Live Tracking and Mapping from Both General and Rotation-Only Camera Motion

Steffen Gauglitz* Chris Sweeney* Jonathan Ventura* Matthew Turk* Tobias Höllerer*

Department of Computer Science, University of California, Santa Barbara

Abstract

We present an approach to real-time tracking and mapping that supports any type of camera motion in 3D environments, that is, general (parallax-inducing) as well as rotation-only (degenerate) motions. Our approach effectively generalizes both a panorama mapping and tracking system and a keyframe-based Simultaneous Localization and Mapping (SLAM) system, behaving like one or the other depending on the camera movement. It seamlessly switches between the two and is thus able to track and map through arbitrary sequences of general and rotation-only camera movements.

Key elements of our approach are to design each system component such that it is compatible with both panoramic data and Structure-from-Motion data, and the use of the 'Geometric Robust Information Criterion' to decide whether the transformation between a given pair of frames can best be modeled with an essential matrix E, or with a homography H. Further key features are that no separate initialization step is needed, that the reconstruction is unbiased, and that the system continues to collect and map data after tracking failure, thus creating separate tracks which are later merged if they overlap. The latter is in contrast to most existing tracking and mapping systems, which suspend tracking and mapping, thus discarding valuable data, while trying to relocalize the camera with respect to the initial map.

We tested our system on a variety of video sequences, successfully tracking through different camera motions and fully automatically building panoramas as well as 3D structures.

1 Introduction

Over the past decade, there has been a tremendous amount of work on real-time monocular vision-based tracking and mapping (T&M) systems, that is, systems that simultaneously determine the position and/or orientation of the camera with respect to a previously unknown environment and create a model of this environment. Aside from other applications (such as navigation of an autonomous vehicle), T&M is an important enabling technology for Augmented Reality (AR) in unprepared environments.

An important characteristic of a T&M system is the type of camera motion and the geometry of the environment that it supports. For example, a system may assume a planar environment [25], or a camera that is rotating around its optical center [8, 37]. Simultaneous Localization and Mapping (SLAM) systems such as [7, 9, 14, 22] can deal with environments of arbitrary geometry and any camera motion that induces parallax (referred to as *general* camera motions). However, with few exceptions [4], they do not support rotation-only camera motion: Since SLAM systems are designed primarily to handle a traveling camera, their mapping is, intrinsically, built upon triangulation of features. Thus, they require that each feature be observed from two distinct camera locations and may produce degenerate maps or fail completely if the camera rotates from one part of the environment to another.

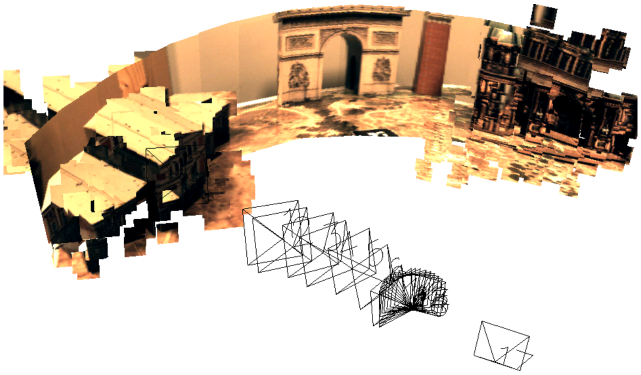


Figure 1: Our system supports both general and rotation-only camera motion. In the former case, it acts as a SLAM system and models 3D structure in the environment; in the latter case, it acts as a panorama mapper. It seamlessly switches between the two modes, thus being able to track and map through arbitrary sequences of general and rotation-only camera movements and — fully automatically and in real time — creating combinations of 3D structure and panoramic maps, as shown here.

Therefore, most SLAM systems need to be initialized with a distinct "traveling" movement of the camera *for each newly observed part of the environment*. This restriction is acceptable for vehicle navigation or if building a model of the environment is the user's intent (cf. Pollefeys et al. [26]: "Here we will assume that care is taken during acquisition to not take multiple images from the same position so that this problem doesn't occur"). However, it is a major limitation for their use in AR, where the model building is assumed to be done in the background and ideally transparent to the user, who should not be required to move a certain way in order to make the system work. Moreover, rotation-only "looking around" is a very natural motion and may occur in many AR applications.

Our particular motivation is the use of a T&M system for remote collaboration [11]. Here, the emerging model of the environment is used to allow a physically remote user to view and navigate the environment, and spatial annotations that are registered to this model and overlaid onto the real world via an appropriate AR display are used to communicate visual/spatial information. For this and many other applications, the paradigm for modeling the environment should be to make the best possible use of all data that can be casually collected and to enable viewing and placement of annotations for as much time as possible. In particular, this means not forcing the user to concentrate on model building, and not discarding all frames that stem from rotation-only movements (as in most SLAM systems) or as soon as the user starts moving (as with panorama mapping). It should be noted that there is a trade-off involved between the objective of creating a completely coherent model and the objective of being able to freely move the camera and enable viewing and placement of annotations at all times.

*{sgauglitz, cmsweeney, jventura, mturk, holl}@cs.ucsb.edu

In this paper, we present an approach for the latter: a concept for real-time tracking and mapping that explicitly supports both general and rotation-only camera motions in 3D environments, does not need a separate initialization step, and continues to collect data despite intermittent tracking loss. In the case of intermittent tracking loss, it creates several disjoint maps which are later merged if possible. (The idea of multiple sub-maps resembles the concept of PTAMM [3], but they are created for an entirely different reason.)

One key element of our approach is the use of the 'Geometric Robust Information Criterion' (GRIC) by Torr [34] to decide whether the transformation between a given pair of frames can best be modeled with an essential matrix E, or with a homography H. It should be noted that these two transformations are a complete *partition* of all cases of camera motion that can occur in a static environment, i.e., exactly one of them is correct.

2 Related Work

2.1 Monocular vision-based SLAM

SLAM is the problem of determining the pose of the observer relative to an unknown environment and at the same time creating a model of the environment (which may have arbitrary geometric complexity) while the observer moves around. In the case of monocular vision-based SLAM [7], the only sensor used to accomplish this task is a single camera.

Filter-based SLAM systems [5, 7, 9] maintain estimates of both camera and feature positions (i.e., the map) in a large state vector which is updated using Kalman filters in each frame. In contrast, keyframe-based systems as pioneered by Klein and Murray [14, 15, 16] (PTAM) track features in each frame, but use only selected frames to update their map, typically using bundle adjustment for the latter. While all aforementioned system are based on sparse features, Newcombe et al. [22] presented a keyframe-based system that uses dense mapping. They report extremely robust results, but (in contrast to PTAM) require a GPU.

In both types of systems, the map is designed to store Structure-from-Motion (SfM) data, that is, feature positions in 3D that have been triangulated using observations from multiple viewpoints. Thus, they typically require parallax-inducing camera motion in order to bootstrap their map [7, 14, 22] (otherwise, the features cannot be triangulated and integrated into the map). In some systems [14, 22], the initialization is performed as a dedicated separate step, and tracking quality crucially depends on the quality of this initialization. Rotation-only motions are inherently not supported, unless they are fully constrained to the already observed part of the scene.

For filter-based systems, an alternative, six-dimensional parametrization of the feature locations [5] can provide a remedy, supporting rotation-only motion to some extent by admitting features with an uninformative depth prior and filtering the features through multiple motion models [4] to constrain their uncertainty. However, this support comes at a high computational cost: The — already very high — cost of filtering of each feature point is further increased by doubling the dimensionality of the feature state vector as well as computing the results for multiple motion models (note that Civera et al. [4] use seven models). As a result, the number of features that can be tracked in real time, which is typically already smaller for filter-based SLAM as compared to keyframe-based SLAM[1], is further decreased. (With just one motion model, Civera et al. [5] mention map sizes of up to one hundred features, compared to several thousand for PTAM [14]. For the approach with multiple models [4], no real-time implementation is described.) In the case of long sequences of camera rotation, many of those computation cycles are spent filtering data where no gain is to be expected (namely, re-estimating the (still undefined) feature depth) and are redundant at best and harmful (because noise may lead to spurious depth estimates) at worst.

Therefore, we consider it an advantage of our approach that we explicitly switch to panoramic mapping if supported by the observations, thus being able to take advantage of some of the advantages that panoramic mapping offers — such as, a robust outlier-resilient model (homography), a straight-forward mapping of the entire frame instead of sparse features, as well as easy navigation, all of which are especially important for AR. On the other hand, the approach of Civera et al. [4] appears to have merit for modeling the transition between two types of movement, or when only some of the features exhibit parallax.

While admitting features without depth could in principle be adopted for keyframe-based SLAM (in [16], this approach is employed to admit features before their depth is known), the ability to rely on them exclusively would require fundamental and possibly costly changes to the underlying mapping and bundle adjustment (cf. [31] Sec. 3.7 for issues in the context of points at infinity). We are not aware of any existing keyframe-based SLAM system which explicitly supports rotation-only movements.

2.2 Further related work

Panorama mapping systems. Like a SLAM system, a panoramic tracking and mapping systems aim at modeling the environment while determining the pose of the camera, but in this case, the camera is assumed to rotate around its optical center, so that only its orientation has to be determined. An early real-time system is Envisor [8]. Wagner et al. [37] describe a system that operates robustly and in real time on a mobile device.

Environment modeling using other sensors. In theory, using stereo cameras [18, 39] solves the problem of requiring the camera to travel, since the baseline required to triangulate features is built-in. In practice, however, using stereo cameras is only a partial remedy, since the baseline has to be significant in relation to the distance to the environment in order to reliably estimate depth. Thus, a wearable stereo system would be unable to map a building across the street without requiring the user to provide additional baseline by traveling (while a panorama system, though unable to provide depth, will produce very usable information).

Further, systems based on alternative sensor types should be considered. In particular, active depth sensors based on time-of-flight or structured light [19, 23] have recently generated significant interest and can arguably provide for more detailed models and more robust tracking when lighting conditions for vision-based tracking are unfavorable. However, all of these systems have other inherent limitations such as limited range, inability to work in sunlight, and need for additional special hardware. Further, to present the model to the user, it is necessary to include a color camera and thus additional algorithmic steps (sensor fusion) have to be integrated.

We thus argue that there are both theoretical and practical interests in solving T&M using monocular vision.

Model selection, GRIC score and applications. Model selection is defined as choosing the right model that best describes a set of observations. Various metrics (frequently dubbed "information criteria") have been proposed to assess the fitness of a particular model given the data, for example minimum description length [28], AIC [1], and BIC [30]. Torr described both a maximum likelihood [35] and a Bayesian formulation [34] of GRIC, and we use the latter in this work. These information criteria are very general in nature and can be applied to various types of models. In SfM, the GRIC score has been applied particularly to detect homographies in order to *avoid* them during keyframe selection [26, 27].

[1] For an interesting comparison of the relative computational cost of the two approaches see Strasdat et al. [33].

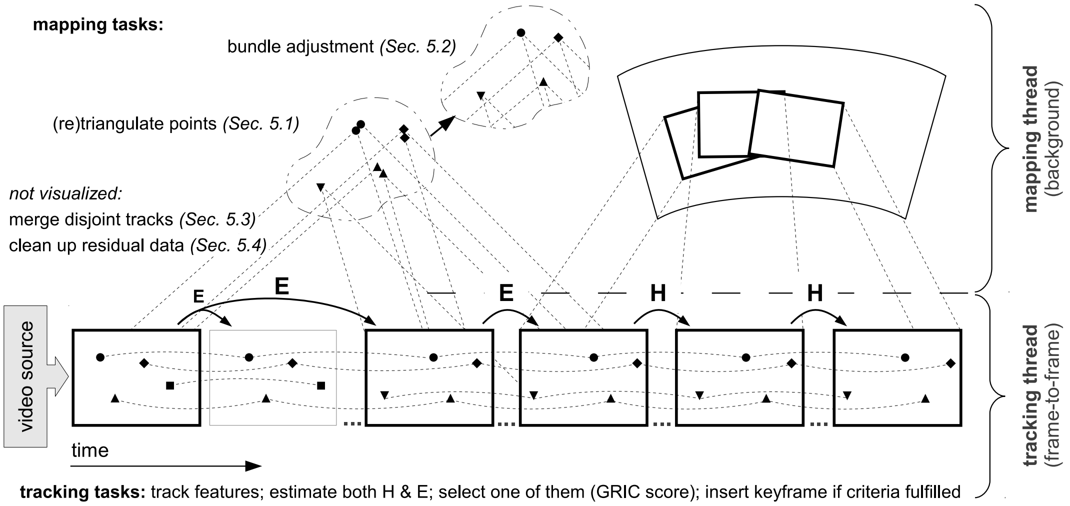

mapping tasks:
bundle adjustment (Sec. 5.2)
(re)triangulate points (Sec. 5.1)
not visualized:
merge disjoint tracks (Sec. 5.3)
clean up residual data (Sec. 5.4)
E
E
E
H
H
video source
time
tracking tasks: track features; estimate both H & E; select one of them (GRIC score); insert keyframe if criteria fulfilled
mapping thread
(background)
tracking thread
(frame-to-frame)

Figure 2: Conceptual overview of the main components of our system. The tracking thread is responsible for processing the incoming frames. When certain criteria are fulfilled, it inserts a keyframe (Section 4.1), which is connected to the previous keyframe by either an essential matrix or a homography, depending on the GRIC score (Section 6). The emerging chain of keyframes is clustered in *keyframe groups* based on the type of transformation between them (Section 3.1). Each keyframe group governs one sub-map of the environment model, consisting either of 3D structure or a (partial) panorama. When tracking is intermittently lost, a new *track* is started, which can later be merged if the sub-maps overlap (Section 5.3).

3 System Architecture Overview

Our concept borrows two key ideas from Klein and Murray's Parallel Tracking and Mapping (PTAM) system [14, 15, 16], namely, the central role of keyframes, and the splitting of tracking and mapping into two parallel threads. For the latter, the split is taken even further in that the tracking thread does not immediately require any data from the mapping thread to process the next frame. By doing so, the tracking thread can operate in much the same way independent of whether the camera currently undergoes a traveling or a rotation-only movement.

Fig. 2 presents a conceptual overview of the main components of our system. In designing the system, we followed the following guidelines: (1) the tracking thread should be as fast as possible and leave all tasks that are not imminent in order for the next frame to be processed to the mapping thread, which runs asynchronously in the background; (2) the first steps in the pipeline should not be dependent on the model that will be selected. The tracking and mapping threads are described in detail in Sections 4 and 5, respectively. Section 6 describes the problem of model selection and the GRIC score in detail.

3.1 Data structures

Fig. 3 visualizes the main data objects that store the current system state and the emerging map as well as their relations.

The set of keyframes form a graph, with individual keyframes as nodes, and transformations (essential matrices and homographies) as links (cf. the representation in [9]). The set of keyframe groups also forms a graph, where each node represents a set of keyframes (cf. the "epipolar graph" used for incremental SfM by Klopschitz et al. [17]). While tracking is continuous, both graphs are linear, but more complex topologies emerge from tracking loss (disconnected subgraphs) or recovery (branching graph).

The most central element is the keyframe group: each keyframe group governs one sub-map consisting of a set of keyframes which are all linked by essential matrices (E-group), or all linked by homographies (H-group). The keyframe group also determines the frame of reference, with respect to which all 3D pose information and homographic warps, respectively, are stored. A keyframe may be part of several groups (for example, in one H-group and one E-group), in which case it gets assigned a pose in each of its groups (e.g., keyframe 3 in Fig. 3).

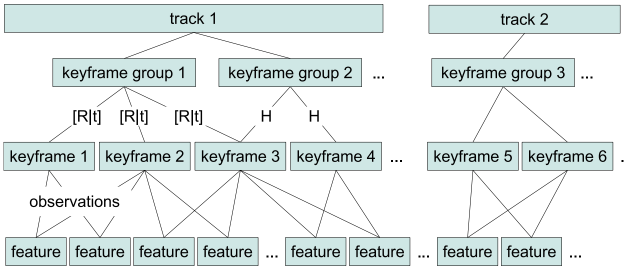


Figure 3: Overview of the main data structures used to store the emerging map(s).

When tracking is lost, all links to the current keyframe and keyframe group are lost, and the tracker starts a new track. Initially, the new track is completely unconnected to the previous data, but can later be merged (if there is some overlap in what is observed during both tracks) as explained in Section 5.3.

Rotation-only camera motion vs. planar environments. It should be noted that there are two reasons for why the transformation between two images is best modeled by a homography: either because the camera underwent rotation-only movement, or because the observed part of the environment is planar. For all rotations, one can construct an equivalent camera motion along a planar environment, so without making further assumptions the two cases cannot be distinguished based on image data alone. However, we emphasize that our concept does not require the ability to distinguish the two cases. Only for *visualization* of the results (Figs. 1 and 5) do we make the assumption that homographies describe rotation-only movements, but the system is agnostic to this assumption.

Since the two cases cannot be distinguished, one can indeed not recover the exact 3D trajectory of the camera. With the application of AR in mind, one can, however, ensure correct registration of all annotations with respect to the scene.

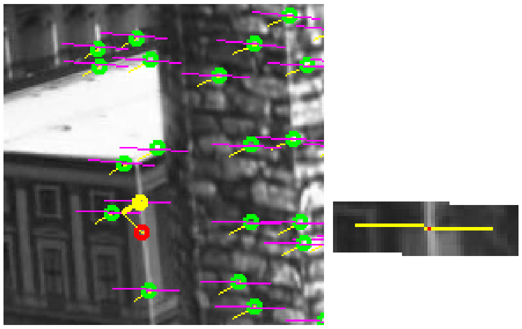


Figure 4: Feature re-estimation constrained by epipolar geometry. Tentative feature correspondences are created using NCC-based template matching. Features shown in green are identified as inliers to the estimated essential matrix $\hat{E}$, with epiline segments indicated in purple. The outlier (red) is re-estimated by sampling a strip along the epiline segment (right image, enlarged) and re-evaluating the NCC scores along the epiline. The re-estimated feature position is shown in yellow in the left image.

4 Tracking

Keypoints are detected using the FAST corner detector [29]. We enforce a spatially well-distributed set of keypoints, which was shown to improve tracking robustness [10, 13], by overlaying a simple rectangular grid over the image and selecting the strongest keypoint in each cell. Frame-to-frame feature correspondences are created using a multi-level, active search patch tracker with normalized cross-correlation (NCC)-based template matching. On the full-resolution image, the feature location is refined to subpixel accuracy by using a quadratic fit to neighboring scores. This is similar in particular to the keypoint tracking by Wagner et al. [36]; however in contrast to their system, the multi-level tracking is executed on a per-feature basis (instead of interleaved with the pose estimation), to make the first steps in the tracking pipeline agnostic about the type of camera motion (and thus the model that would need to be enforced).

From all feature correspondences, we estimate both a homography $\hat{H}$ and an essential matrix $\hat{E}$ between the last keyframe k_0 and the current frame using MAPSAC [34] (for the latter, Nistér's five-point algorithm [24] is used to generate the hypothesis inside MAPSAC).

Next, the GRIC score (cf. Section 6) is computed for both models, and the better model (that is, the one with lower GRIC score) is selected. If $\hat{E}$ is determined to be the better fit, it is decomposed into relative pose information $[\hat{R}|\hat{t}]$.

The measurement error σ (needed for MAPSAC, and, more crucially, the GRIC score) is estimated from all inliers and used for the next frames, averaged over a window of 10 frames to be more robust to a single frame with bad model fit. With our test sequences (cf. Section 7), the range of estimates for σ is about 0.5 to 1.5 pixels — if tracking of a feature succeeds, it is highly accurate.

For either model, outliers are re-estimated. This is trivial in the case of $\hat{H}$, since the model defines a unique mapping. In the case of $\hat{E}$, each outlier is re-estimated by sampling a strip along the epiline segment, re-evaluating the NCC scores along the epiline, and setting the feature position to the point with highest NCC score. This process is visualized in Fig. 4 and was found to improve tracking and model quality, since the feature tracks can be maintained for longer (instead of finding new features to replace the lost ones). Features that prove unreliable (i.e., features that repeatedly are outliers and need to be re-estimated) are removed.

If no keyframe is added (cf. next section), processing of this frame is completed.

4.1 Inserting a new keyframe

The tracking thread adds the current frame as a new keyframe k_{+1} when several conditions (similar to the ones suggested by Klein and Murray [14]) are met: (1) Tracking quality is good (as determined by the fraction of inliers that MAPSAC finds); (2) enough time has passed since the last keyframe insertion; (3) in the case of $\hat{H}$, when the median 2D distance that the keypoints "traveled" since the last keyframe is large enough, and in the case of $\hat{E}$, when the median feature triangulation angle is large enough. For homographies, the 2D distance is directly correlated with the camera's angle of rotation if $\hat{H}$ describes a rotation, but it also captures pure translational movements (along a planar surface). In the case of $\hat{E}$, requiring a minimum angle under which a feature is observed is equivalent to requiring a minimum baseline relative to the distance to the model and ensures that the depth estimates are reasonably conditioned [31].

If the transformation between the second-last keyframe k_{-1} and the last keyframe k_0 is the same type as the one estimated between k_0 and the newly inserted keyframe k_{+1}, the new information gets merged into the existing keyframe group as described in the next two paragraphs. If the new transformation is of a different type, a new keyframe group gets created, such that k_0 is the last keyframe in the old group, and the first keyframe (together with k_{+1}) of the new group (cf. keyframe 3 in Fig. 3).

Merging a new keyframe into an H-group. For homographies, a common frame of reference is adopted by multiplying the current estimate $\hat{H}$ with the k_0's pose H_{k_0} in the current group.

Merging a new keyframe into an E-group. For essential matrices, adopting a common frame of reference follows the same idea, but is slightly more involved: the new keyframe's pose can be described as $[R|t]_{k_{+1}} = [R|t]_{k_0} \cdot [\hat{R}|\hat{t}]$. However, the scale of $\hat{E}$ is arbitrary, and thus the scale of $[R|t]_{k_{+1}}$ is not defined. To arrive at a common scale, we use the set of all feature observations that have a triangulated position in both the existing E-group as well as with respect to $\hat{E}$, and calculate the ratios of their distances to k_0 in both coordinate systems. We then take the median of those ratios as a robust measure of the scale between the two point clouds and scale $t_{k_{+1}}$ accordingly.

This strategy of merging two local reconstructions is similar to the offline incremental SfM system by Klopschitz et al. [17] (although the merging of two reconstructions is implemented differently). It should be noted that with this strategy, all of the keyframe pairs and thus all of the local maps are given the same weight (the order of $\{k_{-1}, k_0\}$, $\{k_0, k_{+1}\}$ does not influence the result), i.e., the reconstruction is unbiased. This is in contrast to SLAM systems that have a dedicated initialization step, in which the quality of the map is dependent on the frames involved in this initialization step in particular.

New features are detected in all uncovered image regions by applying the same grid as in the first frame and choosing new features for each cell that is not covered by currently tracked features.

4.2 Relocalizing vs. starting a new track

When tracking gets lost — in our case, when MAPSAC fails to find a model with sufficient support for either E and H — the standard strategy employed in most T&M systems (e.g., in [14, 15, 25, 37]) is to continuously try to *relocalize* the camera pose with respect to the current map with each new frame until successful. However, this means that tracking and mapping are suspended and no data is collected until relocalization is successful.

Here, we employ an alternative strategy proposed by Eade and Drummond [9]: instead of trying to relocalize, we let the tracker start a new track immediately, and leave it to the background thread to later merge tracks if possible (cf. Section 5.3). The benefit of this

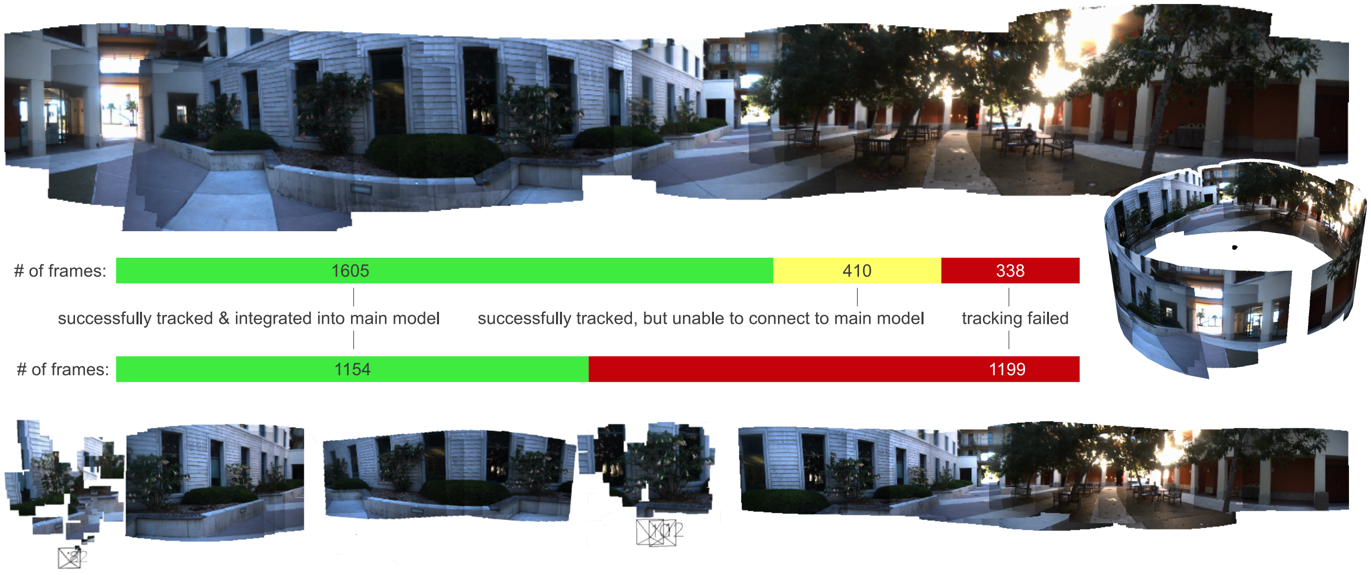


Figure 5: Merging of tracks (top) vs. relocalization (bottom). The rotation-only input video for the data shown here contains several rapid camera motions (cf. description of the dataset in [6]), resulting in intermittent tracking loss. After each loss, our system starts a new partial panorama, and, while the incoming live frames are being processed, attempts to stitch them together in the background. The final panorama (top) is an almost complete horizontal panorama of the scene (cf. cylindrical rendering on the right). 1605 frames are registered to this panorama. Another 410 frames are registered to partial panoramas (not shown) which the system was unable to connect to the main model (whether these should be counted as success or failure depends on the application that the system is used for). For 338 frames, tracking failed. In comparison, with the same tracking system but using relocalization, only the model at the bottom gets mapped, with 1154 registered frames. (Here, the model is visualized in five segments because the transition between two pairs of keyframes got modeled with an essential matrix, but the model is — by construction — fully connected.) The data of all other 1199 frames (which "passed by" while the system unsuccessfully tried to relocalize) are discarded.

method, illustrated in Fig. 5, is that the system continues to collect data even after tracking failure occurs, and, if the tracks overlap, efficiently merges them (if they do not overlap, a recovery-based system would never recover), so that no data is lost.

5 Mapping

The mapping thread is executed in parallel to the tracking thread and cycles through the following set of tasks:

1. triangulate features;
2. run bundle adjustment;
3. merge disjoint tracks;
4. clean up residual data.

These tasks are allowed to be more computationally intensive than the tracker's tasks, since the system does not depend on them in order to process the next frame.

5.1 Triangulating features

A feature that was observed in at least two keyframes within the same E-group G_E is triangulated and gets assigned a 3D location in G_E's frame of reference. When the tracker adds a new keyframe with a new observation of a feature f, a flag is set within f, and f is re-triangulated using all information when the mapper cycles through this step the next time.

5.2 Bundle adjustment

E-groups with at least three keyframes get passed through a standard bundle adjuster [20] that globally optimizes keyframe (i.e., camera) poses and feature positions in this group's frame of reference.

5.3 Merging disjoint tracks

As mentioned above, when tracking gets lost, the tracker immediately starts a new, independent *track*, rather than continuously trying to relocalize with respect to the existing map. In doing so, the tracker continues to collect data and 'stitch together' keyframes even though the spatial relation to the first map is unknown.

The algorithm that merges tracks is very similar to the keyframe-based recovery by Klein and Murray [15] (also used in [25, 37], among others): as observed by Eade and Drummond [9], recovery, loop closure, and (here) merging of tracks are effectively equivalent. The only difference lies in when the algorithm is executed and how its result is used.

Whenever a new keyframe is taken, the system stores a downsampled, blurred copy of the image (here: 80×60 pixels, blurred with a Gaussian with $\sigma = 1.5$px), dubbed small blurry image (SBI).

Merging of tracks is done as follows: the algorithm chooses a keyframe k_1 and computes the normalized cross-correlation (NCC) of its SBI with the SBI of all other keyframes. Keyframes on the same track as k_1 are omitted, as are keyframes to which a previous merge attempt failed. The keyframe k_2 with the highest NCC score is selected, and the SBIs of k_1 and k_2 are aligned to each other using inverse compositional image alignment [2] of an affine homography H_A. The features of k_1 are then projected into k_2 using H_A, and a regular "tracking" step (cf. Section 4) is executed.

If the tracking step fails, k_2 is "blacklisted" in k_1 as a failed attempt (so that the algorithm does not attempt the same combination again), and k_1 stores a timestamp of when this merge attempt occurred. The next time the algorithm tries to merge tracks, the keyframe that has not been chosen as k_1 the longest is chosen as k_1.

If the tracking step succeeds in estimating a model that is supported by a sufficient fraction of feature correspondences, the two

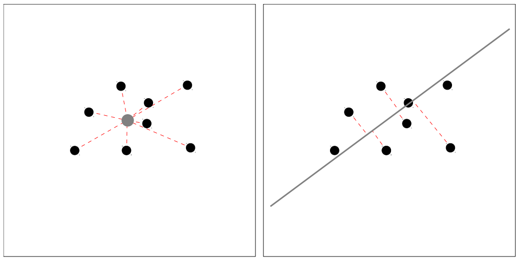


Figure 6: Intuition of why model selection with models of differing dimensions is hard: Trying to fit a 2D point and a 2D line to a set of noisy measurements. Both models have the same number of degrees of freedom (2), yet the dimensionality of the error is different (2 vs. 1), and the sum of errors (dashed red) is guaranteed to be smaller in the case of the line.

tracks are considered successfully merged. H-groups can be immediately merged by concatenating the homographies accordingly (cf. Section 4.1). To merge E-groups, one more tracking step is needed to generate feature observations common to both the existing groups and the newly inserted group that merges the two tracks. Then, they can be merged by adopting a common frame of reference analogous to the procedure when a new keyframe is inserted (Section 4.1).

The features are transferred to a new frame of reference by re-triangulating their positions using the observations from all keyframes in the merged group (Section 5.1).

The benefit of merging tracks is visualized in the case of panorama data in Fig. 5. In this particular case, the data of the merged tracks covers almost all areas that were visible during the input sequence, and is merged to an almost complete horizontal panorama. In comparison, while the “relocalization” strategy is able to recover and continue the first map several times, it discards imagery of large parts of the scene which were observed by the camera while the algorithm unsuccessfully tried to relocalize.

5.4 Cleaning up residual data

When a new track is started and thus a new keyframe is created (Section 4.2), but tracking gets lost again immediately after that, this new keyframe (and all features found in it) are not connected to any other data and provide very little useful information. To make sure that this kind of residual data does not accumulate, the mapping thread goes through the set of keyframes, identifies isolated keyframes and removes them.

6 Model Selection

If observed data may have arisen from several different models, one faces the problem of selecting the right model additionally to the common estimation of the model parameters. Model selection is a complex problem in particular if the models in question are of fundamentally different kind, as is the case here: a homography is a bijective 2D map, and thus the observed error between an estimated and measured feature location is two-dimensional, whereas the essential matrix maps a 2D point to a line in 2D, and thus the error is one-dimensional (perpendicular to the line). This is analogous to trying to determine if an observed set of 2D points can best be described by a point or by a line (Fig. 6). It becomes apparent that the sum of the fitting errors is not sufficient to select the model.

For this reason, several different metrics have been proposed. Here, we use the Bayesian formulation of the ‘Geometric Robust Information Criterion’ (GRIC) as derived by Torr [34], which is based on the Bayesian probability that a given model generated the observed data.

In this section, we use the same notation and variable names as Torr. The subscript m is used to indicate that a quantity is dependent on the model m. $\log(x)$ denotes the natural logarithm.

6.1 Torr’s GRIC score

The most generic formulation of Torr’s GRIC score is

$$\mathrm{GRIC}_m = -2\mathcal{L}_{\mathrm{MAP},m} + k_m \log n \tag{1}$$

(cf. [34] Eq. (46)) where k_m is the number of parameters of the model, n is the number of observations, and the maximum a-posteriori log likelihood $\mathcal{L}_{\mathrm{MAP},m}$ is derived according to the problem at hand. For the problem of model selection based on two-view correspondences with constrained search regions in the presence of outliers,

$$\mathrm{GRIC}_m = \sum_i \rho_2\left(\frac{e_{i,m}^2}{\sigma^2}\right) + \lambda_1 n d_m + k_m \log n + \text{const} \tag{2}$$

$$\text{with} \quad \rho_2(x) = \min\{x, T_m\} \tag{3}$$

$$T_m = 2\log\left(\frac{\gamma}{1-\gamma}\right) + (D - d_m)\lambda_1 \tag{4}$$

$$\lambda_1 = \log(S^2/(2\pi\sigma^2)) \tag{5}$$

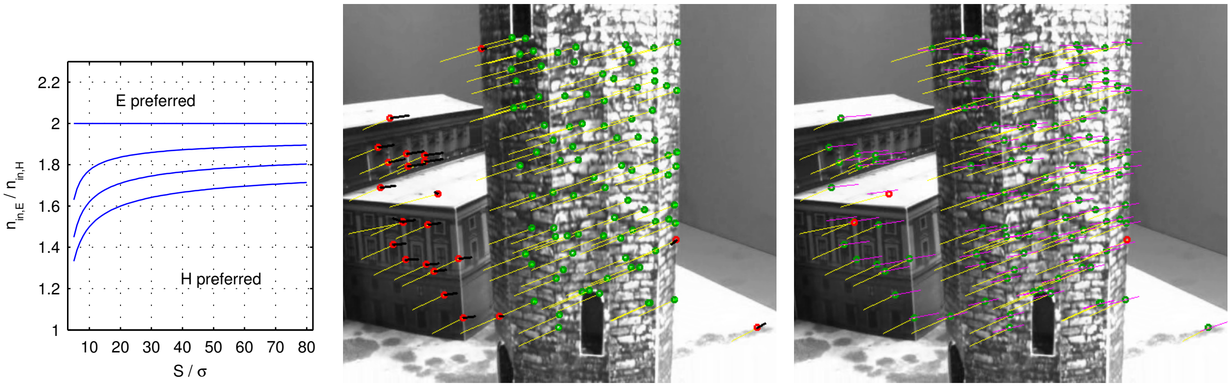
(cf. [34] Eqs. (48,49,18,19)) where d_m is the dimensionality of the model manifold (2 for homography, 3 for essential matrix), γ is the prior expectation that a random correspondence is an inlier, D is the dimensionality of each observation (here: a pair of 2D points, i.e., $D = 4$), $S \times S$ is the size of the constrained search region (and thus the volume from which outliers may appear), and finally σ is the standard deviation of the measurement error, which is assumed to be Gaussian. (It should be noted that despite the name, the GRIC score is a *cost* function, that is, lower score indicates better model fit.)

6.2 GRIC score for large active search regions

To arrive at a reasonably simple closed-form formula for the GRIC score, Torr [34] employs several approximations and assumptions. One of these assumptions — namely, that the depth disparity of inliers is distributed uniformly across the entire search region — is not fulfilled within reasonable bounds for applications with large active search regions. In this section, we will first explain the assumption, provide evidence that it would introduce a significant bias in our use case, and finally derive a more general form of the GRIC score which decreases this bias.

Among the quantities that are included in the calculation of the score are the volumes of the spaces in which certain measurements may occur. For example, assuming that each image has dimensions $L \times L$, then any individual 2D point measurement may occur on the volume $L \times L$. In an application that uses constrained search regions to establish correspondences (like ours), an arbitrary correspondence (i.e., pair of points) may occur on the volume $v = L \times L \times S \times S$, where S is the size of the search region. However, a correspondence that is an *inlier* to a (given) bijective 2D map such as a homography is distributed on the volume $c_H = L \times L$ (because the second point is uniquely fixed by the bijective map). If the model is a non-bijective relation such as an essential or fundamental matrix, which constrain a point to lie on a line but do not fix its position along the line, the volume becomes $c_E = L \times L \times R$, where R is the range of the disparity along which the feature match is expected to occur.

R is limited to $[0, S]$, as otherwise the match will not be found and the correspondence will become an outlier. For simplicity, Torr sets $R = S$, which causes several terms to cancel each other in the derivation of the final GRIC score (Eq. (2)). However, this assumes that the inliers are uniformly distributed in disparity in the entire



(a) preferred model w.r.t. inlier ratio (b) tracking with H (c) tracking with E

Figure 7: (a) Preferred models (according to GRIC score) as a function of the inlier ratio $n_{\text{in},E}/n_{\text{in},H}$ (expressed as a function of S/σ and $\gamma = \{0.5, 0.6, 0.7, 0.8\}$ (lines from top to bottom)). Below the respective line, the GRIC score will prefer H, even if E produces more inliers. (b) Tracked feature correspondences in scene classified by H, (c) the same correspondences classified by E. Inliers are shown in green, outliers in red. Even though E models all correct correspondences correctly and H misclassifies 19 correct correspondences as outliers, $\text{GRIC}_H = 1633.83 < \text{GRIC}_E = 1919.53$ and the algorithm would chose H over E. (Here: $n = 112$, $S = 40$, $\sigma = 1$, $\gamma = 0.5$.) The more general score derived in Section 6.2 decreases this problem.

search region.[2] For large S, this is not the case: The search region is large because we want tracking to be robust to fast camera movement (including camera rotations), but we expect the (2D) movement of all features to be strongly correlated; the range of disparity is still likely to be only a few pixels. Hence, the log likelihood for E is decreased because it does not match the model well, creating a significant bias towards selecting H.

Evidence that this bias is significant and can lead to incorrect model selection. Assume that a traveling camera observes a scene which contains a near-planar object on which many (but not all) correctly and accurately tracked feature correspondences lie. Those correspondences will be inliers to both E and H, with $e_i \approx 0$. Assume that there are further accurately tracked features on other surfaces (will be inliers to E, but outliers for H), and, optionally, several spurious correspondences. Following Eq. (2), the GRIC score for either model m then is

$$\text{GRIC}_m \approx (n - n_{\text{in},m}) \cdot T_m + \lambda_1 n d_m + k_m \log n + \text{const} \tag{6}$$

where $n_{\text{in},m}$ is the number of inliers for model m. We are interested in the threshold when the scores for the essential matrix E and the homography H are equal:

$$\begin{aligned} (n - n_{\text{in},E}) \cdot T_E + \lambda_1 n d_E + k_E \log n &= \\ (n - n_{\text{in},H}) \cdot T_H + \lambda_1 n d_H + k_H \log n & \end{aligned} \tag{7}$$

Assuming that $(k_E - k_H) \log n << (n - n_{\text{in},m}) \cdot T_m$:

$$\Rightarrow \quad \frac{n_{\text{in},E}}{n_{\text{in},H}} = \frac{2 \log \frac{\gamma}{1-\gamma} + (D - d_H) \cdot \lambda_1}{2 \log \frac{\gamma}{1-\gamma} + (D - d_E) \cdot \lambda_1} \tag{8}$$

With λ_1 as given in Eq. (5), $D = 4$, $d_E = 3$ and $d_H = 2$:

$$\Rightarrow \quad \frac{n_{\text{in},E}}{n_{\text{in},H}} = \frac{2 \log \frac{\gamma}{1-\gamma} + 2 \log \left(\frac{S^2}{2\pi\sigma^2} \right)}{2 \log \frac{\gamma}{1-\gamma} + 1 \log \left(\frac{S^2}{2\pi\sigma^2} \right)} \tag{9}$$

This ratio is plotted for a range of reasonable values for S, σ, γ in Fig. 7(a). It can be seen that the range of $n_{\text{in},E}/n_{\text{in},H}$ for which the algorithm will prefer H (i.e., the area below the line) increases with the search region S, up to the point that the algorithm will, for very large S and γ close to 0.5, prefer H even if E produces almost twice as many inliers ($n_{\text{in},E}/n_{\text{in},H} = 2$). Fig. 7(b,c) illustrates a real-world case in which exactly this is the case: $\text{GRIC}_H < \text{GRIC}_E$ even though E models all correspondences correctly (including rejection of three spurious correspondences), while H discards a sizable set of correct correspondences as outliers.

Generalized GRIC score. Thus, we re-derive the GRIC score for the general case of R independent of S. The full derivation is given in Appendix A. Our final formula for the GRIC score is

$$\begin{aligned} \text{GRIC}_m = \sum_i \rho_2 \left(\frac{e_{i,m}^2}{\sigma^2} \right) + n \left((D - d_m) \log 2\pi\sigma^2 + 2 \log \frac{c_m}{\gamma} \right) \\ + k_m \log n \end{aligned} \tag{10}$$

with $\rho_2(x) = \min\{x, T_m\}$, and

$$T_m = 2 \log \left(\frac{\gamma}{1-\gamma} \cdot \frac{v}{c_m} \right) - (D - d_m) \log 2\pi\sigma^2 \tag{11}$$

We show in Appendix B that this is equivalent to Torr's formula ([34] Eq. (48)) for the special case $R = S$.

7 Evaluation

We implemented our system prototype in C++, making use of the OpenCV[3], TooN[4] and libCVD[5] libraries. Our system runs in real-time on a commodity PC without specific optimizations. Typical timings are presented in Fig. 8. They are, of course, strongly dependent on the used hardware and parameter configuration (in particular, number of keypoints per frame) and presented here only as a reference point.

At this point, our prototype is not optimized to run in real-time on a mobile device (such as light-weight tablet or smartphone). However, the most expensive parts (in particular, the tracking with NCC-based matching, cf. Fig. 8) are computationally similar to T&M

[2] Torr [34] explicitly warns that "care should be taken to rederive [this quantity] according to the exact distribution [...] in different scenarios."

[3] http://opencv.willowgarage.com/

[4] http://www.edwardrosten.com/cvd/toon.html

[5] http://www.edwardrosten.com/cvd/

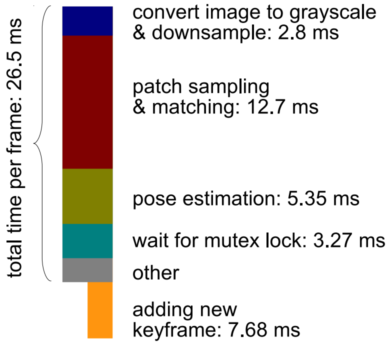


Figure 8: Breakdown of timings in tracking thread. These times were taken on a commodity PC with Intel i7 Core and 4 GB RAM running Ubuntu 10.10, without specific optimizations or the use of a GPU. All times are averaged per frame over the entire sequence. Since creating a new keyframe is executed at most every 20th frame, its contribution to the average time is minimal. The time spent waiting for the mutex lock could be significantly reduced by implementing more fine-grained mutex locks in the mapping thread.

systems that have been shown to operate in real time on such devices [16, 36] several years ago. Thus, we argue that with appropriate algorithmic and device-specific optimizations [16], running an implementation of our concept on a mobile device is feasible.

We tested our system on a variety of video sequences, including videos of a panorama dataset by Coffin et al. [6] [6], sequences from the "City of Sights" repository [12] [7], as well as further self-recorded videos, using models from the "City of Sights" as backdrop. Results from those videos are presented in Figs. 1, 5, and 9. While our system has a number of parameters which were tuned by hand, we determined one set of parameters and kept it constant for all results presented here.

A quantitative comparison of our prototype proves difficult, since to our knowledge, there is no other system that is able to create panoramas as well as 3D structures (such as illustrated in Fig. 1 top right) in real time. Qualitatively, we can state that the frame-to-frame tracking is very robust. In panorama mode, our system is able to stitch together panoramas with high accuracy. Merging of tracks works very robustly, with very few observed false positives.

However, our implementation of tracking during general motion and 3D reconstruction is still susceptible to common SLAM pitfalls such as adversarial camera motion and fails for unfavorable sets of keyframes. As correctly noted by Newcombe et al. [22], the criteria used to insert keyframes in feature-based systems, ours included, are heuristics only. Therefore, the robustness of the 3D mapping part can currently not compete with the robustness demonstrated by systems such as PTAM [14] or DTAM [22]. Here, especially the design choice of waiving the requirement of a separate initialization step (which is used to bootstrap the map in [14, 22]) appears to somewhat compromise model coherency.

The model selection with the GRIC score appears to be robust against outliers and image noise. For example, for 125 keyframe pairs that were evaluated during processing of the rotation-only video sequence to Fig. 5, the algorithm chose correct model (H) 123 times. One concern is the occurrence of somewhat ambiguous situations, such as the presence of a dominant planar object, or rotations with very small shifts of the optical center. In these cases, the ratio of the GRIC scores is sensitive to certain parameters (such as R and γ) and may flip-flop between the two models from one frame to the next. It should be noted that this is not a flaw of the GRIC score per se — the score is as ambiguous as the observed camera motion — and that tracking and mapping does, indeed, continue successfully; however the collected model in this case is fragmented in several (connected, but nevertheless separate) maps (as can be observed in Fig. 5 bottom), which is undesirable.

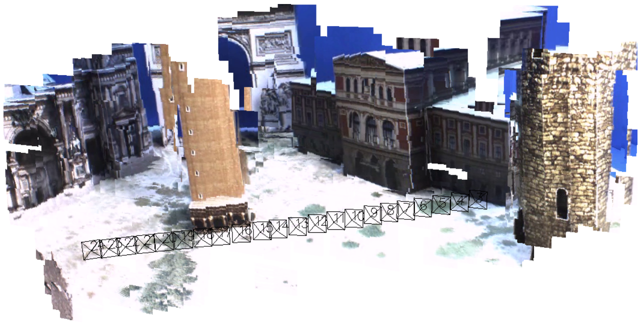


Figure 9: Our system acting as a SLAM system, reconstructing the scene in 3D and recovering the camera trajectory fully automatically. The input sequence for this model stems from the "City of Sights" repository [12] and was recorded by a robot arm. Each reconstructed 3D feature position is visualized as a small image patch (sampled from the frame in which it was first observed).

8 Conclusions

We have presented an approach for real-time tracking and mapping that supports both general (parallax-inducing) and degenerate (rotation-only) camera motions in 3D environments. Our design paradigm was to make use of all data that can be casually collected, and to not require any particular assistance by the user (such as a separate initialization step, or particular types of camera motion).

Our system is able to track and map through motion sequences that neither conventional SLAM systems nor panorama systems can process. Tracking is highly accurate, panorama stitching is fully automatic and seamless. Depending on the video sequence, the strategy of starting a new track and later merging separate tracks (instead of trying to relocalize continuously with respect to the first map) significantly increases the amount of data present in the final environment model.

We acknowledge that tracking robustness during general motion and coherency of the produced 3D model is, at this point, not comparable to systems such as PTAM [14] or DTAM [22]. While coherency of the 3D model was not the main focus of our work, it is clear that the robustness needs to be increased in order to turn our prototype into a system that supports unconstrained user motion and can be used as the basis for an AR application. In particular, more advanced map feature management including filtering of outliers, more sophisticated keyframe selection, estimation of feature normals [21, 38] and thus more sophisticated feature predictions may be beneficial to integrate. In doing so, one challenge is that the earlier steps in the pipeline should be agnostic about the current type of model, as otherwise many redundant computations may be executed. Due of this, existing strategies cannot necessarily be applied one to one.

Aside from improving the 3D reconstruction, there are several other areas with open research questions. With respect to the GRIC score, while its model recommendation is arguably the optimal choice given a particular set of parameters and one pair of frames, it is nontrivial to provide optimal parameters for any kind of data input. Further, given a larger set of frames, coherency across the chosen models may be preferable over selecting the locally optimal model for any pair, in order to minimize "fragmentation" of the model into too many (connected) sub-maps of different type. Finally, while we currently select only one model for the entire frame

[6] http://tracking.mat.ucsb.edu

[7] http://cityofsights.icg.tugraz.at

(which is, theoretically, the correct thing to do, since the motion refers to the camera an thus to the entire frame), there may be cases especially in outdoor AR in which the foreground exhibits enough parallax to be modeled in 3D, while the background exhibits little parallax and might benefit from the stable, dense mapping that homographies offer. This leads to an interesting problem in which image segmentation and scene modeling interact.

Further, it remains an open question how a model that consists of a mixture of structural data and (partial) panoramas can best be visualized, presented to, and navigated by the user. This is not a concern if the model is used only as an anchor for AR annotations (in which case the user never actually sees the model, but only the annotations fused with his/her view onto the real world). However, if the model is to be used in Virtual Reality as well (for example, to allow a spatially remote user to view the scene), the model itself needs to be presented and navigated. This works very well in the case of panoramic mapping, where the emerging model (i.e., the stitched panorama) is easy to interpret and browse. It is inherently more challenging in the case of 3D data (especially if the model is incomplete, so that the viewpoints for which useful views can be rendered are restricted), and, to our knowledge, a completely open research question for the case of live, incomplete data that consists of mixtures of structural and panoramic data. By building on large data collections and offline reconstructions, interesting viewing modalities for mixed data like this have emerged from the Structure-from-Motion community [32].

Appendix

A Derivation of the Generalized GRIC Score

The general Bayesian formulation of the GRIC score (cf. [34] Eq. (46)) is

$$\mathrm{GRIC}_m = -2\mathcal{L}_{\mathrm{MAP},m} + k_m \log n \tag{12}$$

with (cf. [34] Eq. (15))

$$\mathcal{L}_{\mathrm{MAP},m} = \sum_i \log(\gamma_i \cdot p_{\mathrm{in}} + (1-\gamma_i)\cdot p_{\mathrm{out}}) \tag{13}$$

$$p_{\mathrm{in}} = \frac{\sqrt{2\pi\sigma^2}^{d_m - D}}{c_m} \cdot \exp\left(-\frac{e_{i,m}^2}{2\sigma^2}\right) \tag{14}$$

$$p_{\mathrm{out}} = 1/v \tag{15}$$

where $\gamma_i \in \{1,0\}$ indicates if correspondence i is an inlier. Denote with $\gamma = \{P(\gamma_i = 1)\}$ ([34] Eq. (8)) the prior expectation of seeing an inlier, and maximize over γ_i:

$$\begin{aligned}\Rightarrow\ \mathrm{GRIC}_m &= -2\sum_i \log(\max\{\gamma\cdot p_{\mathrm{in}}; (1-\gamma)p_{\mathrm{out}}\}) + k_m \log n \\ &= \sum_i \min\{\underbrace{-2\log(\gamma\cdot p_{\mathrm{in}})}_{(*)}; \underbrace{-2\log((1-\gamma)p_{\mathrm{out}})}_{(**)}\}) + k_m \log n\end{aligned} \tag{16}$$

$$(*) = -2\log\left(\gamma \frac{\sqrt{2\pi\sigma^2}^{d_m - D}}{c_m} \cdot \exp\left(-\frac{e_{i,m}^2}{2\sigma^2}\right)\right) \tag{17}$$

$$= \frac{e_{i,m}^2}{\sigma^2} + (D - d_m)\log 2\pi\sigma^2 + 2\log\frac{c_m}{\gamma} \tag{18}$$

$$(**) = -2\log\left(\frac{1-\gamma}{v}\right) \tag{19}$$

$$= T_m + (D - d_m)\log 2\pi\sigma^2 + 2\log\frac{c_m}{\gamma} \tag{20}$$

$$\text{with}\quad T_m := 2\log\left(\frac{\gamma}{1-\gamma}\cdot\frac{v}{c_m}\right) - (D - d_m)\log 2\pi\sigma^2 \ \equiv (11) \tag{21}$$

$$\begin{aligned}\Rightarrow\ \mathrm{GRIC}_m = \sum_i \Bigg(\rho_2\left(\frac{e_{i,m}^2}{\sigma^2}\right) &+ (D - d_m)\log 2\pi\sigma^2 \\ &+ 2\log\frac{c_m}{\gamma}\Bigg) + k_m \log n \quad \equiv (10)\end{aligned} \tag{22}$$

B Proof that the Generalized GRIC Score is Equivalent to Torr's Formula for R=S

For the special case of $R = S$, note that $U := (v/c_m)^{\frac{1}{D-d_m}} = S$ and $U' := c_m/U^{d_m} = L^2/S^2$ (i.e., both are independent of the model m) for all models considered here. Starting with Eq. (21),

$$T_m = 2\log\left(\frac{\gamma}{1-\gamma}\right) + 2\log\left(U^{D-d_m}\right) - (D - d_m)\log 2\pi\sigma^2 \tag{23}$$

$$= 2\log\left(\frac{\gamma}{1-\gamma}\right) + (D - d_m)\log\left(\frac{U^2}{2\pi\sigma^2}\right) \tag{24}$$

which is equivalent to Torr's definition of T ([34] Eq. (18)) with $\lambda_1 := \log(U^2/(2\pi\sigma^2))$. Further, starting with Eq. (22),

$$\mathrm{GRIC}_m = \sum_i \rho_2\left(\frac{e_{i,m}^2}{\sigma^2}\right) + A_m + k_m \log n \tag{25}$$

$$\text{with } A_m = n\left((D - d_m)\log 2\pi\sigma^2 + 2\log\frac{c_m}{\gamma}\right) \tag{26}$$

$$= n\left(\lambda_1 d_m + D\log 2\pi\sigma^2 + 2\log\frac{c_m}{\gamma} - d_m \log U^2\right) \tag{27}$$

$$= \lambda_1 n d_m + \log\left(\frac{(2\pi\sigma^2)^D}{\gamma^2}\cdot\left(\frac{c_m}{U^{d_m}}\right)^2\right) \tag{28}$$

$$= \lambda_1 n d_m + \log\left(\frac{(2\pi\sigma^2)^D}{\gamma^2}\cdot U'^2\right) \tag{29}$$

$$= \lambda_1 n d_m + \mathrm{const} \tag{30}$$

$$\begin{aligned}\Rightarrow\quad \mathrm{GRIC}_m &= \sum_i \rho_2\left(\frac{e_{i,m}^2}{\sigma^2}\right) + \lambda_1 n d_m + k_m \log n + \mathrm{const} \\ &\equiv [34]\text{ Eq. (48)} \qquad \Box\end{aligned}$$

(Note that all terms that are not dependent on m are considered constant in this context, even if they change from frame to frame.)

Acknowledgements

This work was supported by a fellowship from UCSB, NSF CAREER grant IIS-0747520 and ONR grant N00014-09-1-1113.

References

[1] H. Akaike. A new look at the statistical model identification. *IEEE Trans. Automatic Control*, 19(6):716–723, 1974.

[2] S. Baker and I. Matthews. Lucas-Kanade 20 years on: A unifying framework: Part 1. Technical Report CMU-RI-TR-02-16, Robotics Institute, Carnegie Mellon University, Pittsburgh, July 2002.

[3] R. O. Castle, G. Klein, and D. W. Murray. Video-rate localization in multiple maps for wearable augmented reality. In *Proc 12th IEEE Intl. Symp. on Wearable Computers*, pp. 15–22, 2008.

[4] J. Civera, A. Davison, and J. Montiel. Interacting multiple model monocular SLAM. In *IEEE Intl. Conference on Robotics and Automation (ICRA)*, pp. 3704–3709. 2008.

[5] J. Civera, A. Davison, and J. Montiel. Inverse depth parametrization for monocular SLAM. *IEEE Trans. Robotics*, 24(5):932–945, 2008.

[6] C. Coffin, J. Ventura, and T. Höllerer. A repository for the evaluation of image-based orientation tracking solutions. In *Proc. 2nd Intl. Workshop on AR/MR Registration, Tracking and Benchmarking (TrakMark 2011)*, in conjunction with ISMAR 2011.

[7] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, 2007.

[8] S. DiVerdi, J. Wither, and T. Höllerer. All around the map: Online spherical panorama construction. *Computers & Graphics*, 33(1):73–84, 2009.

[9] E. Eade and T. Drummond. Unified loop closing and recovery for real time monocular SLAM. In *Proc. British Machine Vision Conference (BMVC)*, 2008.

[10] S. Gauglitz, L. Foschini, M. Turk, and T. Höllerer. Efficiently selecting spatially distributed keypoints for visual tracking. In *Proc. IEEE Intl. Conference on Image Processing (ICIP)*, 2011.

[11] S. Gauglitz, C. Lee, M. Turk, and T. Höllerer. Integrating the physical environment into mobile remote collaboration. In *Proc. ACM SIGCHI's Intl. Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI)*, 2012.

[12] L. Gruber, S. Gauglitz, J. Ventura, S. Zollmann, M. Huber, M. Schlegel, G. Klinker, D. Schmalstieg, and T. Höllerer. The City of Sights: Design, construction, and measurement of an augmented reality stage set. In *Proc. IEEE Intl. Symposium on Mixed and Augmented Reality (ISMAR'10)*, pp. 157–163, Seoul, Korea, Oct. 13-16 2010.

[13] L. Gruber, S. Zollmann, D. Wagner, D. Schmalstieg, and T. Höllerer. Optimization of target objects for natural feature tracking. In *Proc. 20th Intl. Conference on Pattern Recognition (ICPR)*, pp. 3607–3610, Istanbul, August 2010.

[14] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Proc. 6th IEEE and ACM Intl. Symposium on Mixed and Augmented Reality*, Nara, Japan, November 2007.

[15] G. Klein and D. Murray. Improving the agility of keyframe-based SLAM. In *Proc. 10th European Conference on Computer Vision*, pp. 802–815, Marseille, France, Oct. 2008.

[16] G. Klein and D. Murray. Parallel tracking and mapping on a camera phone. In *Proc. 8th IEEE Intl. Symposium on Mixed and Augmented Reality*, pp. 83–86, Oct. 2009.

[17] M. Klopschitz, A. Irschara, G. Reitmayr, and D. Schmalstieg. Robust incremental structure from motion. In *Proc. Intl. Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*, vol. 2, 2010.

[18] T. Lemaire, C. Berger, I.-K. Jung, and S. Lacroix. Vision-based SLAM: Stereo and monocular approaches. *Intl. Journal of Computer Vision*, 74:343–364, 2007.

[19] S. Lieberknecht, A. Huber, S. Ilic, and S. Benhimane. RGB-D camera-based parallel tracking and meshing. In *IEEE Intl. Symposium on Mixed and Augmented Reality (ISMAR) 2011*, pp. 147 –155, Oct. 2011.

[20] M. A. Lourakis and A. Argyros. SBA: A software package for generic sparse bundle adjustment. *ACM Trans. Math. Software*, 36(1):1–30, 2009.

[21] N. Molton, A. Davison, and I. Reid. Locally planar patch features for real-time structure from motion. In *Proc. 15th British Machine Vision Conference (BMVC)*, 2004.

[22] R. Newcombe, S. Lovegrove, and A. Davison. Dtam: Dense tracking and mapping in real-time. In *IEEE Intl. Conference on Computer Vision (ICCV)*, pp. 2320–2327, 2011.

[23] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. In *Proc. IEEE Intl. Symposium on Mixed and Augmented Reality (ISMAR)*, 2011.

[24] D. Nistér. An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(6):756–770, 2004.

[25] C. Pirchheim and G. Reitmayr. Homography-based planar mapping and tracking for mobile phones. In *Proc. 10th IEEE Intl. Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 27 –36, oct. 2011.

[26] M. Pollefeys, F. Verbiest, and L. Van Gool. Surviving dominant planes in uncalibrated structure and motion recovery. In *Proc. European Conference on Computer Vision (ECCV)*, pp. 613–614, 2002.

[27] J. Repko and M. Pollefeys. 3D models from extended uncalibrated video sequences: Addressing key-frame selection and projective drift. In *Proc. 5th Intl. Conference on 3-D Digital Imaging and Modeling (3DIM)*, pp. 150–157, 2005.

[28] J. Rissanen. Modeling by shortest data description. *Automatica*, 14 (5):465–471, 1978.

[29] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *Proc. IEEE European Conference on Computer Vision (ECCV)*, vol. 1, pp. 430–443, May 2006.

[30] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461– 464, 1978.

[31] N. Snavely. *Scene Reconstruction and Visualization from Internet Photo Collections*. PhD thesis, University of Washington, 2008.

[32] N. Snavely, S. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3D. In *ACM Trans. Graphics (TOG)*, vol. 25, pp. 835–846, 2006.

[33] H. Strasdat, J. Montiel, and A. Davison. Real-time monocular SLAM: Why filter? In *Proc. IEEE Intl. Conference on Robotics and Automation (ICRA)*, pp. 2657–2664, 2010.

[34] P. Torr. Bayesian model estimation and selection for epipolar geometry and generic manifold fitting. *Intl. Journal of Computer Vision*, 50 (1):35–61, 2002.

[35] P. H. Torr, A. W. Fitzgibbon, and A. Zisserman. The problem of degeneracy in structure and motion recovery from uncalibrated image sequences. *Intl. Journal of Computer Vision*, 32:27–44, 1999.

[36] D. Wagner, D. Schmalstieg, and H. Bischof. Multiple target detection and tracking with guaranteed framerates on mobile phones. In *Proc. 8th IEEE Intl. Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 57–64, Oct. 2009.

[37] D. Wagner, A. Mulloni, T. Langlotz, and D. Schmalstieg. Real-time panoramic mapping and tracking on mobile phones. In *Proc. IEEE Virtual Reality (VR)*, March 2010.

[38] H. Wuest, F. Wientapper, and D. Stricker. Acquisition of high quality planar patch features. *Advances in Visual Computing*, pp. 530–539, 2008.

[39] Z. Zhang and O. Faugeras. Estimation of displacements from two 3D frames obtained from stereo. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(12):1141 –1156, dec 1992.