Video Registration to SfM Models

Till Kroeger[1] and Luc Van Gool[1,2]

[1] Computer Vision Laboratory, ETH Zurich, Switzerland
[2] ESAT - PSI / IBBT, K.U. Leuven, Belgium
{kroegert,Bvangool}@vision.ee.ethz.ch

Abstract. Registering image data to Structure from Motion (SfM) point clouds is widely used to find precise camera location and orientation with respect to a world model. In case of videos one constraint has previously been unexploited: temporal smoothness. Without temporal smoothness the magnitude of the pose error in each frame of a video will often dominate the magnitude of frame-to-frame pose change. This hinders application of methods requiring stable poses estimates (e.g. tracking, augmented reality). We incorporate temporal constraints into the image-based registration setting and solve the problem by pose regularization with model fitting and smoothing methods. This leads to accurate, gap-free and smooth poses for all frames. We evaluate different methods on challenging synthetic and real street-view SfM data for varying scenarios of motion speed, outlier contamination, pose estimation failures and 2D-3D correspondence noise. For all test cases a 2 to 60-fold reduction in root mean squared (RMS) positional error is observed, depending on pose estimation difficulty. For varying scenarios, different methods perform best. We give guidance which methods should be preferred depending on circumstances and requirements.

1 Introduction

Due to recent advances in 3D range imaging highly accurate and large 3D models for real-world environments can easily be obtained [1,20,38] and are already available for many city areas. Given structural information about the world, many new opportunities for computer vision (CV) applications in scene understanding arise. Videos are a rich source for capturing and analyzing social activities, human/vehicular traffic and events. This allows for CV applications such as multi-view object tracking, vehicle and pedestrian trajectory analysis, video cutting, multi-video event and scene summarization. Registration of video data to a 3D world model using visual information is an essential requirement for many of these applications. They benefit from accuracy and robustness of pose estimations (6-DoF, position and orientation) for one or several videos at all frames, rather than live performance and efficiency. As processing for these higher level CV applications happens mostly offline (or in batches) global reasoning is sufficient and preferable over live or incremental pose tracking. Localization using visual information only has the advantage that only visual sensors are required and will work even in GPS-denied environments.

D. Fleet et al. (Eds.): ECCV 2014, Part V, LNCS 8693, pp. 1–16, 2014.

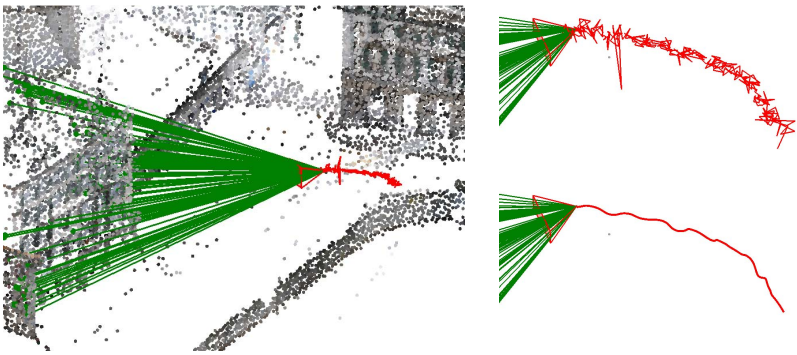

Fig. 1. Left & Top right: frame-wise registered hand-held video with 300 frames, The camera's path (red) is noisy due to PnP estimation errors: The average frame-to-frame position difference is 57cm while the ground truth camera moves with approximately 5cm/frame. Bottom right: refined camera path (with Kernel Regression).

The standard approach of image-based registration to SfM models ([34,35,16,26,25]) involves the following steps for each image/video frame: computing 2D image features, matching them to features associated with 3D points, finding the pose using a standard perspective-n-point (PnP) algorithm in a RANSAC loop with 2D-3D correspondences. Direct application of this technique to video data results in very noisy pose estimates, as illustrated in Fig. 1 (left & top right). The path of the camera is drawn in red and exhibits strong positional noise: the average positional difference between poses of successive frames is one order of magnitude larger than the ground truth motion. The true motion is completely *dominated by uncorrelated positional errors*. Using frame-wise PnP estimates we also have to deal with estimation failures (i.e. gaps) and pose outliers in addition to noisy poses. This is unsatisfactory since many tasks, such as multi-view tracking or augmented reality, require *accurate*, *gap-free* and *smooth* poses as input. This is why we explore several regression methods to exploit temporal smoothness for refining PnP camera poses, which were independently estimated for every video frame. Our aim is to bridge the gap between unreliable, noisy, incomplete, frame-wise pose estimates in SfM models to accurate and smooth pose trajectories to be used for higher-level CV applications.

Our main contribution is the reduction of pose errors for all frames of a video, for which approximate and possibly incomplete frame-wise estimated poses are available. In order to achieve this, we adapt several model fitting techniques (Splines Smoothing, Kernel Regression, Non-Linear Least-Squares optimization) to the problem. We propose a new pose parametrization to be able to use spline and kernel smoothing methods for camera poses. In Non-Linear Least-Squares optimization we introduce a novel bending energy minimization extension for camera pose smoothing. We discuss several combinations of the three methods. All methods are evaluated on real and synthetic data for various difficult

scenarios. We give guidance on which method works best under which circumstances. Until now, no such comprehensive description and evaluation for global pose trajectory refinement exists. We are the first to contribute a carefully designed benchmark on synthetic and real data for this.

Paper Overview: Sec. 2, lists related work. The video pose registration methods (Spline Smoothing, **SP**, Kernel Regression **KR**, non-lin. least-squares optimization **LS**) and variants are proposed in Sec. 3, and evaluated in Sec. 4. Sec. 5 concludes the paper.

2 Related Work

Landmark recognition, localization of images are active fields of research: An image is to be positioned with respect to reference images with known localization (e.g. GPS) [40,17], at city scale, with efficient feature representation [36,21], using databases of building facades [32], and even larger world-wide approximate localization [14]. Other methods rely on localization by recognition of landmarks, e.g. [4]. These methods do not rely on 3D structure, but on localized reference images and 2D features. Generally, most of these techniques employ image retrieval techniques and 2D similarity measures based on feature matching. In contrast to this [11] relies on 3D features to recognize places.

Image-based registration to 3D SfM models is concerned with complete (6-DoF) pose estimation. Poses are computed with 2D-3D correspondences based on feature matching [37]. Poses retrieved in this way are more accurate and dependency on visually similar reference images is reduced. Scaling these techniques is difficult, due to the large amounts of features in the matching step. Perspective-n-Point (PnP) algorithms find the pose given a set of 2D-3D correspondences [43,42,23,24]. [25] improves the feature matching step by RANSAC co-occurrence-based sampling. Focus has been put on efficient feature storage and matching [5], with vocabulary trees [16], prioritized matching [26], efficient correspondence search [35,34], match pre-filtering using image retrieval [7], and discriminative visual element mining in challenging scenarios, such as registering paintings [3]. [12,9] address the problem of finding the (6-DoF) pose of observed objects. [31] estimates the pose using lines instead of point features.

Localization of image sequences and videos has received attention as well. [10] registers video frames by employing fundamental matrix constraints between a video frame and the two closest GPS-annotated reference images. GPS coordinates are extracted for all frames by Spline Smoothing. Similarly, [39] retrieves geolocalized images and uses Bayesian tracking for refinement. [18] coarsely localizes image sequences with large time-gaps, such as series of photographs of entire tourist trips on a world-wide grid. Visual odometry in car-mounted cameras is used for localization in a known road network [6]. [2] optimizes poses when no model but frame-to-frame pose changes and measurement uncertainties are available from essential matrices or inertial sensors.

Video registration to 3D SfM models received less attention than image-based registration. However, it is an integral part of SLAM [8,19,30,29]. There,

the focus lies on jointly tracking features, and improving their and the camera's localization. In contrast to this the scene structure is predetermined in our task. We do not have a prior on the camera's location. Additionally, the reconstructed environments in SLAM are usually small controlled indoor environments. Imaging conditions for SfM and localization are the same in SLAM, which is generally not the case when a query video has to be registered to separately reconstructed 3D models. [37] localizes video sequences by matching and tracking SIFT features. Similarly, [27] estimates poses by matching and tracking DAISY features. Registration to high-quality CAD models has been worked on as well: ego-motion is tracked in [22] by edge matching in omnidirectional videos, in [15] by feature tracking and coarse-to-fine refinement of edge alignment. [16] finds poses for every frame of videos separately, simplifying the matching by computing virtual views. [41,33] rely on computing SfM from a query video first, and retrieve poses by alignment of the world model and the SfM model from the query video.

SLAM and feature-tracking based techniques work for small datasets or when features can be matched reliably. If matching is difficult (larger city scenes, strongly varying imaging conditions), tracking features will easily result in propagation of matching errors. Techniques that reconstruct the sequence first and match later suffer from typical SfM problems: model deformation and fragmentation, matching problems and the need for manual subsequent alignment with a world model. Because of these principled problems we want to match as many frames as possible directly to the world model and rely on global pose refinement.

3 Registration of Videos to SfM Models

Frame-wise registered videos can exhibit strong noise in individual poses, estimation gaps and pose outliers as illustrated in Fig. 1. Noise, outliers and estimation gaps can be dealt with when incorporating temporal smoothness. On approximately and incompletely registered poses for each frame, described in Sec. 3.1, we build the refinement methods proposed in Sec. 3.2, 3.3, and 3.4. The goal is to improve every frame's pose estimate while being robust towards outliers. We chose Spline Smoothing as a well-known representatives of regularization and basis expansion techniques, Kernel Regression as a representative for probabilistic kernel methods and Non-Linear Least-Squares optimization as representative for direct optimization of re-projection errors as also used in bundle adjustment.

3.1 Image-Based Registration

A SfM model is represented by 3D points and associated SIFT [28] feature descriptors from the views in which the 3D points were observed. We match a new query image by extracting SIFT features, matching them to all features associated with the 3D points, and thereby retrieve a putative set of 2D to 3D correspondences. For known internal parameters many recent pose estimation algorithms (EPnP[23], ASPnP[43], OPnP[42], RPnP[24]) can be used

directly in a RANSAC-loop to retrieve (6-DoF) camera position and orientation [34,35,16,26,25]. In the remainder of the paper we assume given internal camera parameters (focal length, projection center, no radial distortion).

3.2 Spline Smoothing

In Spline Smoothing (**SP**) piece-wise polynomial functions $f(x_i)$ are fitted to N sites x_i with observations y_i by minimizing the residual sum of squares (RSS):

$$RSS(f,\lambda) = \lambda \sum_{i=1}^{N} w_i \left(y_i - f(x_i)\right)^2 + (1-\lambda) \int (f''(x))^2 dx\,. \tag{1}$$

The camera pose estimate at time x_i is denoted with y_i (observed), and $f(x_i)$ (smoothed). The N data sites correspond to the number of video frames. A camera pose is represented as a position t and rotation matrix R. We parametrize the pose as a 9-dimensional vector $y = [t^T\ r_1^T\ r_2^T]$ with unit vectors r_1 and r_2 as viewing direction and up-vector of the camera. The RSS is regularized by f's second derivative, i.e. to minimizing the bending energy. The data fidelity term is weighted by w_i, the inlier count after RANSAC, down-weighting poses with few 2D-3D correspondences. The regularization parameter $\lambda \in [0,1]$ is found via leave-one-out cross-validation.

We propose a variant of a smoothing spline including the camera parameters' covariance Σ and the Mahalanobis distance in the data fidelity term (**SP+C**). Deviation from estimated poses are penalized stronger in the data fidelity term if the cameras' pose estimates are with low variance:

$$RSS(f,\lambda) = \lambda \sum_{i=1}^{N} w_i \left(y_i - f(x_i)\right)^T \Sigma^{-1} \left(y_i - f(x_i)\right) + (1-\lambda) \int (f''(x))^2 dx\,. \tag{2}$$

The solution for both spline variants is a weighted linear combination of the observations. The chosen pose parametrization is an approximation to rigid Euclidean motion which is part of the Special Euclidean Lie Group SE(3). Some constraints on orientation cannot be enforced by the spline formulation: orthogonality ($r_1 \cdot r_2 = 0$) and unit norm ($\|r_1\| = \|r_2\| = 1$). However, if the change in R is small, we can assume $\|r_1\| \approx 1 \approx \|r_2\|$ and $r_1 \cdot r_2 \approx 0$. This allows using this under-constrained approximation in smoothing. We can enforce constraints afterwards: For each $f(x_i)$ we re-normalize r_1, r_2 to unit norm and recover $R' = [r_3\ r_1\ r_2]^T$ with $r_3 = r_1 \times r_2$. To get a valid rotation matrix we enforce orthogonality by singular value decomposition $[U,S,V] = \mathrm{svd}(R')$ and set $R' = U \cdot V^T$. This approximation is valid as long as the between-frame change in R is slow. See experiments in Sec. 4.1, 4.4 for an analysis of the limits of this parametrization. As alternative parametrization, we experimented with quaternions and an angle-axis representation, with less stable results.

3.3 Kernel Regression

A smoothing spline works well in cases of outlier-free data, perturbed by Gaussian noise. However, even after RANSAC a few pose outliers can remain. In order

to avoid a hard inlier-outlier decision for poses, we can still use a RANSAC-inspired pose estimation approach. But instead of keeping only the best result (i.e. sample with highest inlier count) we keep the M best pose samples. This leads to M pose estimates for all N frames. Using the *best* RANSAC samples requires randomly distributed outliers. If outliers are systematic, M *random* samples have to be used to avoid biased estimates. A Nadaraya-Watson model, or Kernel Regression (**KR**), can represent poses over N data sites probabilistically:

$$p(y,x) = \frac{1}{W} \sum_{i=1}^{N} \sum_{j=1}^{M} w_{i,j}\ k(x - x_{i,j}, y - y_{i,j}) \tag{3}$$

where k is the density function, $w_{i,j}$ sample inlier count, $W = \sum_{i=1}^{N} \sum_{j=1}^{M} w_{i,j}$. The pose sample j at time x_i is denoted $y_{i,j,}$. We use a Gaussian kernel

$$k(x,y) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x^2}{h_x} + \frac{y^2}{h_y}\right)\right) \tag{4}$$

with bandwidths h_y in parametric space for the camera pose and h_x in time. Both parameters are found in leave-one-out cross-validation. The regression function $f(x_i)$ corresponds to conditional averages of target y_i conditioned on time x_i:

$$f(x_i) = \mathbb{E}(y|x_i) = \int y\ p(y,x_i) dy\,. \tag{5}$$

This representation allows for a non-parametric probabilistic interpretation of the camera's pose at all times. Outliers can be filtered out effectively. However, it depends on integration over all kernel functions, which can be time consuming.

3.4 Non-linear Least-Squares Optimization

SP and KR (Sec. 3.2, 3.3) operate directly on estimated poses, and are therefore dependent on initial PnP pose quality. We propose a similar objective function as eq. (1) with data fidelity and smoothing term, but instead of using estimated poses in the data fidelity term, we can use the 2D-3D correspondences directly by measuring the 3D point re-projection error. The objective function can be minimized as a Non-Linear Least-Squares (**LS**) problem:

$$RSS(P_1, \ldots, P_n) = \sum_{i=1}^{N} \sum_{j=1}^{J_i} (z_{i,j} - P_i\, Z_{i,j})^2 + \lambda\, T^T K T \tag{6}$$

where $P_i = C \cdot [R_i\ ,\ -R_i \cdot t_i]$, with C known camera calibration, J_i the number of 2D-3D correspondences after RANSAC for pose P_i, and $z_{i,j}$, $Z_{i,j}$ the known 2D and 3D locations of correspondence j in pose P_i. K is the bending penalty matrix as in the Reinsch–form for SP[13, p.154]. $T = [t_1, \ldots, t_n]$ is a matrix of camera locations. PnP poses are only required as initialization, reducing dependency on PnP methods. The additional advantage of this approach is that other constraints, such as planarity of camera movement (**LS+CP**), can be integrated:

$$RSS(P_1, \ldots, P_n, CP) = \sum_{i=1}^{N} \sum_{j=1}^{J_i} (z_{i,j} - P_i\, Z_{i,j})^2 + \lambda\, T^T K T + \theta \sum_{i=1}^{N} \mathrm{D}^2(CP, t_n) \tag{7}$$

where D (3rd term) returns the distance of camera position t_n to camera plane CP. The camera plane is a free variable in the optimization. Because the regularization parameters λ and θ cannot easily be found automatically, we set them manually to balance the influence of all residuals. Starting the optimization with PnP poses and associated 2D-3D correspondences, we cannot easily remove the influence of incorrect 2D-3D correspondences. However, we can use a Cauchy loss for the re-projection error $\Delta(x) = \log(1 + x)$ to mitigate the influence from outlying correspondences. The Ceres-Solver[1] is used to minimize eq. (6, 7).

3.5 Combinations and Variants

Besides the discussed methods SP, KR, LS and variants SP+C, LS+CP we include several combinations in our experiments when suitable. The estimated poses from PnP algorithms can be further refined by using Non-Linear Least-Squares optimization of the re-projection error, i.e. eq. (6) without smoothing ($\lambda = 0$) (**LSWS**). Based on LSWS we can again start Spline Smoothing (**LSWS+SP**) or Kernel Regression (**LSWS+KR**). When LS is started from PnP pose estimates, outliers are corrected due to the smoothing term. This correction is improved when LS is initialized from Spline Smoothing solutions (**SP+LS**), and may also be combined with the assumption of planarity of camera movement (**SP+LS+CP**). For some frames no pose estimates exist. This is due to RANSAC failures because of too many outliers or noisy correspondences. Gaps in PnP pose estimates can also be deliberate: Feature matching and RANSAC pose estimation is the main bottleneck for large SfM scenes. It may be necessary to consider only every nth camera pose and interpolate in between. Such gaps can be closed after pose refinement by using standard cubic interpolating splines with knots given by the output of our proposed methods.

4 Experiments

In three experiments we evaluate the performance of the proposed methods and variants, while assuming that unreliable PnP poses from an arbitrary source are available as input. We evaluate with exemplary state-of-the-art PnP methods as mentioned in Sec. 3.1. The first experiment (Sec. 4.1) on synthetic data shows the stability of the smoothing methods with respect to different degrees of pose changes, 2D observation noise, number of 2D-3D correspondences and outlier contamination. The second experiment (Sec 4.2,4.3) shows the performance of the methods on real SfM data of city environments. In the first two experiments

[1] `http://code.google.com/p/ceres-solver/`

we ignore any occurring gap in the PnP pose estimates. In the third experiment (Sec. 4.4) we compare the methods when interpolating over gaps following the smoothing. We focus on the positional (root mean squared, RMS) error to ground truth camera locations. The reasons are 1) space constraints, 2) the camera position is more sensitive to typical problems in feature-based pose estimation (noisy/incorrect 2D-3D correspondences) than orientation and 3) position and orientation errors are strongly correlated. See supplementary material for full results. We consider frame-wise PnP pose estimates, computed with ASPnP [43], as baseline. We chose ASPnP as best performing PnP method (See table 3 in Sec. 4.2). Results for further refined poses (LSWS) without smoothing, and simple Kalman Filtering (KF) are also included. The experimental setup remains the same in all experiments: regularization parameters for SP and KR are automatically determined via cross-validation. Regularization parameters for LS are set manually and are the same for all experiments: $\lambda = \theta = 10^5$. We scale the 3D model to real-world scale. All variants of LS run for 10 iterations. State and observation covariances for KF are computed in an EM-style algorithm. To limit memory-complexity in KR, we set $M = 20$ pose samples per frame.

4.1 Synthetic Video Sequence

The synthetic data consists of a camera (focal length 1000 px, 1280x720 resolution), viewing a simple 3D structure (2 walls at a 135 angle) with 800 3D-points at a distance of 10 meters. We create sequences of 300 frames by rotating the camera around the visible structure (Fig. 2, top left). For every frame we randomly sample 25 2D-3D correspondences and compute the pose. In different sequences with increasing speed of rotation (degrees/frame) we test the effect on smoothing and stability of the pose parametrization. All experiments are repeated 25 times and results averaged. Fig. 2 shows the positional error of our proposed methods against the ASPnP baseline. Increasing speed of rotation around the structure, shown in Fig. 2 (a), enlarges pose differences between successive frames. This shows how each method is affected by increasing pose differences and the sensitivity of the pose parametrization for SP and KR, outlined in Sec. 3.2. In Fig. 2 (b) we examine the performance if the number of 2D-3D correspondences before computing the PnP pose in each frame is decreased, (c) Gaussian noise is added to the 2D feature locations, and (d) the percentage of (uniformly distributed) pose outliers is increased. These plots show the reliability of each method with respect to typical challenges in feature-based pose estimates. We observe:

- KR and SP perform well for slow pose changes. The under-constrained pose parametrization leads to a rising performance loss for fast pose changes.
- LSWS and LS are unreliable (out of scope in plots) due to strong 2D, 3D noise, leading to local optima in optimization. SP+C is unreliable as well.
- The overall best performing and stable method is LS optimization initialized with the result of Spline Smoothing (SP+LS): The local optima problem of LS and LSWS are avoided by initializing the poses near the real optima. SP+LS is hardly affected by low feature count, noise, outliers and fast pose changes.

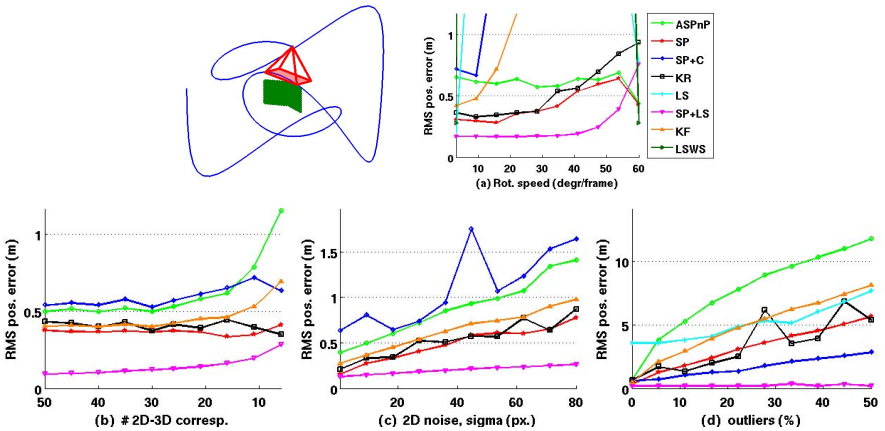

Fig. 2. Refinement results for synthetic video. Top Left: Synthetic sequence (300 frames) of camera (red) rotating around structure (green). Refinement result for (a) varying speed of movement around the structure (degrees/frame), (b) number of 2D-3D correspondences, (c) 2D Gaussian noise, (d) contamination with PnP pose outliers. The legend of (a) also applies to (b,c,d). In (a) LS,LSWS,SP+C,KF are partly out of scope.

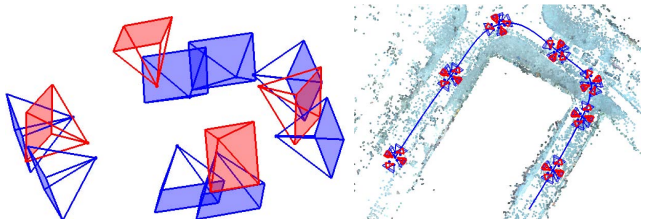

Fig. 3. Left: Rigid camera setup for street-view image capture. Blue camera: used in SfM, red: used only for evaluation. Right: Exemplary SfM Model from 300 frames. Path of van and mounted cameras in scene every 40 frames.

4.2 Street-View Video Sequence – Dataset and PnP Baseline

Since no public dataset for video registration is available we created our own: Street-view image data was captured with 8 cameras, rigidly mounted onto a van, in 1628x1236 resolution, at 10 fps for 30 seconds. The visible street scene was reconstructed with (on average) 400K 3D points. Additionally to the 8 cameras used for SfM, 2 or 4 additional cameras were mounted on the van for evaluation. SfM reconstruction with a rigid multi-camera installation on a moving van returns poses for all cameras and the van at all times. Using known rigid camera setup and van pose, precise pose ground truth can be inferred for the additional cameras as well. The rigid camera configuration and an example SfM Model can be seen in Fig. 3. Data was gathered in 4 locations, one of which is

Table 1. Time for parameter estimation via Cross Validation (SP, SP+C, KR), Expectation Maximization (KF), and solution (sec). Parameters in LS are set manually.

	KF	SP	SP+C	KR	LS
P.Estim.	< 1	7	81	637	NaN
Solution	< 1	< 1	2	164	52

Table 2. PnP failure rates (percent of all frames). A: pose estim. failure, no pose returned, B: failure to find good pose (pos.err < 1m), C: pose estim. failure when at least one other method found a good pose, D: failure to find good pose when all other methods returned good poses.

	OPnP	ASPnP	EPnP	RPnP
A	0.17	0.03	0.64	2.19
B	13.46	11.82	11.85	14.98
C	**0**	0	0.14	0.69
D	1.27	0.30	0.69	2.82

displayed in Fig. 3, resulting in 12 videos with each 300 frames. Typical problems for feature matching in this case are over/underexposure of the images, uneven distribution of feature locations, motion blur, lack of sufficient view overlap (the SfM cameras are looking down, the additional cameras are looking up). The additional cameras are (independently) registered to the SfM Model in our evaluation. PnP poses are obtained from OPnP, ASPnP, EPnP, RPnP. Table 2 compares (A) all methods in terms of percentages of failed pose estimates, (B) failure to find good (pos. err $< 1m$) poses, (C) failed pose estimates when at least one other method found a good pose, and (D) failure to find a good pose when all other methods found one. ASPnP offers the overall best performance.

Because strong pose outliers are still present for all PnP methods, we proceed to identify and remove outliers, and provide all refinement results for varying levels of removal. Ideally, outliers are identified automatically without the help of ground truth. This can be achieved by using positional differences between PnP poses of successive frames if outliers are randomly distributed and not systematic. For evaluation purposes we simplify the task and use the ground truth error for outlier removal: We define five positional error thresholds: $\rho_{1,\ldots,5} = \{.54, .65, 8.7, 47, \infty\}$ representing meters of allowed error in camera position to ground truth position. They correspond to $\{80, 85, 90, 95, 100\}$ percent of the data as inliers. Poses with an error above a chosen threshold are removed from the PnP baseline and considered as gaps. For ρ_5 we do not remove any pose.

In table 3 all PnP methods are listed with RMS positional error to ground truth in meters (left) and error of viewing direction in degrees (right) for $\rho_{1,\ldots,5}$. We note that the positional error increases significantly in all PnP methods once fewer outliers are removed. The same order of magnitude for positional errors for image-based registration in typical city scenarios is reported independently in [34,35,26,15]. Confirming [43], ASPnP offers the best results. EPnP is not as precise in easy pose estimation scenarios ($\rho_{1,2}$) but gains if outliers are present ($\rho_{3,4,5}$). Note how the orientational and positional errors compare: for pose errors around 5-6 meters, the error in orientation is still < 2 degrees. Average PnP runtimes (seconds) are: OPnP 1.31, ASPnP 0.22, EPnP, 0.25, RPnP: 0.13.

Table 3. PnP pose estimation errors. Left 4 col.: positional RMS error in meters, Right 4 col.: viewing direction errors in degrees (ignoring roll). Rows: Estimation error for outlier varying outlier threshold ρ. Thresholds are chosen such that $\{80, 85, 90, 95, 100\}$ percent of the data are inliers. See also table 4 (left) for median positional PnP errors.

	OPnP-t	ASPnP-t	EPnP-t	RPnP-t	OPnP-R	ASPnP-R	EPnP-R	RPnP-R
$\rho = \rho_1$	0.44	0.14	0.16	0.25	0.29	0.27	0.28	0.3
$\rho = \rho_2$	0.44	0.15	0.67	1.69	0.29	0.27	0.38	0.44
$\rho = \rho_3$	6.58	4.76	2.97	3.48	2.61	1.97	1.72	1.85
$\rho = \rho_4$	12.23	10.49	8.26	9.26	6.62	5.06	5.59	5.09
$\rho = \rho_5$	26.09	23.77	17.96	27.48	10.49	8.85	8.44	9.42

	OPnP-t	ASPnP-t	EPnP-t	RPnP-t
$\rho = \rho_1$	0.381 (K)	**0.065 (J)**	0.066 (J)	0.133 (J)
$\rho = \rho_2$	0.375 (K)	**0.064 (I)**	0.290 (E)	0.576 (I)
$\rho = \rho_3$	1.042 (L)	**0.599 (J)**	1.300 (L)	0.809 (J)
$\rho = \rho_4$	**1.479 (J)**	1.697 (L)	2.281 (L)	1.531 (L)
$\rho = \rho_5$	2.609 (J)	2.684 (L)	2.812 (J)	**2.267 (J)**

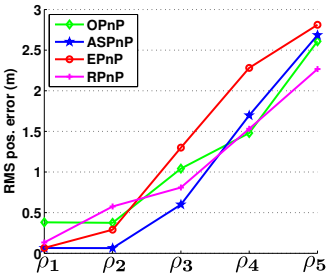

Fig. 4. Best refinement result (RMS position error) for all PnP methods. The letter (same as in table 5) indicates the method that gave best results. The graph (right) corresponds to the table (left) and plots best refinements for each PnP method over ρ.

In the remainder of the experiments ASPnP will be used as the preferred PnP baseline. We will report the results of *all* refinement methods based on ASPnP poses (Table 5). Additionally, we will provide the results of *only the best* refinement method based on all PnP baselines (Table 4 (right) & Fig. 4).

4.3 Street-View Video Sequence – Video Registration

We adopt the following shorthand notation. **A**: PnP baseline error, **B**: LSWS, **C**: KF, **D**: SP, **E**: SP+C, **F**: LSWS+SP, **G**: KR, **H**: LSWS+KR, **I**: LS, **J**: LS+CP, **K**: SP+LS, **L**: SP+LS+CP. Fig. 4 lists the best performing refinement method over all PnP methods with absolute positional RMS error. In table 5 the relative scores for all refinement methods in relation to the ASPnP baseline RMS positional error (Col. A) are listed (Col. B:L). Table 4 shows the best refinement results using median errors (right), and median PnP baseline error (left), to illustrate the performance when disregarding outliers. Table 1 lists runtimes for parameter estimation and solutions for the proposed methods. Comparing the best smoothing results for all PnP methods (Table 4 & Fig. 4) we observe:

- Refinement after ASPnP offers significantly better results with few outliers ($\rho_{1,2,3}$) than other PnP methods: the best method reduces the error of the

Table 4. Median PnP baseline positional error (left) and best median refinement error (right). This table shows the gain in positional accuracy from PnP baseline to best performing refinement method with respect to the median ground truth positional error. The letter (same as in table 5) indicates the used method.

	OPnP-t	ASPnP-t	EPnP-t	RPnP-t	OPnP-t	ASPnP-t	EPnP-t	RPnP-t
$\rho = \rho_1$	**.053 (A)**	.055 (A)	.074 (A)	.073 (A)	.042 (J)	**.041 (J)**	.043 (K)	.041 (F)
$\rho = \rho_2$	**.053 (A)**	.055 (A)	.075 (A)	.074 (A)	**.042 (K)**	.042 (K)	.043 (L)	.042 (I)
$\rho = \rho_3$	**.056 (A)**	.057 (A)	.079 (A)	.079 (A)	.044 (K)	**.043 (K)**	.044 (K)	.043 (J)
$\rho = \rho_4$	**.060 (A)**	.062 (A)	.086 (A)	.084 (A)	.046 (K)	**.045 (K)**	.047 (J)	.046 (J)
$\rho = \rho_5$	**.066 (A)**	.067 (A)	.092 (A)	.091 (A)	.053 (J)	.050 (J)	**.048 (L)**	.049 (J)

Table 5. Refinement results. Top 5 rows: positional RMS error (meters). Bottom 5 rows: orientation error (degrees). Col A: Baseline PnP error, Col B-L: avg. of *relative* improvement over baseline PnP pose error in all videos. Notation: **A**: PnP error, **B**: LSWS, **C**: KF, **D**: SP, **E**: SP+C, **F**: LSWS+SP, **G**: KR, **H**: LSWS+KR, **I**: LS, **J**: LS+CP, **K**: SP+LS, **L**: SP+LS+CP

	A	B	C	D	E	F	G	H	I	J	K	L
$\rho = \rho_1$	0.14	1.30	0.90	1.19	1.01	1.56	0.51	0.52	2.18	2.26	2.16	2.22
$\rho = \rho_2$	0.15	1.24	0.96	1.19	1.01	1.53	0.46	0.44	2.23	2.26	2.19	2.24
$\rho = \rho_3$	4.76	1.03	1.15	1.82	1	1.91	1.23	1.27	20.21	23.56	19.34	21.92
$\rho = \rho_4$	10.49	1	1.11	2.23	1.24	2.20	2.01	2.03	18.40	21.99	19.65	20.76
$\rho = \rho_5$	23.77	1	1.23	3.51	1.60	3.56	5.50	5.62	59.68	58.36	37.03	38.85
$\rho = \rho_1$	0.27	1.10	0.94	1.10	1.01	1.20	0.79	0.80	1.34	1.35	1.34	1.35
$\rho = \rho_2$	0.27	1.08	1.02	1.10	1.01	1.18	0.72	0.73	1.34	1.33	1.35	1.33
$\rho = \rho_3$	1.97	1.04	0.99	1.36	1	1.39	1.06	1.13	1.18	1.18	1.43	1.42
$\rho = \rho_4$	5.06	1.02	0.98	1.65	0.97	1.70	1.31	1.30	1.04	1.06	1.29	1.29
$\rho = \rho_5$	8.85	1.01	0.97	2.54	0.95	2.58	1.76	1.76	0.99	1	1.45	1.44

worst method by 83 percent. For ($\rho_{4,5}$) PnP dependency decreases: best method reduces the error of the worst method by only 19 percent.

- In general, least-squares techniques, LS (I), LS+CP (J), SP+LS (K), SP+LS+CP (L) offer the best performance for all outlier levels.
- For $\rho_{4,5}$ the introduction of the camera plane assumption and the initialization using splines SP+LS (K), SP+LS+CP (L) offer a small gain.
- For median errors (table 4) variants of LS (I,J,K,L) also perform best. CP inclusion gives no improvement. The results do not depend on the PnP method.

Comparing *relative* method accuracy for ASPnP as baseline (table 5) we observe:

- SP (D) is increasingly helpful with growing outlier contamination ($\rho_{3,4,5}$). Similar to results on synthetic data, SP+C (E) does not help much.
- In general, all LS variants (I,J,K,L) offer significantly better results than any other method. In contrast to our experiments on synthetic data, for real data

LSWS (B) and, as a consequence LS (I) / LS+CP (J) have similar scores in positional accuracy to SP+LS (K) / SP+LS+CP (L). Initialization using splines slightly improves orientation estimation. Inclusion of a CP in the optimization marginally improves the result, but leads to a slower convergence.

- KF (C) and KR (G,H) help primarily in case of many outliers ($\rho_{4,5}$), LSWS (B) helps for ($\rho_{1,2,3}$). Initializing SP (D) or KR (G) with LSWS (B) in LSWS+SP (F), LSWS+KR (H) leads to a marginal improvement.

As in our experiment on synthetic data, variants of LS perform best on real data as well. The influence of initial PnP poses is weak if few outliers are present.

Comparison with registration after reconstruction: An alternative way of video registration is SfM reconstruction of a query video, and alignment of the new model to the ground truth. We reconstructed every video with standard SfM tools. The camera poses were rigidly aligned to the ground truth by minimizing the RMS positional error. The resulting positional error of **9.051 meters** is significantly worse than our best refinement result with an error of **2.267 meters** (See fig. 4, ρ_5 for no outlier removal). This is mainly due to SfM model deformation and fragmentation. See supplementary material for more details.

4.4 Gap Interpolation

There are three scenarios where gaps, i.e. missing pose estimates for a consecutive number of frames, can occur: 1) Failure of the PnP algorithm to converge, 2) Removal of identified pose outliers, 3) deliberate speed-up by matching every nth frame to the SfM model. In our third experiment gaps are created deliberately in the synthetic dataset (Sec. 4.1, fast non-linear camera motion) and street-view dataset (Sec 4.2, mostly linear, slow camera motion). We keep every nth pose, leaving the remaining frames as gaps, and refine the camera path. We interpolate over the gaps with cubic interpolation splines by using the refined poses as knots,

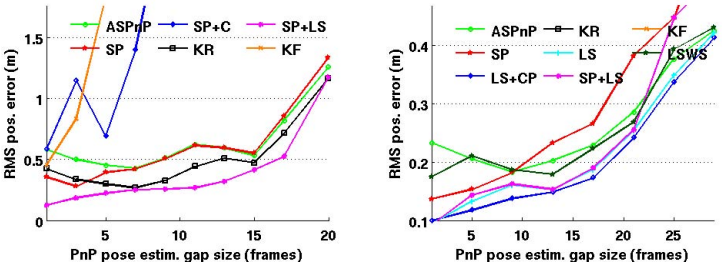

Fig. 5. Positional RMS pose error for refined poses after gap interpolation. Every nth PnP pose is kept, remaining frames are gaps, poses are smoothed, and result interpolated with cubic splines. Left: Synthetic dataset, Right: Street-view videos. Note: due to high, very volatile error LSWS, LS (left) and SP+C (right) were not plotted. KF and KR (right) exhibit high error and are out of scope.

and evaluate on all frames including gaps. Fig. 5 shows the positional RMS error for increasing gap sizes on real (right) and synthetic data (left). We observe:

- KR and KF are rarely helpful: For large gaps KF's linear dynamics assumption is violated, conditional averaging in KR is unstable. LS,LSWS,SP+C have the same problem as in Sec. 4.1.
- SP shows reliable refinement over gaps. The gain over the ASPnP baseline depends on the degree of non-linearity of camera motion: The camera motion eliminates any gain after 7,10 (synthetic,real) skipped frames.
- As in our previous experiments, SP+LS offers the overall best performance on both datasets. The inclusion of a CP as constraint offers an additional boost in real data. SP+LS (left) and LS+CP (right) lose their gain over the baseline only after 20 and 30 skipped frames, respectively.

5 Discussion and Conclusion

The three experiments show that in *all test cases* in synthetic and real data most proposed methods improve pose accuracy over frame-wise registered poses by including temporal smoothness. The best achieved improvement ranges from 2 to 60-fold reduction in RMS positional error depending on the outlier contamination and magnitude of pose changes between frames.

Generally, variants of LS provide the best results for positional, but not for orientational accuracy. The positional accuracy can be improved further by adding additional constraints, such as a planar motion assumption (LS+CP). Robust initialization (SP+LS) can help with convergence when strong noise in 2D and 3D is present. SP provides the fastest method with good results. Inclusion of camera parameter covariances (SP+C) did not improve accuracy due to many spurious feature matches. KR was able to handle outlying poses efficiently, but conditional averaging decreases accuracy when poses are already good. If speed is a constraint SP and LS scale linearly and are close to real-time performance. In case orientation is more important than position and the data is strongly contaminated with outliers, SP offers the best performance. Even for medium sized SfM models ($\sim 10^5$ 3D points) frame-wise feature matching and pose estimation is likely to be slower than our proposed pose refinements. This can be mitigated by matching only every nth frame, smoothing and interpolating. In case of interpolation, LS performs best, followed by SP. For real SfM data we note that refinement results strongly depend on the initially used PnP algorithm in case of few outliers (ρ_1) but not so for many outliers (ρ_5) : ratio of best to worst result: 0.17 for ρ_1, but 0.81 for ρ_5. (See Fig. 4). The resulting refinement methods are applicable in many domains where video poses are needed: Besides 2D-3D correspondences no further knowledge is required.

The present work opens three main branches of future work. First, from the large body of works on regularization, basis expansion, and probabilistic kernel methods, we adapted several techniques (SP, KR) to the problem of video registration. Different parametrizations and techniques, such as random regression forests can be examined. Second, combinations of this method with pose estimation through feature tracking [27,37] can be explored. Third, the LS refinement

can naturally be combined with previous works on video pose estimation where SLAM is applied to a video first, and matched to a 3D world afterwards [41,33].

Acknowledgments. This work was supported by the European Research Council (ERC) under the project VarCity (#273940).

References

1. Agarwal, S., Snavely, N., Simon, I., Seitz, S., Szeliski, R.: Building Rome in a day. In: ICCV (2009)
2. Agrawal, M.: A Lie Algebraic Approach for Consistent Pose Registration for General Euclidean Motion. In: IEEE/RSJ IROS (2013)
3. Aubry, M., Russell, B.C., Sivic, J.: Painting-to-3D Model Alignment Via Discriminative Visual Elements. ACM TOG (2013)
4. Bergamo, A., Torresani, L.: Leveraging Structure from Motion to Learn Discriminative Codebooks for Scalable Landmark Classification. In: CVPR (2013)
5. Boix, X., Gygli, M., Roig, G., Van Gool, L.: Sparse Quantization for Patch Description. In: CVPR (2013)
6. Brubaker, M.A., Geiger, A., Urtasun, R.: Lost! Leveraging the Crowd for Probabilistic Visual Self-Localization. In: CVPR (2013)
7. Cao, S., Snavely, N.: Graph-Based Discriminative Learning for Location Recognition. In: CVPR (2013)
8. Davison, A.J., Reid, I.D., Molton, N.D., Stasse, O.: MonoSLAM: real-time single camera SLAM. PAMI (2007)
9. Gordon, I., Lowe, D.G.: What and Where: 3D Object Recognition with Accurate Pose. In: CLOR 2006 (2006)
10. Hakeem, A., Vezzani, R., Shah, M., Cucchiara, R.: Estimating Geospatial Trajectory of a Moving Camera. In: ICPR (2006)
11. Hao, Q., Cai, R., Li, Z., Zhang, L., Pang, Y., Wu, F.: 3D visual phrases for landmark recognition. In: CVPR (2012)
12. Hao, Q., Cai, R., Li, Z., Zhang, L., Pang, Y., Wu, F., Rui, Y.: Efficient 2D-to-3D Correspondence Filtering for Scalable 3D Object Recognition. In: CVPR (2013)
13. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning, Data Mining, Inference, and Prediction, 2nd edn. Springer (2009)
14. Hays, J., Efros, A.A.: IM 2 GPS: estimating geographic information from a single image. In: CVPR (2008)
15. Hsu, S., Samarasekera, S., Kumar, R., Sawhney, H.S.: Pose estimation, model refinement, and enhanced visualization using video. In: CVPR (2000)
16. Irschara, A., Zach, C., Frahm, J.M., Bischof, H.: From structure-from-motion point clouds to fast location recognition. In: CVPR (2009)
17. Kalantidis, Y., Tolias, G., Avrithis, Y.: Viral: Visual image retrieval and localization. Multimedia Tools and Applications (2011)
18. Kalogerakis, E., Vesselova, O., Hays, J., Efros, A.A., Hertzmann, A.: Image Sequence Geolocation with Human Travel Priors. In: ICCV (2009)
19. Klein, G., Murray, D.: Parallel Tracking and Mapping for Small AR Workspaces. In: ISMAR (2007)
20. Klingner, B., Martin, D., Roseborough, J.: Street View Motion-from-Structure-from-Motion. In: ICCV (2013)

21. Knopp, J., Sivic, J., Pajdla, T.: Avoiding confusing features in place recognition. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part I. LNCS, vol. 6311, pp. 748–761. Springer, Heidelberg (2010)
22. Koch, O., Teller, S.: Wide-Area Egomotion Estimation from Known 3D Structure. In: CVPR (2007)
23. Lepetit, V., Moreno-Noguer, F., Fua, P.: EPnP: An Accurate O(n) Solution to the PnP Problem. IJCV (2009)
24. Li, S., Xu, C., Xie, M.: A Robust O(n) Solution to the Perspective-n-Point Problem. PAMI (2012)
25. Li, Y., Snavely, N., Huttenlocher, D., Fua, P.: Worldwide pose estimation using 3D point clouds. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part I. LNCS, vol. 7572, pp. 15–29. Springer, Heidelberg (2012)
26. Li, Y., Snavely, N., Huttenlocher, D.P.: Location Recognition using Prioritized Feature Matching. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part II. LNCS, vol. 6312, pp. 791–804. Springer, Heidelberg (2010)
27. Lim, H., Sinha, S.N., Cohen, M.F., Uyttendaele, M.: Real-time image-based 6-dof localization in large-scale environments. In: CVPR (2012)
28. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. IJCV (2004)
29. Newcombe, R.A., Davison, A.J.: Live dense reconstruction with a single moving camera. In: CVPR (2010)
30. Newcombe, R.A., Lovegrove, S.J., Davison, A.J.: DTAM: Dense tracking and mapping in real-time. In: ICCV (2011)
31. Ramalingam, S., Bouaziz, S., Sturm, P.: Pose Estimation Using Both Points and Lines for Geolocation. In: ICRA (2011)
32. Robertson, D., Cipolla, R.: An Image-Based System for Urban Navigation. In: BMVC (2004)
33. Rodriguez, J., Aggarwal, J.: Matching aerial images to 3-D terrain maps. PAMI (1990)
34. Sattler, T., Leibe, B., Kobbelt, L.: Fast image-based localization using direct 2D-to-3D matching. In: ICCV (2011)
35. Sattler, T., Leibe, B., Kobbelt, L.: Improving Image-Based Localization by Active Correspondence Search. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part I. LNCS, vol. 7572, pp. 752–765. Springer, Heidelberg (2012)
36. Schindler, G., Brown, M., Szeliski, R.: City-Scale Location Recognition. In: CVPR (2007)
37. Se, S., Lowe, D., Little, J.: Vision-based mobile robot localization and mapping using scale-invariant features. In: ICRA (2001)
38. Tanskanen, P., Kolev, K., Meier, L., Camposeco, F., Saurer, O., Pollefeys, M.: Live Metric 3D Reconstruction on Mobile Phones. In: ICCV (2013)
39. Vaca-Castano, G., Zamir, A.R., Shah, M.: City scale geo-spatial trajectory estimation of a moving camera. In: CVPR (2012)
40. Zamir, A.R., Shah, M.: Accurate image localization based on google maps street view. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 255–268. Springer, Heidelberg (2010)
41. Zhao, W., Nister, D., Hsu, S.: Alignment of continuous video onto 3D point clouds. In: CVPR (2004)
42. Zheng, Y., Kuang, Y., Sugimoto, S., Aström, K., Okutomi, M.: Revisiting the PnP Problem: A Fast, General and Optimal Solution. In: ICCV (2013)
43. Zheng, Y., Sugimoto, S., Okutomi, M.: ASPnP: An Accurate and Scalable Solution to the Perspective-n-Point Problem. IEICE TIS (2013)