Real-time 2D Video/3D LiDAR Registration

C. Bodensteiner
Fraunhofer IOSB
christoph.bodensteiner@iosb.fraunhofer.de

M. Arens
Fraunhofer IOSB
michael.arens@iosb.fraunhofer.de

Abstract

Progress in LiDAR scanning has led to the availability of large scale LiDAR datasets for urban areas.

We use such pre-acquired data to determine the poses of 2D monocular cameras highly accurately in real-time. This is achieved by first correctly aligning key-frames of the multi-modal data using a combination of feature and intensity-based 2D/3D registration methods. The online pose is then determined in realtime by densely sampling and tracking features within the 2D video stream. The 3D coordinates of these features are determined by a fast GPU-based backprojection. The observed 2D/3D feature data is then fused using a recursive Bayesian filter in order to exploit temporal coherency.

The method is evaluated using ground truth camera trajectories and different filter implementations. The proposed registration and filter framework executes at video-frame rate and it is up to 15% more accurate then a registration only solution.

Applications are numerous and include, for instance, augmented-reality applications, online georeferentiation or metric online 3D reconstruction from monocular video data.

1. Introduction

The accurate 3D mapping of the environment using LiDAR sensors has seen great progress in the last decade. This has led to the availability of large scale LiDAR datasets for urban areas. In this work we describe a realtime method for using such previously acquired LiDAR scans in combination with a monocular 2D vision system.

We use the backscattered laser intensity information to generate synthetic 2D views, which in turn enable multi-modal appearance-based registration of camera images w.r.t. the coordinate system of the global scan. We provide insights on how to pre-process raw LiDAR datasets and how to achieve an accurate online multi-modal registration using Bayesian filter techniques.

2D Video/3D LiDAR Registration: There exists a vast body of literature concerning 2D/3D registration methods. Close to our application scenario are the following works: Mastin et al. [7] register height color coded 2D renderings with camera images using mutual information (MI) [10]. Vasile et al. [9] derive pseudo-intensity images from LiDAR data, including shadows, to allow for a 2D/3D registration with aerial imagery. Feature based approaches [5, 2] mostly rely on the detection and alignment of geometric features like corners or line segments in the camera image and projections of those from the 3D data. For a good introduction to filter based localization methods we refer the reader to the work of Thrun et al. [8].

Contribution: To our best knowledge realtime multi-modal video/LiDAR registration for metric online localization based on large-scale LiDAR data have not been presented in the literature. We show how such a system can be built. In particular, we point out what the enabling assumptions are and what our design of the relevant probability distribution functions is. We further show how recursive Bayesian filter techniques are integrated within the 2D/3D registration to exploit temporal coherency. Additionally, we propose to use LiDAR point clouds as a metric multi-modal calibration body for an accurate intrinsic calibration of the camera.

The outline of the paper is as follows: first the key elements of the method are described. The registration accuracy is then evaluated using ground truth camera trajectories; in the evaluation different recursive filter techniques are evaluated. Finally, the results and further research directions are discussed.

2. Method

The proposed method comprises the following main elements:

(A) The raw LiDAR point cloud data is processed to allow for efficient rendering of 2D synthetic views. This

involves the 3D registration of multiple terrestrial and aerial laser scans and a multi-scale representation via octree downsampling methods using the PCL library.

(B) The rendered views enable appearance-based registration with keyframe camera images. For each camera keyframe, the registration is based on local feature correspondences and a 2D/3D PnP solver [6]. The registration can then be refined using mutual information/gradient correlation (MI/GC) [10, 4]. With multiple keyframes registered, the intrinsic camera parameters can be also accurately computed by optimizing the distance measure over the intrinsic parameters.

(C) A dense set of features is extracted and tracked in the camera images and their corresponding 3D coordinates are computed using the keyframe registrations; for feature tracking we used Harris-Foerstner corners in combination with a sparse Lukas-Kanade optic flow method. The resulting dense 2D/3D correspondences are employed in a robust, RANSAC-based realtime 2D/3D PnP solver [6] in combination with a particle filtering backend.

2.1 (A) - LiDAR Scan Preprocessing

We generate synthetic intensity images from the 3D LiDAR data. The intensity information for the synthetic views stems from the backscattered laser pulse information. Local features are extracted from these images using SURF descriptors. The 3D-coordinates of the features are determined based on the GPU-depth buffer information at the feature detector positions.

3D LiDAR Scans Registration: The registration of 2D images with multiple raw LiDAR datasets requires a common scan coordinate system and therefore the prior registration of all local laser scans. We choose the local coordinate system of a central LiDAR dataset as reference system. We then extract local features from the virtual views to automatically find 2D/2D appearance based correspondences between the scanning positions. Corresponding features are backprojected to obtain 3D/3D correspondences for the estimation of the rigid 3D transformation between the scans. The transformation is robustly estimated using Horns method.

2.2 (B) - Keyframe Pose Registration

For the multi-modal 2D/3D keyframe registration we utilize a point-based rendering approach to generate synthetic 2D views from the 3D dataset. 2D/2D correspondences with the camera image are robustly identified by searching for local regions of features where corresponding features have similar geometric relationships by employing a Generalized Hough Transform

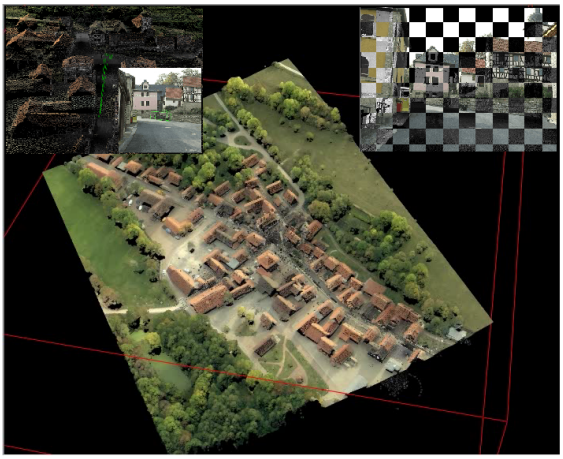


Figure 1. Combined airborne and terrestrial LiDAR-scan 3D background model.

[1]. The 3D positions for the synthetically generated 2D features can easily be determined using the GPU depth buffer information. The 2D/3D feature-based registration is carried out by the Ransac-based PnP solver [6] (inlier threshold 3px,2500 iterations). The resulting pose is refined by an intensity-based registration.

Intensity-Based Pose Refinement: To increase registration accuracy for registered poses an intensity-based similarity measure between rendered views and query images is maximized. The convergence range of this optimization problem is usually small. However, the pose computation based on local features generally provides a sufficiently good starting point. An important choice involves the selection of an appropriate distance measure. The distance measure MI[10] is considered the gold standard similarity measure for multi-modal matching. It measures the mutual dependence of the underlying image-intensity distributions:

$$D_{(MI)}(I_R, I_{T_\theta}) = H(I_R) + H(I_{T_\theta}) - H(I_R, I_{T_\theta}) \quad (1)$$

where $H(I_R)$ and $H(I_{T_\theta})$ are the marginal entropies and

$$H(I_R, I_{T_\theta}) = \sum_{X \in I_{T_\theta}} \sum_{Y \in I_R} p(X,Y) log(\frac{p(X,Y)}{p(X)p(Y)}) \quad (2)$$

is the joint entropy. $p(X,Y)$ denotes the joint probability distribution function of the image intensities X, Y in I_R and I_{T_θ}, and $p(X)$ and $p(Y)$ are the marginal probability distribution functions. We linearly combine the MI measure above with the gradient correlation measure in [4] to enhance robustness and accuracy. To speed up computation time, we restrict the distance

computation to local regions around inlier features in the local feature-based registration.

2.3 (C) - Recursive-Bayes Pose Filtering

The goal of this step is to estimate the camera trajectory, represented by the posterior distribution $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$ and the current belief about the camera pose at time t, represented by the marginal distribution $p(\mathbf{x}_t|\mathbf{y}_{1:t})$.

The state sequence $\{\mathbf{x}_t : t \in I\!N^*\}, x_t \in X$ is assumed to be Markovian with initial distribution $p(\mathbf{x}_0)$ and transition probability $p(\mathbf{x}_t, \mathbf{x}_{t-1})$.

The observations $\{\mathbf{y}_t : t \in I\!N^*\}, \mathbf{y}_t \in Y$ consist of sets of tracked 2D features in combination with their corresponding 3D point positions in the LiDAR background model. The observations are assumed to be conditionally independent given the state e.g. $p(\mathbf{y}_t|\mathbf{x}_t, \mathbf{y}_{1:t-1}) = p(\mathbf{y}_t|\mathbf{x}_t)$, which is approximately true in our data set.

Given the above mentioned independence assumptions this leads to the well known recursive update scheme [3]:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \alpha p(\mathbf{y}_t|\mathbf{x}_t) \int_{\mathbf{x}_{t-1}} p(\mathbf{x}_t|\mathbf{x}_{t-1}) p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) \tag{3}$$

However, the integral does not have a closed form solution except in its basic form. We test two alternatives for overcoming these restrictions. One is to assume linear transitions and observations with additive Gaussian noise distributions and arrives at the Kalman filter equations. The other is particle filtering.

Particle filtering approximates the solution by a set of weighted particles $\{\mathbf{x}_t^{(i)}, \pi_t^{(i)}\}$, where each particle $\mathbf{x}_t^{(i)}$ is an instance of a possible camera position at time t and $\pi_t^{(i)}$ is its corresponding weight, reflecting the confidence level on this position. The above integral is thus approximate by:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \approx \alpha p(\mathbf{y}_t|\mathbf{x}_t) \sum_{i=1}^{n} \pi_{t-1}^{(i)} p(\mathbf{x}_t|\mathbf{x}_{t-1}) \tag{4}$$

In order to implement the particle filter one must specify three distributions: 1) the dynamical distribution $p(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)})$ of the state process, 2) the distribution $p(\mathbf{y}_t|\mathbf{x}_t^{(i)})$ of the observation likelihood, and 3) a proposal distribution $p_p(\mathbf{x}_t^{(i)}|\mathbf{x}_{1:t}^{(i)}, \mathbf{y}_{1:t})$ for updating the particle set.

The state process models the motion of the camera, for which we use a constant acceleration model $p(\mathbf{x}_t|\mathbf{x}_{t-1}) \sim \mathcal{N}(f(\mathbf{x}_{t-1}, \dot{\mathbf{x}}_{t-1}), \Sigma)$, where $\mathcal{N}(\mu, \Sigma)$ denotes a normal distribution with mean μ and covariance Σ and :

$$f(\mathbf{x}_{t-1}, \dot{\mathbf{x}}_{t-1}) = A\mathbf{x}_{t-1} + B\dot{\mathbf{x}}_{t-1} \tag{5}$$

The constant coefficients of the matrices A and B have been determined experimentally.

The distribution of the observation likelihood is based on the reprojection error between the current corresponding tracked 2D image features $\mathbf{m_i}$ and their backprojected 3D points $\mathbf{M_i}$ given the intrinsic parameters K of the camera and the rotation R and translation parameters t from the state $\mathbf{x}_t$:

$$g(R, \mathbf{t}) = \sum_i \|K(R\mathbf{M_i} + \mathbf{t}) - \mathbf{m_i}\| \tag{6}$$

For the computation of weights $\pi_t^{(i)}$ we used the convex Huber cost function $\rho(g(R, \mathbf{t}))$ which intrinsically handles outliers (with reprojection errors $\geq 5px$) by a linear penalty.

The proposal distribution is modeled by $p(\mathbf{x}_t^{(i)}|\mathbf{x}_{1:t}^{(i)}, \mathbf{y}_{1:t}) \sim \mathcal{N}(h(\mathbf{y}_t), \Sigma_p)$, where $h(\mathbf{y}_t)$ refers to the calculated pose parameters from the current tracked feature set.

At each iteration in the recursive update we use the sequential importance resampling scheme in [3], based on the current particle set $\{\mathbf{x}_t^{(i)}, \pi_t^{(i)}\}$. First we sample n particles from the current set according to a low variance sampler as described in [8]. We then update the current particle set with particles sampled from the proposal distribution $p(\mathbf{x}_t^{(i)}|\mathbf{x}_{1:t}^{(i)}, \mathbf{y}_{1:t})$ and finally reweight and normalize the particles according to:

$$\pi_t^{(i)} \propto \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(i)}) p(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)})}{p_p(\mathbf{x}_t^{(i)}|\mathbf{x}_{1:t}^{(i)}, \mathbf{y}_{1:t})} \tag{7}$$

3. Experiments and Numerical Results

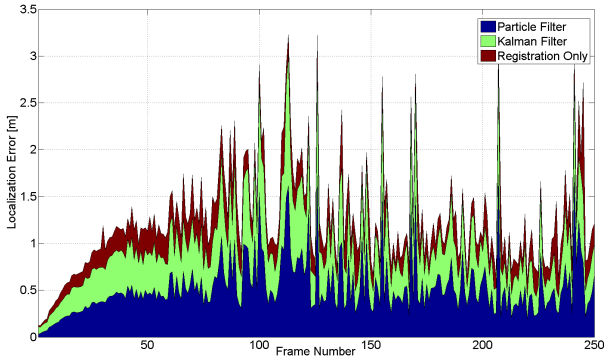
To measure the performance of the proposed framework we used LiDAR scans of a small city and video streams from hand-held cameras and miniUAV systems moving in the same city. We also generated synthetic video-streams based on very dense LiDAR scans ($\approx 10^9$ data points) to generate ground truth camera trajectories. We then measured the increase in performance due to recursive filtering methods over the registration only solution.

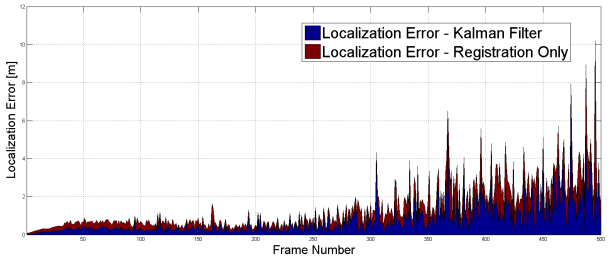
Implementation Details: The algorithms are implemented in C/C++ based on the OpenCV/PCL libraries; we note that many parts of our software framework can be further runtime-optimized. The reported time measurements therefore provide only an early estimate. The

test system is equipped with an i5-2500K processor.

Registration Accuracy: The measurements results are based on the registration of 4500 camera frames. The registration accuracy is defined as the distance of the computed camera positions to the ground truth camera poses. The means and variances of the positional accuracy in x,y,z coordinates expressed in meters were $0,18$ - $1,91$, $0,21$ - $0,77$ and $0,17$ - $1,00$ respectively. The means and variances of the rotational accuracy in x,y,z-Euler angles expressed in degrees were $0,12$ - $0,17$, $-0,07$ - $0,50$ and $-0,24$ - $0,21$, respectively.

Filter Performance: The figure below shows comparative plots of the L_2 distances to the ground truth camera centers, when using a registration only, a Kalman filter and the proposed particle filter for two example sequences of 250/500 frames with one initial keyframe registration. The camera trajectories stem from low altitude miniUAV flights. Note that the best registration results were obtained using the particle filter. The measured positional registration accuracy (deviation from a ground truth pose) (x,y,z) for the kalman filter was $0,23$m (mean) and $1,87$ (variance), $0,24$ with variance $0,46$ and $0,18$ with variance $0,33$ respective $0,18$m (mean) and $0,27$ (variance), $0,20$ with variance $0,19$ and $0,07$ with variance $0,13$ for the particle filter.





Runtime Parameters: The runtime of the registration algorithm depends on various parameters. Parameters that typically need to be considered include the number of tracked features (1000-3000), ransac iterations (1000-5000) and video resolution (720x576px;1280x720px). Setting the parameters to maximal values led to frame rates of 20Hz-24Hz. The Kalman filter solution produces no measurable runtime overhead. Particle filtering however induces a runtime overhead of 6-7fps based on an average particle set size of 400. We note that this overhead is, at least partly, caused by the fact that our particle filter implementation is not parallelized so far.

4. Conclusion And Outlook

In this work we proposed and implemented a realtime 2D/3D vision/LiDAR registration system. The proposed registration and filter framework works at video frame-rate and achieves up to 15% accuracy improvements compared to a solution based on registration only. Prompted by the recent trend of dense methods in computer vision, we intend to investigate windowed bundle adjustment methods in combination with dense variational optic flow for high accuracy vision based localization. We are currently working on sparse filtering methods to investigate the trade-off between accuracy and computational speed for resource limited devices.

References

[1] C. Bodensteiner, W. Huebner, K. Juengling, J. Mueller, and M. Arens. Local multi-modal image matching based on self-similarity. In *Proc. IEEE-ICIP*, 2010.

[2] M. Ding, K. Lyngbaek, and A. Zakhor. Automatic registration of aerial imagery with untextured 3d lidar models. In *CVPR*, 2008.

[3] A. Doucet, N. Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001.

[4] G. P. et. al. A comparison of similarity measures for use in 2-d-3-d medical image registration. *IEEE TMI*, 17(4):586–595, 1998.

[5] C. Frueh, R. Sammon, and A. Zakhor. Automated texture mapping of 3d city models with oblique aerial imagery. In *3DPVT 2004*, pages 396–403, 2004.

[6] V. Lepetit, F. Moreno-Noguer, and P. Fua. Epnp: An accurate o(n) solution to the pnp problem. *International Journal of Computer Vision*, 81:155–166, 2009.

[7] A. Mastin, J. Kepner, and J. Fisher. Automatic registration of lidar and optical images of urban scenes. In *CVPR*, 2009.

[8] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, Cambridge, MA, 2005.

[9] A. Vasile, F. R. Waugh, D. Greisokh, and R. M. Heinrichs. Automatic alignment of color imagery onto 3d laser radar data. In *AIPR*, 2006.

[10] P. Viola and W. Wells. Alignment by maximization of mutual information. *International Journal of Computer Vision*, 24(2):137–154, 1997.