° and true heading 0.1°. As described in the introduction, both the SfM system and the LiDAR system use the same INS and thus have the same absolute position and pose error. Therefore the accuracy of the INS is irrelevant for the comparison of both systems.

Because of the low noise of the laser system (some centimeters) compared to the SfM system (some meters) that noise will be neglected in this paper.

For the later fusion to be successful, the measured data is registered in time first. Different frequencies as well as the integration time of the IR camera are taken into account.

Both the coordinate frame of the IR camera and of the laser scanner are given in respect of the INS reference coordinate frame. Therefore coordinate transformations between the IR camera and the laser scanner are known and later evaluation is possible.

4 3D RECONSTRUCTION

The developed system automatically calculates a set of 3D points from given IR images, pose and position information. For implementation Intel's computer vision library OpenCV (Intel, 2006) is used.

A system overview is given in figure 1. After initialization, detected features are tracked image by image. To minimize the number of mismatches between the corresponding features in two consecutive images the algorithm checks the epipolar constraint by means

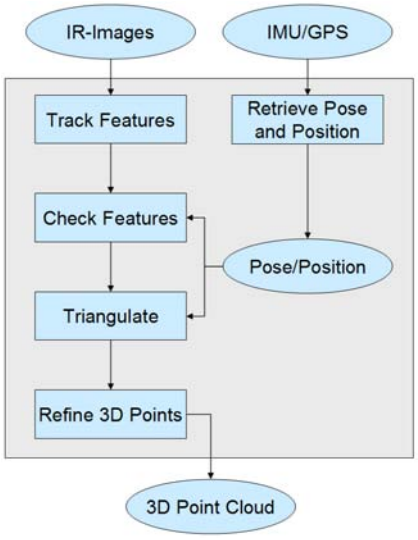

Figure 1: System overview of the modules. Features are tracked in consecutive images and checked for satisfaction of the epipolar constraint. Linear Triangulation of each track of the checked features gives the 3D information. In both steps – constraint checking and triangulation – the retrieved pose and position information is used. Finally each 3D point is evaluated and optimized.

of the given pose information retrieved from the INS. Triangulation of the tracked features results in the 3D points. Each 3D point is assessed with the aid of its covariance matrix which is associated with the respective uncertainty. Finally a non-linear optimization yields the completed point cloud.

The modules are described in detail in the rest of this section.

4.1 Tracking Features

To estimate the motion between two consecutive images the OpenCV version of the KLT tracker (Shi and Tomasi, 1994) is used. This pyramidal implementation is very computational efficient and does a better job of handling features near the image borders than the original version. A precise description can be found in (Bouguet, 2000).

The algorithm tracks point features such as corners or points with a neighbourhood rich in texture. For robust tracking a measure of feature similarity is used. This weighted correlation function quantifies the change of a tracked feature between the current image and the image of initialization of the feature.

4.2 Retrieve Pose and Position

The INS gives the Kalman-filtered absolute position and pose of the reference coordinate frame. Three angles (roll, pitch and yaw) describe the orientation in space. The position is given in longitude and latitude according to World Geodetic System 1984 (WGS84).

After converting the data into absolute rotation matrices R_i^{abs} and position vectors $\tilde{\mathbf{C}}_i$ for the absolute pose and position of the i-th camera in space, the projection matrices P_i, needed for triangulation, are calculated as follows

$$\mathrm{P}_i = \mathrm{K}\mathrm{R}_i^{abs}[\mathrm{I}_3 | -\tilde{\mathbf{C}}_i], \tag{1}$$

where I_3 is the unit matrix. Altogether P_i is a 3×4-matrix. The intrinsic camera matrix K is defined as

$$\mathrm{K} = \begin{pmatrix} f & 0 & -x_c \\ 0 & f & -y_c \\ 0 & 0 & 1 \end{pmatrix}, \tag{2}$$

with focal length f and principle point (x_c, y_c).

4.3 Epipolar Constraint

Let R and $\mathbf{t}$ be the relative rotation and translation of the camera between two consecutive images. Further let $\mathbf{x}$ be a feature in one image and $\mathbf{x}'$ be a feature in the other image. If both features belong to the same point $\mathbf{X}$ in 3-space, then the image points have to satisfy the following constraint

$$\mathbf{x}'^{\mathrm{T}}\mathrm{F}\mathbf{x} = 0. \tag{3}$$

The fundamental matrix F is the unique (up to scale) rank 2 homogeneous 3×3-matrix which satisfies the constraint given in equation (3). To retrieve the fundamental matrix, as described in (Hartley and Zisserman, 2004), the essential matrix has to be computed by means of the given relative orientation of the cameras

$$\mathrm{E} = [\mathbf{t}]_\times \mathrm{R}. \tag{4}$$

The 3×3-matrix $[\mathbf{t}]_\times$ is the skew-symmetric matrix of the vector $\mathbf{t}$. Next, F can be calculated directly as follows

$$\mathrm{F} = \mathrm{K}^{\mathrm{T}^{-1}}\mathrm{E}\mathrm{K}^{-1} = \mathrm{K}^{\mathrm{T}^{-1}}[\mathbf{t}]_\times \mathrm{R}\mathrm{K}^{-1}. \tag{5}$$

To check if $\mathbf{x}'$ is the correct image point corresponding to the tracked point feature $\mathbf{x}$ of the previous image, $\mathbf{x}'$ has to lie on the epipolar line $\mathbf{l}'$ defined as

$$\mathbf{l}' \equiv \mathrm{F}\mathbf{x}. \tag{6}$$

Normally a corresponding image point doesn't lie exactly on the epipolar line, due to noise in the images and inaccuracies in pose measures. Therefore if the distance (error) of $\mathbf{x}'$ to $\mathbf{l}'$ is too large, the found feature is rejected and the track ends.

4.4 Triangulation

During iteration over the IR images, tracks are built of detected and tracked point features. If an image feature can't be retrieved or doesn't satisfy the epipolar constraint, the track ends and the corresponding 3D point $\mathbf{X}$ is calculated. In (Hartley and Sturm, 1997) a good overview of different methods for triangulation is given as well as a description of the method used in our system.

Let $\mathbf{x}_1 \dots \mathbf{x}_n$ be the image features of the tracked 3D point $\mathbf{X}$ in n images and $\mathrm{P}_1 \dots \mathrm{P}_n$ the projection matrices of the corresponding cameras. Each measurement $\mathbf{x}_i$ of the track represents the reprojection of the same 3D point

$$\mathbf{x}_i \equiv \mathrm{P}_i \mathbf{X} \qquad \text{for } i = 1 \dots n. \tag{7}$$

With the cross product the homogeneous scale factor of equation (7) is eliminated, which leads to $\mathbf{x}_i \times (\mathrm{P}_i \mathbf{X}) = 0$. Subsequent there are two linearly independent equations for each image point. These equations are linear in the components of $\mathbf{X}$, thus they can be written in the form $\mathrm{A}\mathbf{X} = 0$ with

$$\mathrm{A} = \begin{bmatrix} x_1 \mathbf{p}_1^{3\mathrm{T}} - \mathbf{p}_1^{1\mathrm{T}} \\ y_1 \mathbf{p}_1^{3\mathrm{T}} - \mathbf{p}_1^{2\mathrm{T}} \\ \vdots \\ x_n \mathbf{p}_n^{3\mathrm{T}} - \mathbf{p}_n^{1\mathrm{T}} \\ y_n \mathbf{p}_n^{3\mathrm{T}} - \mathbf{p}_n^{2\mathrm{T}} \end{bmatrix}, \tag{8}$$

where $\mathbf{p}_i^{k\mathrm{T}}$ are the rows of P_i. The 3D point $\mathbf{X}$ is the unit singular vector corresponding to the smallest singular value of the matrix A.

4.5 Non-Linear Optimization

After triangulation the reprojection error can be estimated as follows

$$\varepsilon_i = \begin{pmatrix} \varepsilon_i^x \\ \varepsilon_i^y \end{pmatrix} = d(\mathbf{X}, \mathrm{P}_i, \mathbf{x}_i) = \begin{pmatrix} x_i - \frac{\mathbf{p}_i^1 \mathbf{X}}{\mathbf{p}_i^3 \mathbf{X}} \\ y_i - \frac{\mathbf{p}_i^2 \mathbf{X}}{\mathbf{p}_i^3 \mathbf{X}} \end{pmatrix}. \tag{9}$$

To calculate the covariance matrix, the Jacobian matrix J, which is the partial derivative matrix $\partial\varepsilon/\partial\mathbf{X}$, is needed first:

$$\mathrm{J} = \begin{bmatrix} \frac{\partial \varepsilon_1^x}{\partial X} & \frac{\partial \varepsilon_1^x}{\partial Y} & \frac{\partial \varepsilon_1^x}{\partial Z} \\ \frac{\partial \varepsilon_1^y}{\partial X} & \frac{\partial \varepsilon_1^y}{\partial Y} & \frac{\partial \varepsilon_1^y}{\partial Z} \\ \vdots & & \\ \frac{\partial \varepsilon_n^x}{\partial X} & \frac{\partial \varepsilon_n^x}{\partial Y} & \frac{\partial \varepsilon_n^x}{\partial Z} \\ \frac{\partial \varepsilon_n^y}{\partial X} & \frac{\partial \varepsilon_n^y}{\partial Y} & \frac{\partial \varepsilon_n^y}{\partial Z} \end{bmatrix} \tag{10}$$

With the assumption of a reprojection error of one pixel, the back propagated covariance matrix of a 3D point is calculated

$$\Sigma_{\mathbf{X}} = \left(\mathrm{J}^{\mathrm{T}}\,\mathrm{J}\right)^{-1}. \tag{11}$$

The euclidean norm of $\Sigma_{\mathbf{X}}$ gives an overall measure of the uncertainty of the 3D point $\mathbf{X}$, and enables the algorithm to reject poor triangulation results.

With non-linear optimization, a calculated 3D point can be corrected. Let $\varepsilon(\mathbf{X}+\xi)$ be the corrected version of (9) by a vector ξ. A first order Taylor approximation yields

$$\varepsilon(\mathbf{X}+\xi) \approx \varepsilon + \mathrm{J}\xi. \tag{12}$$

Then the correction ξ can be estimated assuming $\varepsilon(\mathbf{X}+\xi) = 0$ by

$$\xi = -\left(\mathrm{J}^{\mathrm{T}}\,\mathrm{J}\right)^{-1} \mathrm{J}^{\mathrm{T}} \varepsilon. \tag{13}$$

This procedure can be repeated until the corrections of the 3D point fall below a certain threshold. In our case the algorithm performs only one iteration which showed to be a reasonable optimization.

4.6 Results

We tested our system on an infrared image sequence with 470 images with a resolution of 624×480. During the recording, the pitch angle of the sensor carrier had been 45° from the initial down looking position. Figure 2 shows four sample images of the processed sequence. To get a better overview of the whole sequence, figure 3 displays a panorama of all images. It was generated with image to image homographies calculated with the tracked features, see section 4.1.

Working on that sequence and taking pose and position information into account the system calculates an optimized point cloud of about 17,500 points,

Figure 2: Four sample images of the sequence.

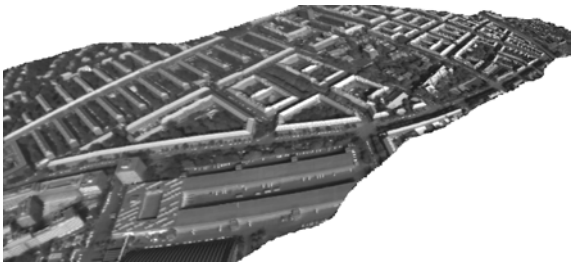

Figure 3: Panorama of the processed image sequence.

Figure 4: Calculated point cloud of the image sequence with the magnification of one building. The overall number of points is 17,606.

see figure 4. One building is magnified for a better view. The height of each point is coded in its color. Although it is a sparse reconstruction, the structure of each building is well distinguishable and there are only a few outliers due to the performed optimization.

5 EVALUATION WITH LIDAR

The previously presented SfM system reconstructs the area by a sparse 3D point cloud, see figure 4. Next the calculated result shall be evaluated quantitatively with the aid of LiDAR data. Therefore a novel benchmark for a SfM system by means of given 3D information has been developed. For the evaluation an error measure for the overall accuracy of the point cloud is obtained.

5.1 3D Reconstruction with LiDAR

The employed LiDAR system scans the area line-by-line. The distance d of each 3D point to the sensor is calculated based on the time of flight t of the emitted light pulse as follows

$$d = \frac{t}{2} c_0, \tag{14}$$

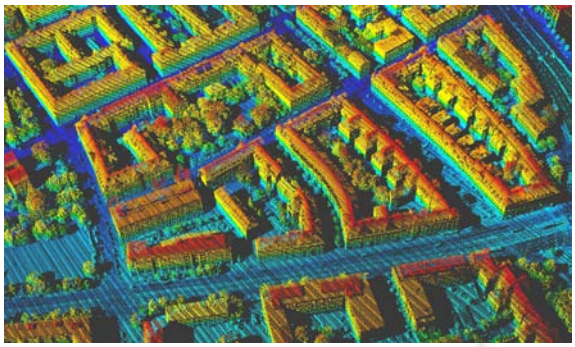

Figure 5: Reconstructed area with color coded height of the LiDAR points.

where c_0 is the speed of light. After conversion into the reference coordinate frame an almost dense reconstruction in form of a 3D point cloud is obtained. Figure 5 shows a cutout of the area.

Let $\hat{d}$ be the correct distance of a point to the camera and Δd the measuring error, then the following relation is obtained

$$\hat{d} = d + \Delta d. \tag{15}$$

With the angle of beam ψ and count of measures κ along a scan line, the average point spacing $\Delta\bar{b}$ can be calculated as follows:

$$\Delta\bar{b} = \frac{2\hat{d}\tan(\frac{\psi}{2})}{\kappa} \tag{16}$$

$$= \frac{2d\tan(\frac{\psi}{2})}{\kappa} + \frac{2\Delta d\tan(\frac{\psi}{2})}{\kappa}. \tag{17}$$

With the second summand of equation (17) it is also possible to estimate the average error of the distance between points along a scan line.

In our case the error of the used LiDAR system in direction of the line of sight is about 0.2m at a distance of 550m. With 1000 measures along one scan line and an angle of beam of 60°, the obtained average spacing between points is about 0.7m. The calculated error between points, see equation (17), is less than 1cm and therefore this error can be neglected.

5.2 Comparison

Both point clouds have been acquired during the same flight. Our SfM algorithm has calculated about 17,500 points. In contrast, the dense point cloud obtained by the 3D scanner consists of more the 1.5 million points. Because both 3D point clouds are georeferenced it is possible to superimpose the sparse SfM point cloud to the dense LiDAR point cloud, see figure 6.

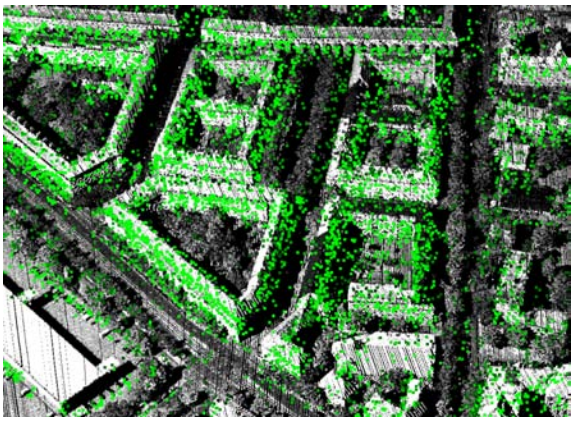

Figure 6: LiDAR point cloud with superimposed (green) points calculated by the SfM system.

Though it is possible to superimpose one point cloud to the other, no quantitative measure of the accuracy can be obtained in this way because of the following reasons:

- Points of the SfM point cloud represent tracked image features which were good to track whereas points of the LiDAR data are almost uniformly distributed over the observed area.
- There is a significant difference in the size of both point clouds.
- There is a large difference especially in direction of line of sight of both point clouds. The laser points have a variance of approximately 0.2m whereas the 3D points reconstructed by the SfM system are distributed with a variance of several meters in that direction.

Using the ICP algorithm results in a translation and rotation of the sparse point cloud which minimize the distances between the closest points of both point clouds respectively. However there is no guarantee that in each case the closest point is the actually corresponding point. Additionally, it is hard to obtain a significant measure from that calculated displacement.

5.3 Evaluation Approach

As described before a 3D point reconstructed from IR images is mostly inaccurate in the line of sight. Taking that into account as well as the other listed issues of section 5.2 we developed a novel evaluation approach described in the following.

For each 3D point of the SfM point cloud, the algorithm iterates through the IR images in which the considered 3D point was detected during tracking. Then the following steps are executed:

1. Create a line through camera center and considered 3D point of the SfM point cloud
2. Find k nearest 3D points from LiDAR point cloud to that line
3. Calculate distances of found k neighbours to considered 3D point of the SfM point cloud

In each iteration the smallest distance, calculated in step (3), is added to a sum. For each 3D point this sum is normalized afterwards by the number of considered views, which yields the average error.

The following pseudo code describes the algorithm more clearly and it simplifies considerations of complexity of the program.

```
For i = 1..n (all SfM 3D points) do
  Sum[i] = 0
  For j = 1..m (all IR images) do
    If 3D point is visible
      (1) Calculate line through camera center
          and 3D point
      (2) Get k nearest 3D points of LiDAR
          point cloud to that line
          (2.1) dist = distance of nearest
                       one to considered
                       SfM point
      (3) Sum[i] = Sum[i] + dist
    End
  End
  Eval[i] = Sum[i]/m
End
```

After running the program, the i-th entry of the array `Eval` is the average distance of the i-th SfM 3D point to all of its LiDAR counterparts from considered views. As wanted the algorithm prefers LiDAR 3D points with small distances perpendicular to the line of sight during the search for corresponding 3D points.

With that result, we obtained an overall measure for the accuracy of reconstructed 3D points. Consideration of the complexity of our program shows the hotspot step (2) where the k nearest 3D points of LiDAR data to a line are calculated. The complexity of the algorithm is $\mathcal{O}(nmq)$, with n and m defined as in the pseudo code above and q the number of LiDAR points. Because of the negligible number of considered images m compared to the amount of points, m can be omitted and therefore the complexity of $\mathcal{O}(nq)$ is obtained.

For computational simplification the approach to search corresponding LiDAR points to a presently considered SfM point is changed. Instead of the iteration over all LiDAR points for each 3D point of the SfM point cloud and for each considered view, the LiDAR point cloud is reprojected to the view of each IR camera. This computation is executed once and images comparable to the IR images are obtained. With

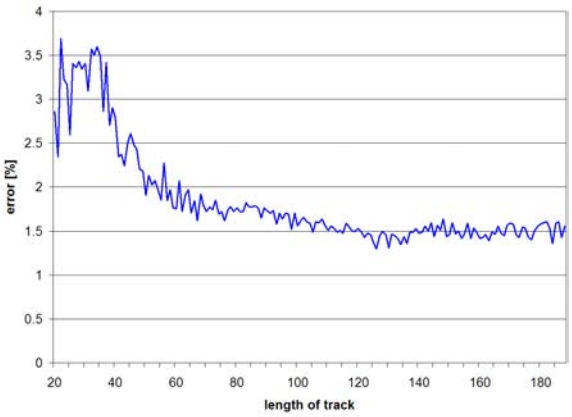

Figure 7: Relationship between the relative error to the distance to the IR camera and the length of track. With increasing length of track, the accuracy of the reconstructed 3D point gets better.

that result there is a direct correspondence between a previously tracked IR image feature and a reprojected laser point at the same coordinates. To estimate the corresponding laser point, all neighbours of the image feature are taken into account. Those are the points of the laser point cloud, which are close to the line of sight of the currently considered view. The one with the smallest distance to the 3D point of the SfM point cloud is regarded as correspondence. The rest of the program stays unchanged.

5.4 Evaluation Results

Evaluation of the reconstructed scene shows that there are some 3D points with no corresponding LiDAR 3D points. These points lie at the outside margin of the considered scene where no LiDAR data is available. However less than 500 points of total over 17,500 points are affected. Therefore an evaluation of about 17,000 points is possible.

In the following the errors of reconstructed 3D points are given relatively to their distance to the IR camera.

The relation of the error of a reconstructed 3D point and the number of considered views for triangulation is shown in figure 7. As expected, an image feature tracked for a long time leads to a better estimation of the corresponding 3D point than one found in a small number of images. Errors of less than 2 percent are reached for tracks longer than 50 frames.

Overall the average error of the reconstructed scene is about 1.8 percent. The absolute error, average distances of SfM 3D points to LiDAR 3D points, is about 10 meters at sensor-scene distances of about 550m. This error is mainly in direction of the line of sight.

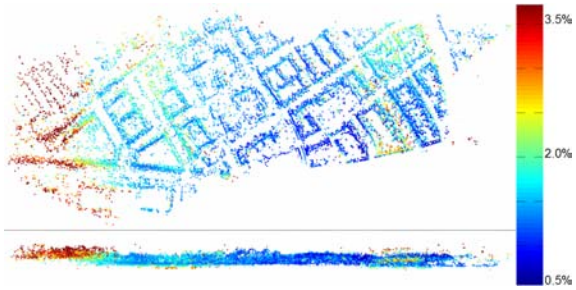

Figure 8: Color coded error of the calculated point cloud. Blue represents a small error whereas the color red represents a high error. The scene is shown from top view (top) and from side view (bottom).

Another presentation of the distribution of the error is given in figure 8. It shows the estimated point cloud with color coded error. Comparably good reconstructed points are blue whereas red points are of lower accuracy. Noticeable is the red area on the left side of the point cloud. There are two reasons which explain this bad reconstructed sector. First, during acquisition the helicopter flew a curve in this area which means that there are a lot of changes between two consecutive images. Second, the considered image sequence started at this section. Therefore average length of track of computed 3D points in this area is less than in the rest of the reconstructed scene.

6 CONCLUSIONS

We have developed a system to reconstruct a scene from an IR sequence. To overcome the drawbacks of IR images used for feature tracking, pose information measured by an IMU are used. For measuring the accuracy of the reconstructed scene, we have presented a novel benchmark by means of given LiDAR data. This evaluation takes the functionality of a SfM system into account.

In future work we will use our developed evaluation method for improvements and tests of our SfM system. Additionally we will test the accuracy of the estimation of ego motion from IR images.

REFERENCES

Akbarzadeh, A., Frahm, J.-M., Mordohai, P., Clipp, B., Engels, C., Gallup, D., Merrell, P., Phelps, M., Sinha, S., Talton, B., Wang, L., Yang, Q., Stewenius, H., Yang, R., Welch, G., Towles, H., Nister, D., and Pollefeys, M. (2006). Towards urban 3d reconstruction from video. In *3DPVT '06: Proceedings of the Third International Symposium on 3D Data Processing, Visu-*

alization, and Transmission (3DPVT'06), pages 1–8, Washington, DC, USA. IEEE Computer Society.

Besl, P. J. and McKay, N. D. (1992). A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256.

Bouguet, J.-Y. (2000). Pyramidal implementation of the lucas kanade feature tracker. The paper is included in the OpenCV distribution, see www.intel.com/technology/computing/opencv.

Hartley, R. I. and Sturm, P. (1997). Triangulation. *Computer Vision and Image Understanding*, 68(2):146–157.

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

Intel (2006). Opencv - open source computer vision library. www.intel.com/technology/computing/opencv.

Nistér, D. (2001). *Automatic Dense Reconstruction from Uncalibrated Video Sequences*. PhD thesis, Royal Institute of Technology KTH.

Nistér, D., Naroditsky, O., and Bergen, J. (2006). Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 23(1).

Pottmann, H., Leopoldseder, S., and Hofer, M. (2004). Registration without icp. *Computer Vision and Image Understanding*, 95(1):54–71.

Rodrigues, M., Fisher, R., and Liu, Y. (2002). Special issue on registration and fusion of range images. *Computer Vision and Image Understanding*, 87(1-3):1–7.

Shi, J. and Tomasi, C. (1994). Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, pages 593–600, Seattle.

Zhao, W., Nistér, D., and Hsu, S. (2005). Alignment of continuous video onto 3d point clouds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1305–1318.