Single Frame Based Video Geo-Localisation using Structure Projection

C. Bodensteiner
Fraunhofer IOSB
http://www.iosb.fraunhofer.de

S. Bullinger
IOSB

S. Lemaire
IOSB

M. Arens
IOSB

Abstract

Community image and video platforms like FlickR and Youtube offer large image collections from different perspectives. However, the majority of publicly available imagery from online communities lack a reasonable exact location and orientation information, which is important for many geo-spatial applications like object geo-referencing, knowledge transfer or augmented reality. In this work we exploit publicly available drone videos in order to bridge the gap between ground and aerial imagery. We propose a framework for the fast determination of full 6-D geo-referenced motion trajectories of online community drone video footage using geo-localized map data. Our method requires the registration of a single video frame from an video sequence in order to exactly geo-reference complete motion trajectories w.r.t. to existing geo-referenced map data. The method relies on SfM and SLAM techniques in combination with a simple, yet efficient appearance and structure matching based on rendered map data (e.g. LiDAR) in order to generate geo-registered 3D feature maps. These maps enable a simple and fast global appearance based geo-registration of visually overlapping community videos and images. We evaluate our method on a large set of community drone videos. Our method produces drift free geo-data overlays at an average speed of 29.7 *frames per second with an average positional error of* $0.4\mathrm{m}$*. In addition we release a large scale processed LiDAR dataset and geo-registered feature maps as an extension to the converging perspectives dataset. This data may provide visual links from ground based sensors to aerial imagery. Possible applications are numerous and include autonomous navigation, map updating/extension, image and video dehazing, object localisation or augmented reality.*

1. Introduction

Previously acquired maps allow for many new exciting applications. Many computer vision applications need the exact geometric transformation with respect to the map data in order to be useful. However, most geo-referenced imagery only provides more or less accurate positional information. This holds true for most of the data from the partially geo-tagged community photo- and video collections. This data usually comes with noisy geo-location information and completely lacks orientation information w.r.t. to existing geo-referenced map data. We propose a method for accurately geo-referencing community photo- and video collections leveraging large scale LiDAR imagery. LiDAR (Light Detection And Ranging) scanners are common sensor systems for the rapid acquisition of scene geometry. Recently large LiDAR data sets became publicly available. However, this data poses a challenge in order to be used with imagery from community image and video collections. There exist two main approaches for the registration of image data to 3D LiDAR data: rendering 2D appearance data (e.g. renderings) from 3D data and performing distance optimization in 2D or reconstructing 3D data from the images (SfM) and optimizing a distance measure in 3D. We propose a hybrid method in order to leverage advantages of both approaches: (a) appearance information is usually more distinctive as geometry and (b) 3D structure allows for an accurate registration by jointly using accumulated information of multiple image frames. We use the backscattered laser intensity information or textures from high resolution ortho

Figure 1. Point based rendering of a large scale LiDAR dataset textured with high resolution ortho-imagery. LiDAR datasets served as our reference model for the exact geo-referentiation of camera images from community video and image collections.

imagery in order to render views which enable an appearance based registration of a single image frame image w.r.t. to a geo-referenced LiDAR scan coordinate system. The registered image imposes structure constraints on a vision-based reconstruction of the input images and allows for augmenting the LiDAR data with image features from this data. This allows for a simple subsequent geo-referencing of visually overlapping vision data.

1.1. Previous Work and Contribution

We specifically focus on camera pose determination using heterogeneous data sets e.g. large scale LiDAR data. Appearance based 2D/3D registration techniques, e.g., image to map techniques for monocular vision systems are linked to location recognition systems. Most approaches employ local feature correspondence methods [**?**, **?**]. Schindler et. al. [**?**] proposed a location recognition method using vocabulary trees. Li and Snavely et. al. [**?**] proposed a prioritized feature matching scheme which exploits additional information from SfM. Zamir and Shah [**?**] published a 3DoF location recognition system based on 100,000 geo-tagged Street View images using direct location voting. With respect to 2D Image/3D LiDAR Registration there exists a considerable amount of literature. Vasile et al. [**?**] render pseudo-intensity images from LiDAR data to perform a 2D/3D registration with aerial images. Mastin et al. [**?**] use synthetically height color coded 2D renderings with camera images in combination with Mutual information (MI) [**?**]. Some approaches[**?**, **?**] rely on the detection and registration of geometrical features like line segments or planes in the camera image and their projections from 3D data.

To the best knowledge of the authors there is no literature about using community drone videos for geo-registration and localization of heterogeneous imagery. Published work also does not provide a simple method for using only a single registered view to exactly geo-register whole image sequences in combination with LiDAR data. We also leverage large-scale LiDAR data in combination with the drone videos in order to bridge the gap between airborne and ground based imagery.

A simple strategy would be the direct 3D/3D-registration of reconstructed geometry with given LiDAR data using recently proposed 3D features. However, we explored this approach using correspondence grouping techniques in combination with SHOT and FPFH feature descriptors and failed in establishing correct correspondences between SfM structure points and airborne LiDAR data. However, we explore this research direction in future work.

The contribution of the paper can be summarized as follows:

- First we extend the publicly available converging perspectives dataset with textured LiDAR data [1] in order to encourage research concerned with large geo-referenced 3D data. We also provide frame wise geo-registration information for publicly available drone imagery.
- We propose a method for accurately and efficiently geo-registering video streams to LiDAR data using only a single registered view of an image sequence. In this context we additionally propose a multi-rendering approach for the registration of such keyframes and robustly finding correspondences between challenging point based LiDAR renderings and electro-optical imagery.
- We point out important details about integrating LiDAR data into modern dense visual odometry and SLAM methods for enhanced robustness and the fast registration of geo-registered motion trajectories at video frame rates.

The outline of the paper is as follows: first we describe the involved data and key components of the system and discuss specific details for using LiDAR data as an additional information source. Then we describe our processing pipeline for video trajectory geo-referencing, e.g., view registration with heterogeneous data and structure alignment. The method is then evaluated with various motion sequences from online community collections. Finally, we discuss the results and point out further research directions.

2. Method

We focus on the creation and registration of large and spatially consistent geo-referenced feature maps from online community collections like FlickR and Youtube. Fig. **??** provides a short visual summary of our processing chain. The outline of the proposed approach can be summarized as follows:

After LiDAR data preprocessing we extract a small subset of keyframe images from each downloaded community drone video. Then we perform SfM on the keyframes to autocalibrate the cameras and to generate 3D structure points and corresponding appearance information from feature tracks (feature maps). Our data structure allows for a fast registration of 2D and 3D data using a standard feature matching approach for data association by determining geometric transformations based on the 2D (feature positions) and the 3D structure.

The proposed methodology can be decomposed into four building blocks:

- **LiDAR Map Data and Preprocessing:** First we preprocess the raw LiDAR point cloud data. This in-

[1] Project page http://s.fhg.de/georef

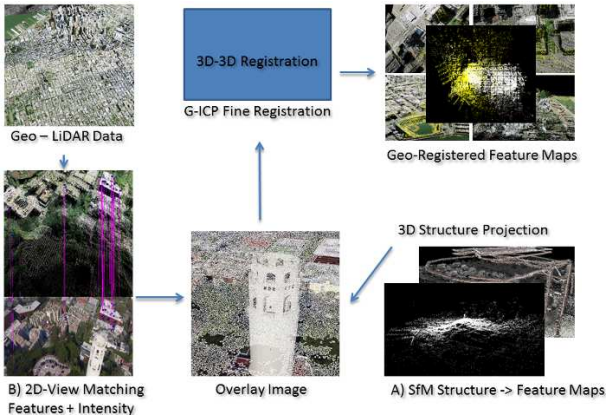


Figure 2. Visual summary of our 2D/3D processing chain

cludes filtering the pointclouds using a statistical outlier methods in order to remove erroneous structure. When available we additionally texture the pointcloud data with coincident ortho-imagery. This was done by sampling in the vertex colors using rendered Nadir views of the pointcloud. This preprocessing step substantially improves a visual correspondence search in some imaging scenarios.

- **(A) SfM based 3D Feature Map Creation:** We extract a small subset of keyframe images from each downloaded community drone video. We perform SfM[**?**] on the keyframes assuming a common camera calibration. This allows for a fast auto-calibration of the cameras. Then we generate 3D feature maps (3D structure points and the corresponding appearance information from the SfM feature tracks). These locally consistent feature maps usually cover only a small area and enable a robust and fast appearance based global registration of camera images and visually overlapping feature maps (Fig.**??**c).

- **(B) 2D/3D Video Image/LiDAR Registration:** We use multiple rendered LiDAR images for a single image frame and select putative inlier 2D-2D correspondences using geometric feature constraints on the jointly determined correspondences. We determine the depth of the rendered LiDAR image features using the GPU Depth Buffer in order to solve for a 2D/3D PnP problem[**?**, **?**] method. This initial registration is then refined by an intensity based 2D/3D approach. This registration allows for a pixel-wise overlay of a rendered LiDAR view with the input image. This overlay image is used for the structure projection stages (e.g. the 3D/3D feature map registration and for SLAM feature initialization and relocalisation).

- **(C) Structure Projection for 3D/3D Registration and SLAM Relocalisation:** In order to geo-reference the 3D featuremap we determine the 3D-transformation from the SfM structure points to the LiDAR geo-coordinate system (e.g. UTM). Based on the overlay images from (B) we generate a pixelwise depth image of the overlaid LiDAR dataset. This registered view is used to project the visible structure points onto the rendered map data. Depending on the reconstructed geometry and the map data we get hundreds of putative 3D/3D correspondences. Based on these correspondences we solve for a similarity transformation using RANSAC. Then we transform the 3D feature map into the geo-coordinate system using the similarity transformation. This structure projection approach is also used in our visual odometry pipeline. Here we use a similar approach for feature initialization and image relocalisation.

We briefly introduce the used mathematical notation. This paper considers 2D/3D camera pose estimation techniques for heterogeneous data, i.e., estimating the external camera $R, \mathbf{t}$ parameters when the internal camera parameters K are known. The projection of 3D world points $\mathbf{M_i}$ to corresponding 2D image points $\mathbf{m_i}$ are modeled by a standard pinhole camera projection P. The intrinsic parameters K_j with the parameters skew s, focal length f, aspect ratio α and principal point $\mathbf{u} = [u_0\ v_0]^T$ are assumed to be known (e.g. by using calibration patterns or SfM auto-calibration which assume shared parameters for drone videos, e.g. $K_j = K_k$).

$$\mathbf{m_i} = P_j \mathbf{M_i}, P_j = K_j \left[R|\mathbf{t}\right], K_j = \begin{bmatrix} f & s & u_0 \\ 0 & \alpha f & v_0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{1}$$

The following sections provide more details concerning each module of our geo-referencing processing chain.

2.1. LiDAR Map Data and Preprocessing

We use LiDAR scans from USGS covering the San Francisco Bay Area. We convert the data into our data format based on the open source PCL library. In order to handle the visualization of huge datasets we create downsampled versions of the data using octree methods. After removal of scanning artefacts using a statistical outlier filter we texture the LiDAR data using high resolution aerial ortho imagery. We sample the LiDAR vertex colors from the EO-map data using rendered nadir views of the LiDAR pointcloud with overlaid ortho-imagery.

2.2. (A) SfM based 3D Feature Map Creation

Recent progress in SfM methods led to very powerful algorithms which fully exploit the sparsity structure of

the underlying problem [**?**]. BA techniques [**?**] are commonly applied to simultaneously determine 3D scene structure and camera motion parameters (SfM) from image data by jointly minimizing the re-projection error of multiple image frames using non-linear least squares techniques. The classical approach is based on a simple measurement model $\mathbf{z} = f(\mathbf{p}, \mathbf{q})$, e.g., a pinhole projection model with $\mathbf{q}$ denoting the structure parameters (e.g., 3D points), $\hat{\mathbf{z}}$ the image observations ,e.g., the measured projections of the 3D points and $\mathbf{p}$ the extrinsic (optionally intrinsic) camera parameters. $v(\mathbf{p}, \mathbf{q}) = f(\mathbf{p}, \mathbf{q}) - \hat{\mathbf{z}}$ denotes the reprojection error. The Bundle Adjustment is then solved using non-linear least squares optimization techniques, e.g. Levenberg-Marquardt. We performed SfM reconstructions on downloaded community drone videos. We extract a small subset of the video frames with sufficient visual overlap and camera baseline distance. We use SIFT[**?**] and PBA[**?**] for the creation of the 3D feature maps. We enforce a common calibration to determine the intrinsic camera parameters of the drone videos and to lower perspective distortions of the reconstructed structure. We additionally filter spurious structure by keeping only descriptors from reasonably long feature tracks (e.g. feature track length ≥ 4). The robustness of SfM techniques strongly depends on textured scene geometry and motion trajectories. However, most of the downloaded drone videos led to good reconstructions due to their favorable motion trajectories and wide coverage of scene geometry.

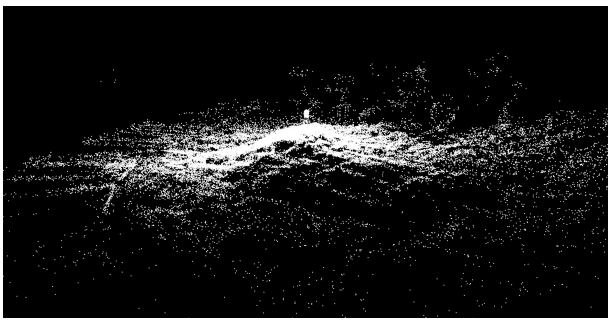


Figure 3. Feature map example (Coit Tower): For structure points (white dots) with sufficiently long feature tracks we associate a corresponding SIFT feature descriptor.

2.3. (B) 2D/3D Video Image/LiDAR Registration

The geo-registration of a query video frame image with LiDAR datasets based on a 2D approach requires correspondences between the query and rendered LiDAR images. We utilize a point based rendering approach in order to generate 2D views from the 3D LiDAR data set. Due to large appearance differences between rendered views and real camera images we jointly use multiple renderings in order to find a sufficient number of 2D correspondences (see Fig.**??**). We render these views using a standardized virtual movement pattern. The images are rendered using a star-like pattern (e.g. 6 views - sideways motion + motion in the direction of the optical axis). Based on these images we extract local image features using standard descriptors (e.g. Surf [**?**]) and detectors with low contrast thresholds. The 3D-coordinates of the feature positions are determined using the GPU-depth buffer at the feature detector positions. The feature descriptors $L = \{d_1, d_2, ..., d_m\}$ are pooled in a putative correspondence feature data set. We use KD-Trees for fast nearest neighbor feature association. We establish 2D/2D point correspondences based on this feature set. Correct correspondences are robustly identified by searching for small regions with similar geometric relationship of local features. The registration is then carried out using an PnP algorithm. Afterward we minimize the re-projection error (**??**) using Levenberg-Marquardt.

$$\text{minimize}_{R,\mathbf{t}} \sum_i \|K(R\mathbf{M_i} + \mathbf{t}) - \mathbf{m_i}\|_2^2 . \qquad (2)$$

Fig.**??** visualizes determined inlier-correspondences between a LiDAR rendering and a video frame from the drone videos. The calculated pose is finally refined with an intensity based approach. A standard approach is to render pose parametrized 2D views $V_{ren}(R, \mathbf{t})$ from the 3D dataset which minimize/maximize an intensity based distance/similarity measure $D_{(Typ)}, D : \mathbb{R}^N \times \mathbb{R}^N \rightarrow \mathbb{R}$ between the reference image I_R and a rendered view over the support of the image region A.

$$\text{minimize}_{R,\mathbf{t}} \int_A D_{(Typ)}(V_{ren}(R, \mathbf{t}), I_R). \qquad (3)$$

We maximized an intensity based similarity measure between rendered views and the query images. Intensity based similarity optimization allows for subpixel accurate alignments but is computationally expensive. However, these methods rely on good initializations to prevent local optima. The convergence range of intensity methods is usually small. We use a multi-scale approach (3 octaves) to extend the convergence range. The local feature based pose usually provides a sufficiently close starting point. The selection of an appropriate distance measure is very important. Mutual Information [**?**] is still considered the gold standard matching approach for heterogeneous data. It measures the mutual dependence of the underlying intensity distributions:

$$D_{(MI)}(I_R, I_{T_\theta}) = H(I_R) + H(I_{T_\theta}) - H(I_R, I_{T_\theta}) \quad (4)$$

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

$$H(I_R, I_{T_\theta}) = \sum_{X \in I_{T_\theta}} \sum_{Y \in I_R} p(X, Y) log(\frac{p(X, Y)}{p(X)p(Y)}) \quad (5)$$

is the joint entropy. $p(X,Y)$ denotes the joint probability distribution of image intensities X, Y in I_R and I_{T_θ}, and $p(X)$ and $p(Y)$ denote the marginal probability distributions. However, MI usually exhibits many local minima. For enhanced accuracy and robustness we combine it with gradient correlation [**?**]. Intensity based registration is computationally intense (approx. 20s). However, the optimization usually starts near to an optimum and needs to be performed only once for a image sequence. As transformation representation for the extrinsic camera parameters R and t we employ the minimal encoding of rigid body transformations SE(3)

$$T = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}, \ \texttt{with} \ R \in SO(3), t \in \mathbb{R}^3, \quad (6)$$

based on the Lie algebra representation. Here, $SO(3)$ denotes the Lie group of rotations with the matrix multiplication as the group operator. In case of $\mathbb{R}^3$ the Lie algebra $\mathfrak{se}(3)$ leads to a minimal representation as 6-vectors $(\omega, \nu)^T$, where ω is the axis angle representation of the rotation R and ν the translation vector. We use a downhill simplex algorithm for the optimization, which works without providing analytical derivatives. This allows a fast adaptation of new intricate distance measures.

2.4. (C) Structure Projection for 3D/3D Registration and SLAM Relocalisation:

The main advantage of using a 3D/3D structure registration based on overlay images (Fig.**??**) is the robustness of the approach regarding a high uncertainty in camera localization. Camera images with high focal distances usually exhibit high positional uncertainty in the direction of the optical axis. This is due to the high correlation of translation and focal distance camera parameters w.r.t. image observations. However, our approach determines only putative correspondences along the projections of visible structure points and their intersection with the LiDAR data. After projection of the feature locations we get a sufficiently large set of 3D/3D correspondences for the robust estimation of a transformation of the structure points. Based on the correspondences we (RANSAC) estimate the transformation connecting both coordinate systems. We then optimize the transformation over elements of the Lie group of similarity transforms SIM(3).

$$S = \begin{bmatrix} sR & t \\ 0 & 1 \end{bmatrix}. \quad (7)$$

The exponential map $exp_{SIM(3)}$ and the inverse mapping $log_{SIM(3)}$ are analogously defined w.r.t. $SE(3)$.

$$exp_{SIM(3)} \begin{pmatrix} \sigma \\ \omega \\ \nu \end{pmatrix} = \begin{bmatrix} e^\sigma exp_{SO(3)}(\omega) & W\nu \\ 0 & 1 \end{bmatrix} = S. \quad (8)$$

with W analogous to the known formula of Rodriguez.

$$W = e^\sigma (I + \frac{1 - cos(\theta)}{\theta^2}(\omega)_x + \frac{\theta - sin(\theta)}{\theta^3}(\omega)_x^2) \quad (9)$$

The transformed feature map structure points are then further aligned using an ICP[**?**] method employing an PCL library implementation. Fig.**??** shows a visualization of the geo-registered structure points of a feature map (depicted as yellow points) in combination with the underlying 3D LiDAR data. While the initial registrations (anchor feature maps) need a supervised initialization, subsequent images and feature maps can be automatically registered - e.g. white point cloud Fig.**??**(c).

The approach of using overlay images is also exploited in our monocular VO system. First we localize query images using the descriptors in the feature map in order to create the overlay image (resp. to determine a relocalisation transformation). We then generate a sparse depth image. However, due to the LiDAR data resolution and point cloud structure not all depth values are valid. Therefore we filter the depth image and remove invalid depth values (e.g. far clipping plane depth buffer values). The depth image can then be used to initialize features or image regions using the structure projection approach. This is especially useful in case of recently proposed dense mapping approaches [**?**].

3. Experiments and Results

We evaluate our geo-referencing method on a large set of community drone videos. To this end we downloaded over 115 drone videos and performed the outlined 3D feature map creation. Except for 17 videos we managed to create locally consistent 3-D feature maps covering large parts of San Francisco (e.g. Fig.**??**f). We downloaded image collections (ca. 120000 images) from FlickR and Panoramio around popular landmarks (Coit Tower, Alamo Park, Golden Gate Bridge, Palace of Fine Arts, Golden Gate Park, Embarcadero, AT&T Stadion) using a Phyton programming interface.

3.1. Results

We performed the 3D feature map creation for reasonable sized subsets (2000-5000 images) of the unstructured image collections. However, our feature map pipeline is not yet optimized for large unstructured image collections. We managed to create and register small ground based feature maps for the Palace of Fine Arts (341/4950 registered/download images) and the Coit Tower (335/1498 images) sites. A combined ground/airborne registration of the Palace of Fine Arts dataset is shown in Fig.**??**b-c).

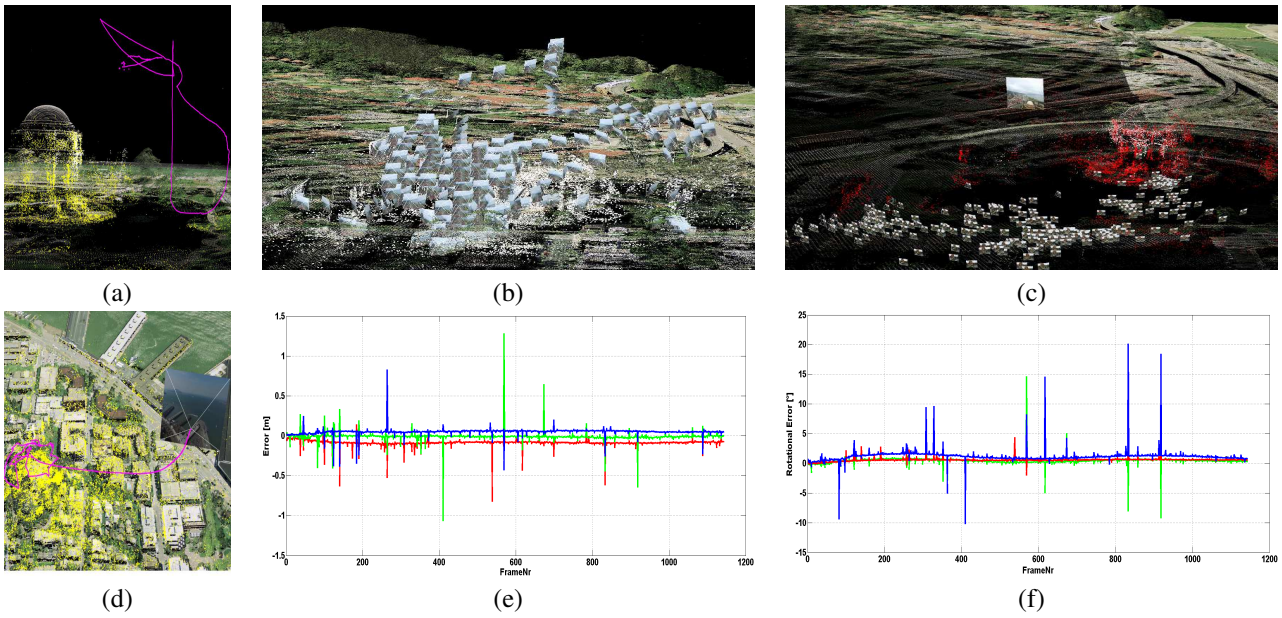


Figure 5. (a) Geo-localised SLAM trajectory and 3D feature map (yellow points) of the palace of fine arts (a-c). (b) Bundle adjusted camera positions of the drone video feature map (white points). (c) geo-localised drone frame position and camera positions of a community photo collection (FlickR) feature map. (d) Geo-localised SLAM trajectory (56 000 frames) and 3D feature map (yellow points) of the coit tower area. Position (e) and orientation (f) camera trajectory distance (x,y,z) of the geo-referenced visual odometry module and a framewise feature based geo-referenced trajectory (1200 frames).

3.1.1 LiDAR/Image Correspondences

We registered multiple video sequences to a LiDAR geo-coordinate system and carefully checked the registration results by visual inspection (see Fig.**??**). We determined the intrinsic parameters of the cameras using auto calibration. We determined the ground truth projection matrices (extrinsic and intrinsic parameters) for multiple camera images. We used a similar evaluation procedure as described in [**?**] based on ROC curves by varying the matching threshold (NN-distance ratio). However, we normalize the values based on the cardinality of the ground truth correspondences. This enables a better comparison of different approaches w.r.t. standard parameters and low inlier rates. The determination of True/False-Positives/Negatives was based on the re-projection error of the projected 3D and 2D local feature coordinates. In this way we determined correct correspondences for recall/precision curves (Fig.**??**). The observed correspondence inlier rate ranged from 3% to 7% depending on the evaluated images. Upright descriptors (SURF) worked best in case of heterogeneous image pairs. Rotational invariant feature descriptors showed significantly lower inlier rates, since invariant descriptors come usually at the price of reduced discriminance.

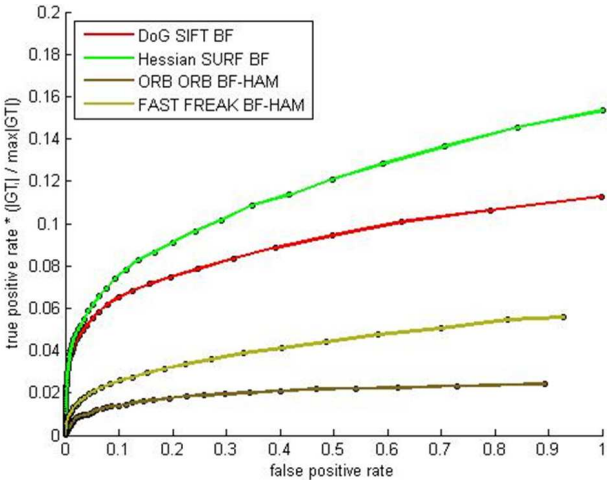


Figure 4. Normalized ROC curve for SIFT, SURF, FREAK and ORB descriptors - LiDAR and optical image correspondences.

3.1.2 Video Trajectory Geo-Referentiation

To evaluate the accuracy of the trajectory optimization module we measured the distances of bundle adjusted camera motion sequences (BA) compared to the motion trajectories of our geo-referenced visual odometry module. We reconstructed the motion and structure of 5 sequences (BA over 1200 video frames) which served as ground truth data.

We geo-referenced the SfM data according to the described method and carefully checked the registration. We initialized our VO pipeline by the registration of the first view and the outlined structure projection. All subsequent frames are determined by the VO pipeline. We calculate the average distance of the camera centers in the metric geo-referenced coordinate system of the LiDAR scans (UTM). W.r.t. the relocalisation with geo-referenced structure our method produced drift free geo-data overlays at an average speed of 29.7 frames per second with an average positional error of $0.4m$.

3.2. Runtime Experiments

The implementation (C,C++) is based on open source software libraries (OpenCV,PCL,PBA) from the computer vision community. Our processing chain is not runtime-optimized. The measurements provide a rough performance estimate. Out test system is equipped with a Core i7-980X with an NVIDIA Titan X GPU with 12GB video memory. The 3D/3D registration takes around 1s while the feature based multi-rendering 2D/3D registration ranges between 15-30s due to the large number of putative 2D/2D correspondences. Camera trajectory geo-referencing (VO pipeline) takes around 34ms per frame. While the visual odometry module is faster, relocalisation with the 3D feature map requires SIFT feature extraction, matching and rendering the depth map overlay from the point cloud data.

4. Conclusion and Future Work

We proposed and implemented a video and image geo-referencing processing chain for community photo and video collections based on LiDAR data. We leverage community drone videos in order to bridge the gap between ground and airborne imagery. The accurate registration of heterogeneous data (e.g. LiDAR and video data) is still a difficult vision problem due to strongly differing object appearances. We augment LiDAR data with appearance information (3D feature maps) from community video and image collections in order to simplify subsequent registrations. Given no additional information the system needs only a minimum amount of supervision for the initial registration of a few anchor feature maps. After achieving a sufficient visual coverage of the area the approach allows for the fast automatic 6D geo-referencing of imagery from online community collections. However, many properties of our appearance based geo-referencing system still remain open. We focus our work on the integration of metric learning and descriptor optimization methods in order to scale our system to very large datasets by jointly leveraging image to image and image to map registration techniques.

References

[1] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110:346–359, 2008.

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

[3] J. Engel, T. Schoeps, and D. Cremers. Lsd-slam: Large-scale direct monocular slam. In *ECCV*, 2014.

[4] C. Frueh, R. Sammon, and A. Zakhor. Automated texture mapping of 3d city models with oblique aerial imagery. In *Proc. 2nd Int. Symp. 3D Data Processing, Visualization and Transmission 3DPVT 2004*, pages 396–403, 2004.

[5] J. A. Hesch and S. I. Roumeliotis. A direct least-squares (dls) solution for PnP. In *Proc. of the Int. Conf. on Computer Vision*, Barcelona, Spain, Nov. 6–13, 2011.

[6] K. Konolige. Sparse sparse bundle adjustment. In *Proc. BMVC*, 2010.

[7] 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.

[8] Y. Li, N. Snavely, and D. P. Huttenlocher. Location recognition using prioritized feature matching. In *ECCV*, 2010.

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

[10] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[11] G. Penney, J. Weese, J. A. Little, P. Desmedt, D. L. Hill, and D. J. Hawkes. A comparison of similarity measures for use in 2-d-3-d medical image registration. *IEEE Transactions on Medical Imaging*, 17(4):586–595, 1998.

[12] G. Schindler, M. Brown, and R. Szeliski. City-scale location recognition. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition CVPR '07*, pages 1–7, 2007.

[13] A. Segal, D. Hhnel, and S. Thrun. Generalized-icp. In *Robotics: Science and Systems'09*, pages –1–1, 2009.

[14] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. *Bundle Adjustment – A Modern Synthesis*. Springer-Verlag, 2000.

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

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

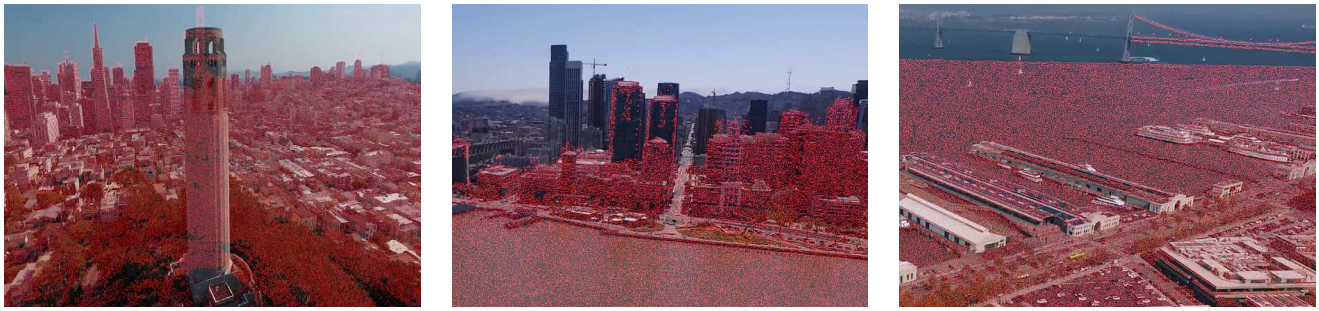
[17] C. Wu. SiftGPU: A GPU implementation of scale invariant feature transform (SIFT). Technical report, University of North Carolina at Chapel Hill, 2007.

[18] C. Wu, S. Agarwal, B. Curless, and S. Seitz. Multicore bundle adjustment. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 3057–3064, 2011.

[19] A. R. Zamir and M. Shah. Accurate image localization based on google maps street view. In *ECCV*, 2010.

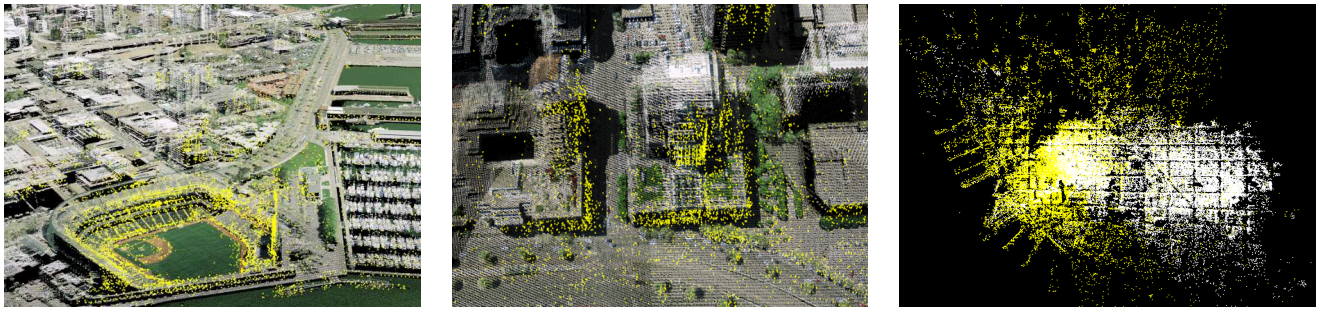
(a) (b) (c)

Figure 6. Visualization of determined inlier-correspondences (a-c) between the 3D LidAR model and a single video frame from community drone videos of San Francisco. The Multi-Rendering approach enables the generation of precise overlay images (point cloud renderings) w.r.t. to the video frame by using either raw LiDAR intensity images (a) or textured LiDAR data renderings (b,c).



(a) (b) (c)

Figure 7. Drone camera images overlaid with edge images extracted from LiDAR intensity images generated with the same intrinsic and extrinsic parameters. The registered overlay images are then used to project the structure and feature descriptors from the SfM reconstructions of the drone videos onto the 3D LiDAR dataset.



(a) (b) (c)

Figure 8. Visualization of the geo-registered structure and feature maps (depicted as yellow points) with the underlying 3D LiDAR model. While the initial registrations (anchor feature maps) need a supervised initialization, subsequent images and feature maps can be automatically registered - e.g. white point cloud - (c).