High-Precision Camera Localization in Scenes with Repetitive Patterns

XIAOBAI LIU, Department of Computer Science, San Diego State University
QIAN XU, XreLab Inc.
YADONG MU, Institute of Computer Science and Technology
JIADI YANG, University of California
LIANG LIN, School of Advanced Computing, Sun Yat-Sen University
SHUICHENG YAN, Qihoo/360 Inc., China; National University of Singapore, Singapore

This article presents a high-precision multi-modal approach for localizing moving cameras with monocular videos, which has wide potentials in many intelligent applications, including robotics, autonomous vehicles, and so on. Existing visual odometry methods often suffer from symmetric or repetitive scene patterns, e.g., windows on buildings or parking stalls. To address this issue, we introduce a robust camera localization method that contributes in two aspects. First, we formulate feature tracking, the critical step of visual odometry, as a hierarchical min-cost network flow optimization task, and we regularize the formula with flow constraints, cross-scale consistencies, and motion heuristics. The proposed regularized formula is capable of adaptively selecting distinctive features or feature combinations, which is more effective than traditional methods that detect and group repetitive patterns in a separate step. Second, we develop a joint formula for integrating dense visual odometry and sparse GPS readings in a common reference coordinate. The fusion process is guided with high-order statistics knowledge to suppress the impacts of noises, clusters, and model drifting. We evaluate the proposed camera localization method on both public video datasets and a newly created dataset that includes scenes full of repetitive patterns. Results with comparisons show that our method can achieve comparable performance to state-of-the-art methods and is particularly effective for addressing repetitive pattern issues.

CCS Concepts: • **Computing methodologies;** • **Applied computing;**

Additional Key Words and Phrases: Visual odometry, feature matching, flow optimization

ACM Reference format:
Xiaobai Liu, Qian Xu, Yadong Mu, Jiadi Yang, Liang Lin, and Shuicheng Yan. 2018. High-Precision Camera Localization in Scenes with Repetitive Patterns. *ACM Trans. Intell. Syst. Technol.* 9, 6, Article 66 (November 2018), 21 pages.
https://doi.org/10.1145/3226111

The first two authors contributed equally to this article. Jiadi Yang worked on this project when he was studying at UCLA. Xiaobai Liu was supported by the DARPA SIMPLEX program (Grant No. 58723A), National Science Foundation (Grant No. 1657600), ONR (Grant No. N00014-17-1-2867), and San Diego State University Presidential Leadership Funds.
Authors' addresses: X. Liu, Department of Computer Science, San Diego State University, San Diego, CA, USA; email: xiaobai.liu@sdsu.edu; Q. Xu, XreLab Inc., San Diego, CA, USA; email: qxu@xrelab.com; Y. Mu, Institute of Computer Science and Technology, Beijing, China; email: muyadong@gmail.com; J. Yang, Google Inc., San Jose, California, USA; email: jyang@cs.ucla.edu; L. Lin, School of Advanced Computing, Sun Yat-Sen University, Guangzhou, China; email: linliang@ieee.org; S. Yan, Qihoo/360 Inc., China, National University of Singapore, Singapore; email: eleyans@nus.edu.sg.

2157-6904/2018/11-ART66 $15.00
https://doi.org/10.1145/3226111

1 INTRODUCTION

Accurately localizing moving cameras from monocular videos [63] is capable of providing scene context for high-level video understanding tasks [30], e.g., behavior analysis, action recognition, and so on, and has wide potentials in many intelligent systems, e.g., robotics, autonomous vehicles, and intelligent helicopters. While GPS devices are popular in these applications, they can only provide camera positions of moderate accuracies [55] and are sensitive to environment changes. In addition, GPS locations are only sparsely available, which is not applicable for time-critical applications, e.g., security systems or autonomous driving systems. In the past literature, vision-based methods have been proposed to estimate camera locations (i.e., visual odometry), to enhance the accuracies and robustness of GPS systems. The basic idea is to build correspondences across consecutive video frames, e.g., using optical flow algorithm [8, 26, 31, 45], and further estimate relative camera movements [43, 59]. These methods have achieved promising results in generic types of scenes, as demonstrated by the odometry benchmark [19]. However, the visual odometry problem still remains unresolved in complicated scenes that are full of repetitive or symmetric patterns, since these patterns bring large ambiguities for feature correspondence as well the whole visual odometry pipeline.

To address the repetitive pattern issues, we propose a robust multi-modal method for localizing moving cameras using monocular videos. The objective of our approach is to estimate camera poses (translation and orientations) at each time-step. Figure 1 shows a few video frames (top) captured by a moving camera, and the scene map (bottom) overlaid with the estimated camera trajectories. The scene is full of repetitive patterns, e.g., parking stalls, trees, buildings, and so on, which might challenge most existing camera localization methods. In contrast, our approach is able to robustly localize cameras in world coordinate with high accuracies.

1.1 Overview of Our Approach

Our goal is to estimate visual odometry of moving cameras [20] from monocular video sequences, and integrate with sparsely available GPS readings to register the cameras in world coordinate. We consider urban scenes that are full of repetitive patterns and propose a multi-modal visual odometry method, which comprises of two important components.

First, we propose a simple and effective energy minimization approach for building feature correspondences across consecutive video frames [34, 60]. Feature tracking is the most important step of visual odometry and in the past literature, most tracking methods [3, 20, 59] employ appearance (color, gradients) or motion information. These cues are not distinctive enough for the scenes that include many repeated or symmetric patterns. Figure 1 (top) shows an exemplar scenario with repetitive patterns. To address this issue, an effective solution is to first detect repetitive patterns and then group spatially adjacent patterns to form distinct structures with large support to enable robust feature matching [16, 24, 50]. Such a strategy is apparently restricted to the quality of repetitive pattern detection that still remains unsolved in wild settings. In this work, we will develop a robust feature tracking method that does not require the separate step of detecting repetitive patterns.

The key idea of our method is to formulate the feature tracking task as a hierarchical min-cost network flow problem. We recursively group individual features in the same video frame to form a hierarchical representation, and collectively match features or feature combinations of different levels across consecutive video frames. We develop a global optimization strategy to adaptively select feature detections that are distinctive against local surroundings, and we feature connections that are consistent across the hierarchy. This joint selection process is regularized with both *flow constraints*, e.g., that two features cannot occupy the same feature location, and *motion heuristics*

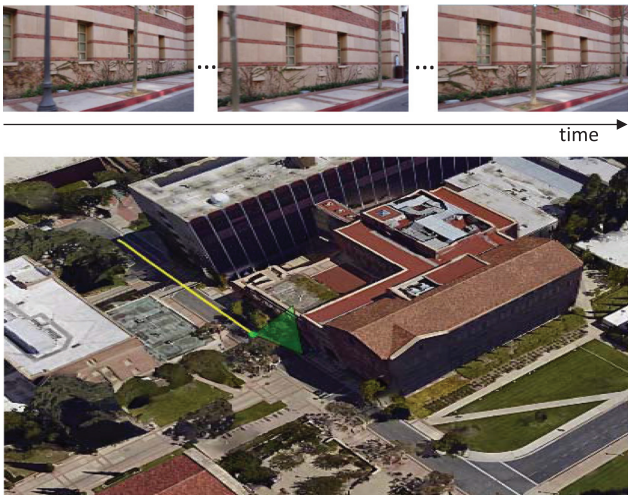

Fig. 1. Camera localization in scenes with repetitive patterns. Top: three video frames from a monocular video captured by a moving camera. The scene is full of repetitive or symmetric patterns (e.g., windows, facades). Bottom: the scene map overlaid with the estimated camera trajectory.

that characterize moving trajectories of cameras. The motion heuristics are used to encourage individual features to move with constant speeds, or to enforce that adjacent features share similar movement patterns. The above objectives and constraints are integrated together using a quadratic integer program, which can be efficiently solved with appropriate relaxations [28]. At each time-step, our method will process multiple video frames to explore the long-range inter-correlations between feature paths. We will demonstrate that the proposed formula can significantly mitigate the effects of repetitive patterns and many other challenges, e.g., lighting changes, low resolution, and so on.

Second, we develop a multi-modal approach that can integrate both GPS readings and visual odometry to further boost localization accuracy and robustness. As aforementioned, GPS locations are usually noisy, sparse, and with large errors (up to 10m on average). In contrast, visual odometry are continuous, but suffer from drifting issues especially for scenes with repetitive and symmetric patterns. Therefore, it is natural to integrate visual odometry and GPS readings in practical camera localization systems. However, point-wise fusion of GPS readings and VO measurements is fragile, because (i) GPS readings are provided in the absolute world coordinate system and the VO points are in metric space, (ii) GPS readings are only available for discrete time-steps, and (iii) both data modalities include tremendous errors that will affect the accuracies of data fusion. In this work, we propose to represent a camera trajectory with a polynomial piece-wise smoothing function, and introduce a multi-modal fitting method to fuse discrete GPS readings and continuous VO points. The innovative idea of our method is a joint solution that simultaneously solves data fusion and data interpretation. We also integrate high-order statistical knowledge over GPS readings, which can further improve system robustness and precision.

For test and evaluation purposes, we apply the proposed camera localization approach over challenging video sequences that include many repetitive patterns, e.g., parking lot, university campus. Both qualitative and quantitative results showed that the proposed feature tracking method significantly improved the standard visual odometry pipeline and achieved state-of-the-art performance. The proposed multi-modal fusion formula can further improve localization performance. We will release the collected video sequences for public usage to foster research in this direction. We also

apply the proposed method over the KITTI Odometry benchmark [19] and another video dataset, Hague [14], to demonstrate its generalization capabilities while dealing with general scenarios.

1.2 Relationships to Previous Works

This work is closely related to five research streams in the areas of computer vision and video processing.

Matching repetitive patterns across images has been studied extensively in the past literature. Most existing methods [2, 16, 24, 27, 40, 50] follow the same pipeline: detecting repetitive patterns in an image, clustering patterns to form large chucks of patterns that are more distinctive, and matching clusters of features into the other image. Technical details of individual steps vary in different methods. For example, Ha et al. [24] clustered features based on feature appearances, Fan et al. [16] proposed to collect distinct pairs of features, Torii et al. [50] presented a rectified bag-of-words descriptors for representing clusters, and Doubek et al. [40] introduced a shift-invariant descriptor for clustering repetitive features using descriptors of 2D spatial layout. The above methods achieved impressive results for images with repetitive patterns. However, they are sensitive to the choices of individual steps, which restrict its potentials in real applications. In this work, we aim to shift this practice by developing a global optimal strategy that can adaptively select both distinctive features and feature connections at multiple resolutions and allow us to skip the tedious step of detecting repetitive patterns.

Visual Odometry (VO) or relative camera motion can be readily estimated from videos. The key step of visual odometry is to associate features (e.g., SURF [4] or SIFT [31]) across consecutive frames, i.e., feature correspondences. In the past decade, there has been significant improvements in visual odometry along with the wide deployments of real applications, e.g., self-driving cars. In particular, Zhang and Singh [59] proposed to extract visual odometry from lidar sensors in real-time and achieved state-of-the-arts results on public benchmark KITTI [19]. Crivelli et al. [10] developed a theoretical framework for constructing dense point trajectories from optical flow fields. Mohamed et al. [37] proposed a robust texture descriptor and used it to develop an illumination-robust constancy for solving feature flow. Heas et al. [25] developed a Bayesian solution for selecting optimal models and hyper-parameters for optical-flow estimation. Botella et al. [6] introduced a novel customizable architecture of optical flow based on reconfigurable hardware, and further developed a bio-inspired system [7] to reduce computational complexity of optical-flow algorithms to enable real-time deployments. Badino et al. [3] introduce a stereo system that integrates feature correspondences between multiple video frames to improve measurement accuracies. Geiger et al. [20] and Song et al. [47] applied structure-from-motion techniques over monocular videos to estimate camera motion. In this work, we focus on monocular videos with geo-tags and aim to develop a general solution to the repetitive pattern issues. Our efforts include a dataset of videos captured in scenes full of repetitive patterns, which is the first odometry benchmark in its catalog, and a practical approach for associating repetitive patterns over time and registering camera in world coordinate.

Visual odometry can also be solved by SLAM (Simultaneous Localization and Mapping) methods, which aim to reconstruct the 3D scene geometry and camera motions from monocular, stereo or RGB-D cameras. Tardif et al. [49] employed various epipolar constraints for estimating camera trajectories in urban scenes with a large amount of clutter. The open-source ORB-SLAM2 system [38] provides a robust SLAM implementation under various camera settings. Guan et al. [21] developed a real-time camera pose estimation system for wide-area augmented reality applications. Engel et al. [15] developed a feature-less monocular SLAM algorithm that allows us to build large-scale, consistent maps of the environment. Guan et al. [23] developed a registration system for wide-area augmented reality application with the aid of scene recognition and feature tracking

techniques. Guan et al. [22] also proposed a city scale on-device visual localization recognition method that can take advantages of inertial sensors and computer vision techniques. These algorithms, however, do not explicitly suppress the impacts of repetitive patterns. In this work, we focus on the development of a novel feature matching solution that is robust against repetitive patterns and can be integrated with and used in the SLAM framework.

Geo-tagged Images based localization [12, 44, 51, 54, 58] has been extensively studied and becomes increasingly popular along with the availability of large-scale geo-tagged images. However, these methods are restricted in two aspects: i) it is very time-consuming to prepare the geo-tagged images; ii) the current view of the camera might be arbitrarily different from the pre-stored images due to the environment changes. Another shortcoming of these methods is the low-precision of geo-localization (usually worse than GPS). In this work, we propose an automatic camera localization method that can leverage noisy GPS readings and visual odometry to provide sub-meter level localization in complicated scenes.

Integrating GPS readings and visual odometry (VO) [1] has been proved to be an effective way for reducing the effects of accumulated errors by visual odometry. In particular, Wei et al. [53] proposed to warp VO points to GPS coordinates and calculated the mean locations of two sources at each time-point. Agrawal and Konolige [1] used Kalman filter algorithm to rectify VO measurement errors, where camera locations are described with state variables. Sukkarieh et al. [48] proposed to detect possible faults before and during the fusion process to enhance the integrity of the navigation loop, including both the low-frequency faults in Inertial Measurement Units (e.g., bias in sensor readings or misalignment of various units), and high-frequency faults from GPS caused by muti-path errors. Najjar and Bonnifait [39] further integrated Kalman filter based fusion method with a belief theory. Parra et al. [41] proposed to improve GPS readings with visual odometry only when the GPS system is not reliable (e.g., Horizontal Dilution Of Position is greater than 10). The above data fusion methods considered camera locations as a set of disjoint points and tried to refine them separately, ignoring the fact that camera trajectories are defined in a continuous time-geography space. As a result, these methods are incapable of modeling long-range temporal changes, and generate frequent localization failures especially when the measurement errors are sparse yet dominant. In this work, we propose to fuse multi-modality data in a continuous space and empirically demonstrate its superior performance for complicated environments.

Min-cost Network Flow Optimization has been widely used in visual tracking. Different from the classical tracking methods, e.g., filtering [32, 52], bayesian sampling [34, 57], or subspace learning [42, 61, 62], network flow methods can respect spatial cooperative/conflicting constraints and are less sensitive to initializations or outliers. They also benefit from the sophisticated optimization techniques, including graph cuts [28], EM type boolean programming [29], dynamic programming [46], and linear programming [5]. In this work, we introduce a hierarchical network flow formula that additionally imposes consistency constraints over flow variables across multi-resolutions to mitigate the effects of repetitive patterns. The formula will exploit both flow constraints and motion smoothness constraints with a quadratic integer program that is a novel technique in its catalog.

1.3 Contributions and Organization

The three major contributions of this work include (i) an effective hierarchical network flow formula for addressing the repetitive pattern issues in visual feature tracking, (ii) a multi-modality camera localization method that can effectively integrate both discrete GPS readings and continuous VO measurements, and (iii) a novel video dataset for studying the repetitive pattern issues. We evaluated the proposed techniques on both the newly collected video dataset and public odometry benchmarks, and we achieved promising results in comparisons to the popular methods.

2 GEO-LOCALIZATION OF MOVING CAMERAS

The objective of this work is aimed at developing a robust camera localization algorithm that can work in scenes with repetitive patterns. The inputs to our method are geo-tagged monocular videos, and the outputs include the sequences of camera locations in world coordinate. The developed techniques can also be used for camera localization from monocular videos without geo-tags.

The structure of this section is organized as follows. In Section 2.1, we introduce how to formulate feature tracking problem in the min-cost network flow framework. In Section 2.2, we present a hierarchical network flow method to mitigate the effects of repetitive patterns. In Section 2.3, we augment the proposed hierarchical network flow formula with smoothness constraints over feature paths, which can further enhance system robustness. In Section 2.4, we develop a unified formula to simultaneously interpolate both visual odometry and GPS readings in a continuous space to obtain high-precision localization results.

2.1 Background: Feature Tracking with Min-Cost Flow Optimization

We first introduce how to formulate feature tracking as a traditional min-cost flow optimization problem. Our goal is to simultaneously track multiple features in a short sequence of video frames. There are two sub-tasks. On the one hand, given feature detections we need to select distinctive features of interest, e.g., corner, bars, sketches, and so on, that remain visible over time, i.e., feature selection problem. On the other hand, we need to match features across video frames or connect features appearing in consecutive frames, i.e., feature connection problem. Therefore, we cast the feature tracking task as a joint optimization problem that simultaneously selects distinctive features and determine their connections across video frames.

The above joint parsing task can be cast as an integer linear program, following the previous works [5, 28, 46]. Let $\mathbf{I}$ denote the input video sequence. Given interest points detected, a flow graph is imposed as the representation. Each graph node represents an interest point detected in video frames. We introduce two binary variables: $x_i \in \{0, 1\}$, that takes 1 if the feature detection i is selected for tracking, and $u_{ij} \in \{0, 1\}$, that takes 1 if the detections i and j are from two consecutive video frames and are connected (i.e., form a track). The index i ranges over possible detections in the input video sequence. Let E denote all the edges in the graph, $\mathbf{x}$ assemble all the binary indicator variables x_i and u_{ij}. The optimization of $\mathbf{x}$ are formulated as an energy minimization problem,

$$\min_{\mathbf{x}} \sum_i c_i x_i + \sum_{<i,j>\in E} c_{ij} u_{ij}, \tag{1}$$

$$s.t. \quad 0 \le x_i, \quad 0 \le u_{ij}, \tag{2}$$

$$\forall j, \sum_{\forall i, <i,j>\in E} u_{ij} = x_j = \sum_{\forall k, <j,k>\in E} u_{jk}, \tag{3}$$

$$\sum_{<a,i>\in E} x_{ai} = \sum_{<i,b>\in E} x_{ib}. \tag{4}$$

In the above formula, c_i denotes the cost of selecting detection i in a video frame and c_{ij} denotes the negative of matching strength between detections i and j. A popular choice for c_i is the local appearance contrast of an interest point [5]: $c_i = -\log \frac{P(x_i=1|\mathbf{I})}{P(x_i=0|\mathbf{I})}$. Accordingly, c_{ij} can be defined as the appearance dissimilarity between the two features i and j. Equation (2) imposes non-negative constraints over both x_i and u_{ij}. The constraints Equations (3) are used to enforce that the flow received by a node j is equal to the sum of flow starting from the same node. Moreover, all detections are linked to a source node a and point to a sink node b, as shown in the constraints Equation (4).

The above linear constraints have the property of being totally unimodular [5], which ensures that relaxing the integer constraints in Equation (1) and solving it with a linear program is still guaranteed to produce plausible integer solutions. Therefore, the optimization problem with relaxed integer constraints can be solved efficiently using existing network flow or linear algebra packages [28], providing a convenient framework to transform the feature correspondence problem into a joint feature selection problem.

In this work, we use the formula in Equation (1) as a starting point and introduce extra terms to regularize the selection process to deal with the repetitive pattern issues.

2.2 Feature Tracking with Hierarchical Network Flow

In this subsection, we extend the standard min-cost formula Equation (1) to develop a robust feature tracking algorithm. In scenes with repetitive patterns, as shown in Figure 1, a feature patch that is not distinctive at a certain scale will result in ambiguities in the selection process, i.e., determining x_i and u_{ij}. As aforementioned, a traditional solution to matching repetitive patterns is to group adjacent features to form distinct feature combinations that have larger image supports. These methods, however, suffer from the choices of grouping, e.g., the ways of calculating adherences or the expected support of feature combinations. To address this issue, we introduce a hierarchical composition procedure to group features of the same level to form feature combinations, and employ an optimization based method to adaptively select the distinctive feature combination. Our method is robust against the initializations of feature grouping, since the following optimization is capable of discarding the bad or non-informative feature combinations.

We develop a bottom-up approach to construct the multi-level feature representation. The first level of representation is composed of interest points detected in images. To form the second level or higher level representations, we recursively employ three main steps: calculate between-point distances; run K-means method to group these data points; prune clusters that have few data points (less than *minPT*) and data points that are far away (beyond *minDist* pixels) from their respect cluster center locations. Each cluster of data points at the level l is considered as a data point at the upper level $l + 1$. Let f_i denote the feature vector of the ith interest point, the distance between two sets of data points A and B is defined as

$$\frac{1}{2|A|}\sum_{i\in A}\min_{j\in B}\mathcal{D}(f_i, f_j) + \frac{1}{2|B|}\sum_{i\in B}\min_{j\in A}\mathcal{D}(f_i, f_j), \tag{5}$$

where $|A|$ indicates the size of A, and $\mathcal{D}$ indicates the Euclidean distance between two feature vectors. To obtain f_i, we use both geometric (i.e., spatial proximity) and appearances cues, as introduced in the section of Experiments. Note that an interest point might be isolated and is not used to compose any nodes of higher level. In this way, we allow parts of the detected features to be tracked separately and independently.

Figure 2 illustrates the hierarchy, where every node at a level is composed of multiple nodes at the lower level and belongs to a parent node at the higher level. With the hierarchy, the joint feature selection and feature connections can be performed for nodes of each level in parallel, while respecting the following cross-level consistency constraints: if one node is selected, at least one of its children nodes should be selected as well.

Formally, we relax the boolean variables x_i in Equation (2) to be integer variables, and we enforce that $x_i = \sum_{k\in Child(i)} x_k$. Let $< V, E >$ denote the flow graph, where V includes all graph nodes, and E the edges between nodes of the same level and the edges between parent and children nodes.

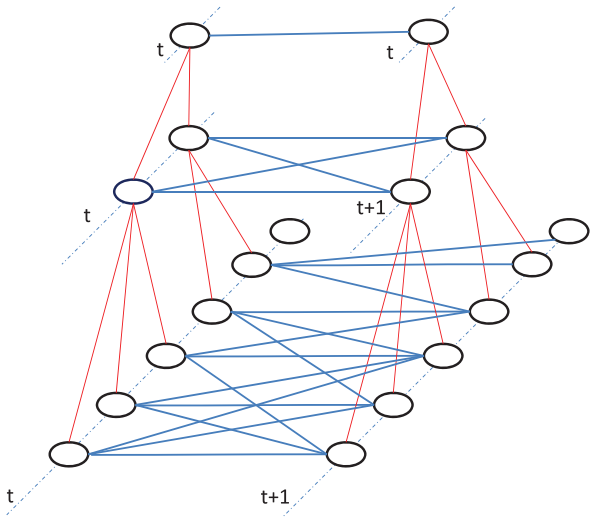

Fig. 2. Hierarchical flow graph at time t and $t+1$. Each blob indicates a feature point or a feature combination and is linked to other nodes of the same level or children nodes of the lower level. Each edge is augmented with a binary variable that indicates the between-node connectivity (blue) or children-parent composition (red).

The joint optimization problem Equation (1) can be re-written as the following:

$$\min_{\mathbf{x}} \sum_{i \in V} c_i x_i + \sum_{<i,j> \in E} c_{ij} u_{ij}, \tag{6}$$

$$s.t. \quad 0 \leq x_i, \quad 0 \leq u_{ij}, \tag{7}$$

$$\forall j, \sum_{\forall i, <i,j> \in E} u_{ij} = x_j = \sum_{\forall k, <j,k> \in E} u_{jk}, \tag{8}$$

$$\sum_{<a,i> \in E} x_{ai} = \sum_{<i,b> \in E} x_{ib}, \tag{9}$$

$$\forall i, x_i = \sum_{k \in Child(i)} x_k. \tag{10}$$

The flow variables x_i and u_{ij} are integers, which are different from their binary definitions in Equations (2), (3), and (4). Accordingly, the constraints in Equation (8) are used to ensure the node j receive and send out the exact same amount of flow. The constraint Equation (10) is used to impose the cross-level constrains. It is noteworthy that the above formula does not impose upper-bounds for the real-valued variables x_i, which is different from the traditional flow-based tracking methods [5, 28]. These variables are confined by the consistency constrains in the proposed hierarchy and are solved in a collectively fashion. We group the constraints in Equations (7) through (9) and denote them as $\mathbf{x} \in \mathbf{O}$.

Given a video sequence, we will detect interest points in individual video frames to obtain a set of possible feature locations and use the aforementioned bottom-up approach to construct the multi-level feature representation. With the graph (V, E), we solve Equation (6) to obtain the optimal feature detections x_i and connections u_{ij}, with which the path of a feature patch in a monocular video can be retrieved.

The integer program in Equation (6) provides an effective min-cost network flow method for matching repeated patterns across images. The upper-level graph nodes of the hierarchy correspond to clusters of repetitive patterns at lower levels. As reviewed in Section 1.2, generating discriminative clusters of regions is among one of the most successful approaches for matching repeated patterns across images. Different from the previous methods that require detections of repeated patterns in a separate step, the proposed formula is capable of automatically selecting distinctive groups of features through optimizing the program in Equation (6).

2.3 Integrating Motion Smoothness Objectives

We further enhance the proposed hierarchical network flow method with heuristic constraints. The key idea is to leverage the fact that a camera usually undergoes smooth movements over time or that the paths of multiple features in the proximity are likely to have similar motion patterns. This observation is particularly effective for monocular videos captured with an autonomous camera-based intelligent system, e.g., self-driving cars or robots. Therefore, we propose to enforce three individual or pair-wise smoothness constraints over the selection variables $\mathbf{x}$. (I) The appearance discrepancies between matched pairs of features are as minimal; (II) The spatial displacements of individual pairs of features should be not significant, i.e., without sudden movements; (III) spatial movements of nearby features are encouraged to be consistent with each other.

Formally, let I_i denote the appearance features of detection i in the videos. Recall that x_i indicates the feature detection i, u_{ij} indicates the connectivity between detections i and j. Let Δ_{ij} denote the spatial displacement between detections i and j. The objectives for enforcing motion smoothness are defined as follows:

$$\Phi(\mathbf{x}) = \sum_{<i,j>\in E} u_{ij}\|I_i - I_j\|^2 + \beta^a u_{ij}\Delta_{ij}^2 + \beta^b \sum_{k\in N(i), l\in N(k)} u_{ij}u_{kl}\|\Delta_{ij} - \Delta_{kl}\|^2, \tag{11}$$

where β^a and β^b are weighting parameters. The term $u_{ij}\|I_i - I_j\|^2$ is minimized when $u_{ij} = 1$ and features i and j have similar appearance. The second term is used to encourage smooth movements of individual feature paths. The third term is used to encourage similar motion patterns between nearby feature paths. It is noteworthy that only non-zero flow variables u_{ij} are regularized with the above objectives.

Thus, we rewrite Equation (6) as the following integer program with pair-wise objectives:

$$\min_{\mathbf{x}} \sum_{i\in V} c_i x_i + \sum_{<i,j>\in E} c_{i,j}u_{ij} + \gamma\Phi(\mathbf{x}), \tag{12}$$

$$\mathbf{x} \in \mathbf{O}, \tag{13}$$

where γ is a constant parameter. The constraints $\mathbf{x} \in \mathbf{O}$ are defined in Equations (7) through (9). Equation (12) includes two linear terms that encode the joint selection of feature detections and feature connections, and a pair-wise term $\Phi(\mathbf{x})$ that represents informative and contextualized knowledge in the selection processes. In particular, the product $u_{ij}u_{kl}$ represents the joint selection of two edges $< i, j >$ and $< k, l >$, corresponding to a pair of connections. The above formula is a quadratic assignment problem that is NP-hard to optimize in general. However, we can relax the selection variables $\mathbf{x}$ to be real values, and thus convert Equation (12) to be a constrained quadratic programming problem. The relaxed problem can be effectively solved by network flow packages or algebra methods [9].

Once solved the selection variables $\mathbf{x}$, we can retrieve feature paths between consecutive video frames [5]. In particular, feature pairs with non-zero (or small) u_{ij} are considered to be connected. In this way, the feature correspondences are directly inferred from flow variables, and are further used for estimating frame-to-frame visual odometry [3, 31, 47, 59].

In summary, we develop a novel quadratic programming formula to track features of interest across video frames. Our formula explicitly explores both hierarchical network flow constraints and spatial regularizations of local motion fields, both of which play critical roles for suppressing the effects of repetitive patterns. It is also effective for dealing other types of challenges, including low resolution, clutters, occlusions, and so on.

2.4 Multi-Modal Trajectory Fitting for High-Precision Localization

In this subsection, we further introduce a multi-modal method to fuse noisy sparse GPS readings and visual odometry (VO) points. Our method involves two coupled subtasks: warping VO points in metric space to GPS coordinate and predicting camera position at each time-step. On the one hand, these two subtasks are challenging, because (i) GPS points are noisy and are sparsely available over time (e.g., one reading for every 1–2min), and (ii) the warped VO points are noisy as well and the warping itself might incur errors. On the other hand, the two subtasks are mutually beneficial: (i) accurate warping is helpful to estimating complete and dense camera positions and (ii) accurate camera positions, once estimated, can help build high-quality warping. Therefore, we propose to jointly solve these two subtasks in a coordinated fashion.

The proposed joint formula includes three major objectives:

- The deformation between VO points and GPS readings is assumed to follow rotation, scale, and translation only, i.e., similarity transformation. This simplification is used to balance model complexity and computational cost in practice;
- Each camera trajectory is parameterized as a continuous function of camera positions in 3D. w.r.t. time, denoted as τ. We use the B-Spline function [13] whose first and second derivatives are both continuous. These high-order constraints are used to enforce smoothness over camera trajectories and allow robust estimation against noises and missing data;
- The optimal camera trajectory should be consistent with both warped VO points and sparse GPS points (if available).

Formally, let M denote the similarity transformation matrix. Let $\bar{x}_t, \hat{x}_t$ denote the homogeneous coordinates of cameras obtained from visual odometry and GPS device at time t, respectively. Let d denote the order of B-Spline, $B_l(t)$ the lth quadratic basis function. The spline function $\tau(t)$ can be written as a linear combination of basis functions: $\tau(t) = \sum_l \alpha_l B_l(t)$, where the basis functions B_l can be directly obtained given time interval and order d. We set $d = 3$ in this article. Let s index the GPS points, $\hat{v}_s$ denote the relative motion from s to $s + 1$. Thus, we aim to optimize the following objective function:

$$\begin{aligned}\arg\min_{M,\{\alpha_l\}} \sum_t \|\bar{x}_t - M\hat{x}_t\|^2 + \lambda^a \|\bar{x}_t - \sum_l \alpha_l B_l(t)\|^2 \\ +\lambda^b \sum_{s,s+1} \sum_l < \alpha_l B_l(s) - \alpha_l B_l(s+1), \hat{v} >,\end{aligned} \tag{14}$$

where λ^a, λ^b are constants, $< \cdot, \cdot >$ returns the cosine distance between two motion vectors. The three terms are used to solve the transformation matrix between GPS points $\hat{x}_t$ and VO points $\bar{x}_t$, interpolate the VO points $\bar{x}_t$ with the spline model, and minimize the disagreement between the predicted moving directions and those estimated from GPS readings from time s to $s + 1$, respectively. Basically, we interpolate the estimated VO points with a continuous spline function and further regularize the interpolation to preserve the second-order spatial structure of noisy GPS data, i.e., the relative motion of consecutive GPS points. Note that we do not directly utilize the original GPS data, which are sensitive to outliers.

Alternative Optimization. Equation (14) is an unconstrained blockwise least-square problem that can be solved alternatively: with fixed M, solve α_l, and vice versa. At each step, the subtask can be optimized analytically. We initialize M by solving the first term of least square and initialize α_l by solving the second term independently. We alternate these two steps until the change of the objective function is less than a small threshold (e.g., 10^{-3}). The alternative optimization often converges in 5-15 iterations.

3 EXPERIMENTS

In this section, we apply the proposed method over both a newly created video dataset and pubic odometry datasets to estimate camera trajectories using both monocular videos and noisy GPS readings.

Datasets. In computer vision community there have been multiple video datasets [19] for studying camera localization. However, most videos in these datasets do not include frequent repetitive patterns and are not suitable for studying the repetitive pattern issues. In this work, we fill in the gap through building a new collection of videos in challenging scenes that are full of repetitive patterns. Our dataset includes 20 video sequences captured from cameras mounted on a moving car or a riding bike. The cameras are equipped with GPS components. These videos are captured in two scenes: parking-lot and university campus. To facilitate quantitative evaluations, we use meter-based GPS coordinates. Each video segment lasts about 60-200 seconds with a frame rate of 30 frames per second. To obtain groundtruth camera trajectories, we develop an interactive toolkit that includes human in the loop. All videos were annotated by the same annotator. We split the video dataset into two even subsets, used for training and testing, respectively.

We also apply the proposed method on two public benchmarks, KITTI [19] and Hague Dataset [14], and compare it to the other popular localization methods. These results are used to demonstrate the generalization capabilities of our method while dealing with general scenarios.

Implementation. We use the SURF method [4] to detect interest points in each video frame. To solve Equation (12), we use the interior point method in network flow packages, e.g., Reference [5]. To calculate between-point distances, we represent each point using its location coordinates (two-dimensional) and extract a histogram of oriented gradients (HOG) [11] from the surrounding region (15 by 15 pixels). For the graph construction, we use three level of representations, and set the Ks (i.e., numbers of clusters) of the second and third level to be 50 and 15, respectively. For the second level of representation, we set *minPT* to be 3 and set the *minDist* to be 20 pixels. For the third level of representation, we set *minPT* to be 2, and set *minDist* to be 0 (i.e., do not change the clustering results).

We use the cross-validation method over training videos to choose the best hyper-parameters, including β^a, β^b, γ, λ^a, and λ^b. We implement the variants of our method using Matlab and run them on an Intel i7 2.8GHZ Quad-Core machine with 16G RAM. To deal with streaming video sequences, we use the popular sliding window method. We set the window size to be a constant (e.g., 20 frames) and slide it with a step (e.g., 10 frames). For a video of 960×540 resolution, the feature tracking step takes 30ms per frame, the multi-modal fitting step takes 20ms per frame.

Evaluation Baselines. We evaluate the proposed method from two aspects: analyzing contributions of individual components, i.e., visual odometry and trajectory fitting, and comparing to the popular alternative methods.

For the *visual odometry step*, we implement the following feature tracking methods: (i) *BI*, the monocular version of the widely used VO software LIBVISO2 [20]; (ii) *BII*, the method by Song et al. [47]; (iii) *Flow*, that optimizes the standard min-cost network flow formula, i.e., Equation (1);

(iv) *HFlow*, that optimizes the hierarchical min-cost network flow formula, i.e., Equation (6); (v) *CHFlow* that optimizes the constrained hierarchical min-cost network flow formula, i.e., Equation (12); and (vi) *CHFlowLinear*, that sets $\beta^b = 0$ in Equation (11) and optimizes the objective function Equation (12), which becomes a linear programming problem. The results of the proposed variants are used to calculate visual odometry using the monocular version of LIBVISO, a standard visual odometry pipeline [20]. It is noteworthy that both BI and BII are considered as the state-of-the-art feature correspondence methods although they are not designed for matching repetitive patterns.

For the *trajectory fitting* step, we implement three variants of the proposed fusion strategy: (i) *S*, that first solve a similarity matrix M (i.e., minimizing the first term in Equation (14)) and then calculates the camera position at each time-step as the average of the warped VO point and GPS reading; (ii) *SS*, that extends the previous variant by interpolating camera trajectories (i.e., VO points) with a cubic spline function of time, i.e., setting $\lambda^b = 0$ in Equation (14); (iii) *SSC*, that additionally utilizes the second-order statistics over GPS readings, i.e., optimizing Equation (14) with nonzero λ^a and λ^b.

The combination of the above two steps results in a total of 18 algorithms for localizing moving cameras. We will test and evaluate these algorithms on the same collection of videos to analyze the contributions of individual components of the proposed method.

Evaluation Metrics. We apply the above methods over monocular videos to estimate camera locations and compare the estimation results with the groundtruth locations. Following previous works [19], we divide all localization results into two subsequences: unlocalized and localized. A video sequence is considered to be localized only when for at least five seconds of the average localization error is less than 20m. The subsequent video sequences beyond the localization point are considered to be localized.

For each video sequence, we calculate and report the localization time of every baseline method, i.e., the time needed for an algorithm to successfully localize the camera. In other words, localization time is equivalent to the number of video frames for a method to process before successfully localizing the camera in the map. We also report average localization errors (in meters) of each algorithm on localized subsequences of the input videos.

Qualitative Results. We first visualize the outcomes of the proposed methods in Figure 3, which plots the estimated VO points and GPS points in world coordinate. We generate the figure using the similarity matrix M solved from Equation (14).

Figure 4 visualizes the features selected by the proposed hierarchical network flow method for three typical scenarios. We show three video frames each of which is overlaid with the originally detected features (left column) as well as the selected features (right column). In the figures, we use circles of green, red and blue to represent features of the first, second and third levels in the hierarchy, respectively. For the image in the last row, we crop the bottom region of vehicle so that no interest points are detected on the vehicle body. It is noteworthy that a node in the flow graph might be independently matched across video frames without hierarchical constraints. In Figure 4, for example, these isolated features at the first level are plotted as green circles not encircled by any blue or red circles. From these figures, we can observe that the nodes with non-zero flow mostly locate on boundaries or corners, which are distinctive to be tracked over time. The nodes of higher levels are often composed of multiple feature points and form discriminative primitive structures that are distinctive even in the scenes with repetitive patterns.

In Figure 5, we show in the top row a video frame overlaid with interest points and their matches in the previous video frame. We also plot in the bottom row the camera trajectories estimated

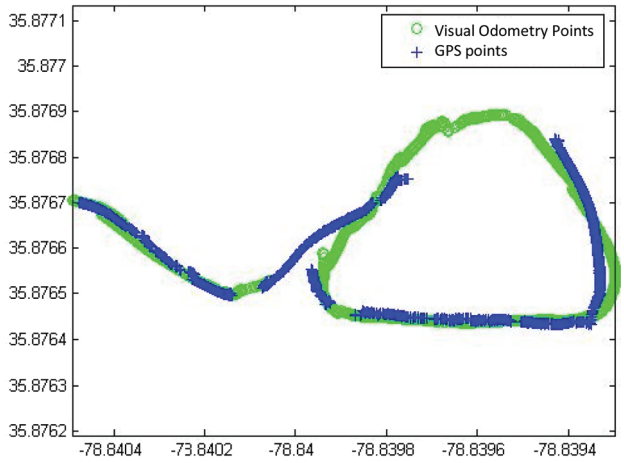

Fig. 3. Exemplar results of fusing Visual Odometry(VO) points (green) and GPS points (blue). The transformation matrix is obtained by solving Equation (14).

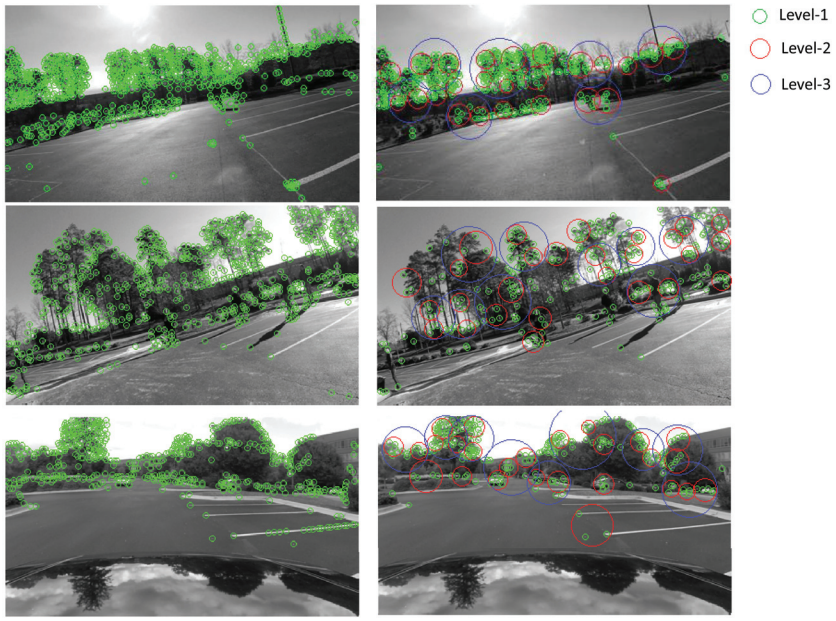

Fig. 4. Features and feature combinations selected by the proposed hierarchical min-cost network flow optimization method. We use a three-level representation, whose nodes are represented as circles of green, red, and blue, respectively. Left: original detections of interest features; right: selected features and feature combinations in the hierarchy. Only nodes with non-zero flow (x_i) are displayed.

by various methods, including the GPS readings, CHFlow+S, and CHFlow+SSC. The groundtruth camera trajectories are included for comparisons as well. It is noteworthy that most interest features appear on the regions of trees, which have similar appearance or texture patterns with each other. Moreover, the method CHFlow+S has large errors around the bottom-mid area due to the

Fig. 5. Exemplar results on the scene of parking-lot. Top: a video frame overlaid with matched keypoints detected by our method, where green points are from the previous frame; Bottom: camera positions in GPS (red points), the groundtruth trajectories(green), the estimated trajectories by the CHFlow method with the fusion strategy *S* (blue) or multi-modal fusion strategy *SSC* (yellow).

Fig. 6. Exemplar video frames for which the proposed method CHFlow+S does not work properly. The corresponding areas are highlighted by a red ellipse in Figure 5. The videos are full of lighting changes, reflections, and sudden object movements.

drifting issues. In contrast, the multi-modal method CHFlow+SSC (in yellow) is quite close to groundtruth trajectories.

In Figure 6, we show two video frames for which the proposed hierarchical network flow method CHFlow+S does not work properly. The corresponding areas are highlighted in Figure 5 (red ellipse). The two scenarios are challenging, because they have sudden foreground movements being very close to the camera, and strong sunshine reflections. In contrast, the proposed CHFLow+SSC

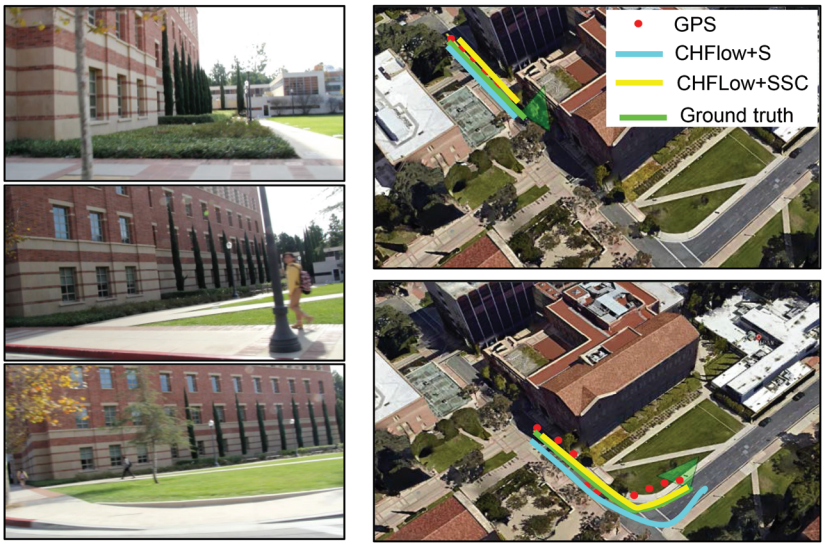

Fig. 7. Exemplar results on the scene of university campus. Column 1: three video frames (not overlaid with tracked points for better display); Column 2: the camera positions in GPS (red points), groundtruth trajectories (green), estimated trajectories by the CHFlow method with fusion strategy *S* (blue) or multi-modal fusion strategy *SSC* (yellow).

method employs additional, noisy and sparse GPS data and can significantly improve camera localization accuracies.

In Figure 7, we show three video frames in the first column, and plot the estimated camera trajectories in the second column (over different time-stamp). We can observe that (i) all methods work well for the time-period shown in the top-right subfigure; (ii) the visual odometry method with sequent fusion strategy (*S*) drifts a bit while the vehicle is making left turn, as shown in the bottom-right figure. The proposed CHFlow+SSC method, in contrast, can rectify such drifting issues by exploring GPS data.

It is noteworthy that the proposed method is capable of addressing repetitive patterns at a variety of scales. According to Wu et al. [56], a scene element (e.g., window corner) might be imaged as sketch, texture, or flat regions, while its distance to the camera varies. Our method employs a hierarchical network flow formula to adaptively select and match features of multiple scales, and imposes smoothness constraints to regularize the matching process. This provides an implicit way for addressing the scaling factor.

Quantitative Results. We further quantitatively evaluate individual components of the proposed method and compare them to the other odometry methods. Table 1 reports the localization time of each method, i.e., how many seconds of video frames this method need to process to successfully localize a video sequence. Table 2 reports average localization errors over localized video sequences. Video sequences that were not localized are indicated with $*$.

In general the proposed method can localize a monocular video sequence in 3–5s with sub-meter level of accuracies (0.57m on average). Moreover, the component-wise comparisons result in the following observations. (I) The proposed flow based feature tracking method can significantly improve the localization performance of the basic monocular visual odometry pipeline [20] and achieved comparable localization accuracies as the other state-of-the-art methods [33, 47]. Note that these baseline algorithms are designed for generic types of scenes, and thus cannot work

Table 1. Localization Time (seconds)

		Parking Lot Videos					University Campus Videos					
Video Duration (s)		67	121	89	132	125	95	68	110	142	186	Mean
Step I	Step II	01	02	03	04	05	06	07	08	09	10	Mean
CHFlow	SSC	**1.2**	**1.1**	**1.3**	**1.8**	**3.4**	**2.8**	**3.9**	**2.5**	**3.4**	**3.7**	**2.5**
CHFlow	SS	2.1	2.2	2.5	1.9	4.9	3.1	5.3	2.9	3.8	3.9	3.3
CHFlow	S	2.7	2.5	3.6	2.4	6.8	4.5	6.4	3.5	4.6	4.1	4.1
CHFlowLinear	SSC	1.5	1.4	2.6	1.8	4.5	3.3	5.3	3.1	3.5	3.8	3.1
CHFlowLinear	SS	2.3	3.1	2.8	2.1	5.3	4.2	5.8	3.5	4.8	4.1	3.8
CHFlowLinear	S	2.9	3.1	3.8	4.5	8.1	4.8	8.1	4.6	4.9	5.3	5.0
HFlow	SSC	2.4	3.1	4.2	3.5	5.9	5.8	7.2	4.8	5.3	5.2	4.7
HFlow	SS	2.9	3.5	4.8	4.7	7.2	6.4	7.3	5.1	6.2	5.9	5.4
HFlow	S	3.5	4.2	6.3	4.9	9.3	7.9	8.6	6.7	5.8	5.8	6.3
Flow	SSC	3.4	4.1	6.4	5.1	8.9	7.7	8.5	6.9	6.5	6.1	6.4
Flow	SS	3.7	4.3	7.2	5.2	9.5	7.8	8.9	7.2	7.2	7.3	6.1
Flow	S	4.1	4.4	6.9	5.4	9.8	9.2	9.1	7.7	8.8	7.8	7.3
BII	SSC	3.9	4.3	6.5	4.8	9.7	7.2	8.1	7.2	6.7	5.6	6.4
BII	SS	5.2	4.5	7.2	9.5	10.4	8.9	9.2	8.7	9.1	7.2	8.0
BII	S	8.7	*	7.5	12.1	12.5	9.3	10.7	10.2	12.8	9.9	10.4
BI	SSC	12.8	9.2	12.5	11.2	14.3	12.7	9.8	15.1	*	15.6	12.6
BI	SS	14.5	*	*	12.8	14.1	13.5	*	15.5	*	16.6	14.5
BI	S	17.8	14.3	16.5	18.0	19.4	*	14.3	*	*	18.1	16.9
[33]		16.8	14.7	*	15.6	*	*	*	*	*	16.2	15.8

Step I, visual odometry; Step II, multi-modal fusion; Flow, the standard min-cost network flow method; HFlow, the proposed hierarchical min-cost network flow method; CHFlow, the proposed hierarchical min-cost network flow method; CHFlowLinear, first-order constrained hierarchical min-cost network flow method; S, sequent data fusion strategy; SS, solving similarity matrix and spline representation jointly; SSC, enforce the second-order constraints over the method SS; BI, the monocular version of LIBVISO2; BII, Song et al. [47]; ∗, failure of localization.

robustly when there exist significant amount of repetitive patterns. For example, the method of Reference [33] cannot localize cameras for six out of ten video sequences. (II) The proposed hierarchical network flow method (i.e., *HFlow*) clearly outperforms the standard network flow method (i.e., *Flow*), in terms of both localization times and localization accuracies. (III) The proposed smoothness constraints can further enhance system accuracies over all testing video sequences. (IV) The proposed multi-modal fusion strategy (i.e., *SSC*) achieved much better performance than the other two fusion strategies (i.e., *SS* and *S*). The comparisons among algorithms using *S* or *SS* showed that enforcing high-order smoothness constraints (through spline representation) in the continuous space can result in improved performance and robustness.

In the above experiments, we test the proposed feature tracking method by evaluating how much it can boost the standard visual odometry pipeline. The developed techniques can be applied for the other vision problems. We also note that there have been extensive studies about the fusion strategies of multi-modal cues, and we compare the proposed fusion strategy *SSC* with the other two popular strategies, i.e., *S* and *SS*, in quantitative experiments.

Results on the KITTI Odometry Benchmark. We apply the proposed method over the KITTI Odometry benchmark [19] and compare it to other popular methods. The benchmark includes 11 video sequences (numbers 0–10) for training, and 11 video sequences (numbers 11–21) for testing.

Table 2. Average Localization Errors (meters) over Individual Video Clips

		Parking Lot Videos					University Campus Videos					
Video Duration (s)		67	121	89	132	125	95	68	110	142	186	Mean
CHFlow	SSC	**0.42**	**0.54**	**0.31**	**0.92**	**0.76**	**0.66**	**0.45**	**0.81**	**0.34**	**0.51**	**0.57**
CHFlow	SS	0.58	0.89	0.66	1.46	0.87	0.70	0.51	0.94	0.46	0.65	0.77
CHFlow	S	1.92	0.92	0.79	1.97	0.99	0.89	0.62	1.68	0.77	0.97	1.15
CHFlowLinear	SSC	0.76	0.87	0.43	1.01	0.82	0.88	0.75	1.25	0.39	0.81	0.79
CHFlowLinear	SS	0.93	1.02	0.81	1.49	1.32	1.11	1.07	1.32	0.87	0.93	1.08
CHFlowLinear	S	2.05	1.32	1.43	2.11	1.45	1.87	1.73	1.89	0.93	1.20	1.60
HFlow	SSC	1.81	1.11	0.92	1.89	1.02	1.97	1.14	1.77	0.91	1.52	1.41
HFlow	SS	2.23	1.34	1.24	2.01	1.54	2.08	1.69	1.94	0.95	1.78	1.68
HFlow	S	2.57	2.51	2.07	2.19	2.07	2.51	2.04	2.35	1.43	2.11	2.19
Flow	SSC	2.89	2.49	1.98	2.40	2.17	2.55	2.32	2.21	2.34	2.45	2.38
Flow	SS	2.94	2.57	2.43	2.59	2.41	2.79	2.46	2.39	2.51	2.49	2.56
Flow	S	3.25	2.78	2.54	2.73	3.04	2.84	2.87	2.51	3.44	3.10	2.91
BII	SSC	3.14	2.69	2.24	3.07	3.11	2.34	2.54	2.98	3.69	2.52	2.83
BII	SS	3.25	2.78	2.51	3.43	3.35	2.78	3.77	3.27	4.78	2.89	3.28
BII	S	4.21	*	3.10	4.11	4.97	2.86	4.14	3.55	5.43	4.14	4.06
BI	SSC	5.87	4.12	3.42	5.09	5.34	3.09	5.14	4.45	*	5.78	4.70
BI	SS	7.97	*	*	6.14	6.12	3.14	*	5.19	*	5.97	5.76
BI	S	10.21	8.32	7.53	7.10	6.97	*	14.3	*	*	6.21	8.66
[33]		11.32	12.73	*	13.10	*	*	*	*	*	18.41	13.89

Step I, visual odometry; Step II, multi-modal fusion; Flow, the standard min-cost network flow method; HFlow, the proposed hierarchical min-cost network flow method; CHFlow, the proposed quadratic constrained hierarchical min-cost network flow method; CHFlowLinear, first-order constrained hierarchical min-cost network flow method; S, sequent data fusion strategy; SS, solving similarity matrix and spline representation jointly; SSC, enforce the second-order constraints over the method SS; BI, the monocular version of LIBVISO2; BII, Song et al. [47]; ∗, failure of localization.

We use the testing sequences in the evaluation. We employ the variant of CHFlow to discover feature correspondences and then call the monocular version of LIBVISIO2 [20] to reason camera poses over time. All model parameters are chosen using the cross-validation method over the provided training sequences. For all test sequences, we use two quantitative results: relative translational error and rotational error. The translational errors are measured in percent (%) and the rotational errors (deg/m) are measured in degrees per meter.

Table 3 reports the results of our method and the other top-ranked published methods that use monocular videos. Only published methods reported on the KITTI leaderboard are included for comparisons. Our method achieved comparable translational and rotational errors as the state-of-the-art methods. The proposed method does not employ any extra knowledge or training process and can largely enhance the basic visual odometry pipeline [20] in terms of both metrics. In contrast, the top-ranked method [18] employed pre-trained deep learning models for estimating ground-planes, which requires extra efforts in deployment. These comparisons clearly demonstrated the superiority of the proposed hierarchical flow optimization formula.

Results on the Hague Dataset [14]. We also evaluate the proposed method on the publicly available Hague dataset, which consists of three sequences of varying lengths, from 600m to 5km. Figure 8 plots sample video frames (top row) and ground-truth camera trajectories on the map (bottom row). These low-resolution video sequences include severe occlusions due to crowded scenes and moving vehicles close to the camera. These scenes also include many repetitive patterns, e.g.,

Table 3. Translational and Rotational Errors for the KITTI Odometry Benchmark

Method	Translation	Rotation
PMO / PbT-M2 [18]	**2.05**%	0.0051[deg/m]
FTMVO [36]	2.24%	0.0049[deg/m]
CHFlow	2.33%	**0.0038**[deg/m]
PbT-M1 [17]	2.38%	0.0053[deg/m]
MLM-SFM [47]	2.54%	0.0057[deg/m]
RMCPE+GP [35]	2.55%	0.0086[deg/m]
VISO2-M [20]	11.94%	0.0234[deg/m]

Only published methods using monocular videos are included for comparisons.

Fig. 8. Hague video dataset. Top, sample video frames; Bottom (from left to right), trajectories of the sequences 1, 2, and 3, respectively.

Table 4. End-point Errors for Sequences in the Hague Dataset [14]

Sequence	Frames	Length (m)	Song [47]	CHFlow
1	2,500	609.34	5.37	4.13
2	3,000	834.39	1.99	1.82
3	19,000	5045.45	4.85	2.51

textures, flat regions, and road marks. Song et al. [47] evaluated their method on this dataset in terms of loop closure, i.e., end-point error relative to map information. In this work, we follow the same strategy to evaluate the proposed CHFlow method. Table 4 reports the comparisons between Song et al. [47] and the proposed method. The proposed method achieved much better accuracies on all the three sequences. Notably, the improvements over longer sequences are more significant, which clearly demonstrate the robustness of the proposed hierarchical network flow formula.

4 CONCLUSION

This article studied the video-based camera localization problem for scenes with repetitive patterns and introduced a multi-modality method that can achieve high-quality camera localization using monocular videos. Our efforts include three aspects. First, we introduce a constrained

hierarchical network flow method for matching features over consecutive video frames while respecting smoothness constraints in local motion fields. This results in a novel quadratic programming formula that can adaptively select and match distinctive features and feature combinations at multiple resolutions. Second, we further introduce a formula that simultaneously warps visual odometry locations to world coordinate and estimate camera locations. Third, we collect new video dataset that comprises of videos with repetitive patterns, which is the first one in its catalog. We exhaustively evaluate the proposed method through analyzing the contributions of individual components and comparing to the other popular methods. Results showed that our method can localize moving cameras with high accuracies. The developed techniques in this work have wide applications in other computer vision tasks, including object tracking, shape matching, robotic mapping, 3D scene reconstructions, and so on.

The proposed method is limited to the fact that it comprises of multiple stages, e.g., feature extraction, feature matching (with the proposed hierarchical network flow method), and modality fusion. The choices at each stage might affect system performance. While this is a common issue existing in most odometry methods, a promising direction is to develop a unified framework capable of learning the optimal parameters from the past experiences, which will be studied in the future.

REFERENCES

[1] M. Agrawal and K. Konolige. 2006. Real-time localization in outdoor environments using stereo vision and inexpensive gps. In *Proceedings of the International Conference on Pattern Recognition*.

[2] K. Schindler B. Leibe and L. V. Gool. 2008. Accurate image matching in scenes including repetitive patterns. In *Proceedings of the International Conference on Robot Vision*.

[3] H. Badino, A. Yamamoto, and T. Kanade. 2013. Visual odometry by multiple-frame feature integration. In *Proceedings of the International Workshop on Computer Vision for Autonomous Driving at (ICCV'13)*.

[4] H. Bay, A. Ess T. Tuytelaars, and L. Van Gool. 2008. SURF: Speeded up robust features. *Comput. Vision Image Understand.* 110, 3 (2008), 346–359.

[5] J. Berclaz, F. Fleuret, E. Turetken, and P. Fual. 2011. Multiple object tracking using k-shortest paths optimization. *IEEE Trans. Pattern Anal. Mach. Intell.* 33, 9 (2011), 1806–1819.

[6] Guillermo Botella, Antonio García, Manuel Rodríguez-Álvarez, Eduardo Ros, Uwe Meyer-Baese, and María C Molina. 2010. Robust bioinspired architecture for optical-flow computation. *IEEE Trans. Very Large Scale Integr. Syst.* 18, 4 (2010), 616–629.

[7] Guillermo Botella, Uwe Meyer-Baese, Antonio García, and Manuel Rodríguez. 2012. Quantization analysis and enhancement of a VLSI gradient-based motion estimation architecture. *Digital Signal Process.* 22, 6 (2012), 1174–1187.

[8] L. Cai, L. He, T. Yamashita, Y. Xu, Y. Zhao, and X. Yang. 2011. Robust contour tracking by combining region and boundary information. *IEEE Trans. Circ. Syst. Video Technol.* 21, 12 (Dec 2011), 1784–1794. DOI: http://dx.doi.org/10.1109/TCSVT.2011.2133550

[9] M. Celis, J. Dennis, and R. Tapia. 1984. A trust region strategy for nonlinear equality constrained optimization. *Numer. Optim.* (1984), 71–82.

[10] Tomas Crivelli, Matthieu Fradet, Pierre-Henri Conze, Philippe Robert, and Patrick Pérez. 2015. Robust optical flow integration. *IEEE Trans. Image Process.* 24, 1 (2015), 484–498.

[11] Navneet Dalal and Bill Triggs. 2005. Histograms of oriented gradients for human detection. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Vol. 1. IEEE, 886–893.

[12] A. J. Davison. 2003. Real-time simultaneous localisation and mapping with a single camera. In *Proceedings of the IEEE Conference on Computer Vision*.

[13] C. de Boor. 1978. *A Practical Guide to Splines*. Springer-Verlag.

[14] Gijs Dubbelman and Frans C. A. Groen. 2009. Bias reduction for stereo based motion estimation with applications to large scale visual odometry. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'09)*. IEEE, 2222–2229.

[15] Jakob Engel, Thomas Schöps, and Daniel Cremers. 2014. LSD-SLAM: Large-scale direct monocular SLAM. In *Proceedings of the European Conference on Computer Vision*. Springer, 834–849.

[16] Bin Fan, Fuchao Wu, and Zhanyi Hu. 2011. Towards reliable matching of images containing repetitive patterns. In *Pattern Recognition Letters* 32, 14 (2011), 1851–1859.

[17] Nolang Fanani, Matthias Ochs, Henry Bradler, and Rudolf Mester. 2016. Keypoint trajectory estimation using propagation based tracking. In *Proceedings of the IEEE Intelligent Vehicles Symposium*. IEEE, 933–939.

[18] Nolang Fanani, Alina Stürck, Matthias Ochs, Henry Bradler, and Rudolf Mester. 2017. Predictive monocular odometry (PMO): What is possible without RANSAC and multiframe bundle adjustment? *Image Vision Comput.* 68 (2017), 3–13.
[19] A. Geiger, P. Lenz, and R. Urtasun. 2012. Are we ready for autonomous driving the KITTI vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*
[20] A. Geiger, J. Ziegler, and C. Stiller. 2011. StereoScan: Dense 3D reconstruction in real-time. In *Proceedings of the Intelligent Vehicles Symposium.*
[21] Tao Guan, Liya Duan, Junqing Yu, Yongjian Chen, and Xu Zhang. 2011. Real-time camera pose estimation for wide-area augmented reality applications. *IEEE Comput. Graph. Appl.* 31, 3 (2011), 56–68.
[22] Tao Guan, Yunfeng He, Juan Gao, Jianzhong Yang, and Junqing Yu. 2013. On-device mobile visual location recognition by integrating vision and inertial sensors. *IEEE Trans. Multimedia* 15, 7 (2013), 1688–1699.
[23] T. Guan and C. H. Wang. 2009. Registration based on scene recognition and natural features tracking techniques for wide-area augmented reality systems. *IEEE Trans. Multimedia* 11, 8 (2009), 1393–1406.
[24] S. Ha, S. Lee, and N. Cho. 2012. Discrimination and description of repetitive patterns for enhancing the performance of feature-based recognition. *Image Vision Comput.* 30, 11 (2012), 817–828.
[25] Patrick Héas, Cédric Herzet, and Etienne Mémin. 2012. Bayesian inference of models and hyperparameters for robust optical-flow estimation. *IEEE Trans. Image Process.* 21, 4 (2012), 1437–1451.
[26] T. Igarashi, T. Moscovich, and J. Hughes. 2005. As rigid-as-possible shape manipulation. *ACM Trans. Graph.* 24, 3 (2005), 1134–1141.
[27] H. Jegou, M. Douze, and C. Schmid. 2009. On the burstiness of visual elements. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09).*
[28] S. Khan and M. Shah. 2009. Tracking multiple occluding people by localizing on multiple scene planes. *IEEE Trans. Pattern Anal. Mach. Intell.* 31, 3 (2009), 505–519.
[29] B. Leibe, K. Schindler, and L. V. Gool. 2007. Coupled detection and trajectory estimation for multi-object tracking. In *Proceedings of the IEEE Conference on Computer Vision.*
[30] G. Lentaris, I. Stamoulias, D. Soudris, and M. Lourakis. 2016. HW/SW codesign and FPGA acceleration of visual odometry algorithms for rover navigation on mars. *IEEE Trans. Circ. Syst. Video Technol.* 26, 8 (Aug 2016), 1563–1577. DOI: http://dx.doi.org/10.1109/TCSVT.2015.2452781
[31] C. Liu, J. Yuen, and A. Torralba. 2011. SIFT flow: Dense correspondence across scenes and its applications. *IEEE Trans. Pattern Recogn. Mach. Intell.* 33, 5 (2011), 978–994.
[32] X. Liu, L. Lin, and H. Jin. 2013. Contextualized trajectory parsing via spatio-temporal graph. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 12 (2013), 3010–3024.
[33] R. Urtasun M. A. Brubaker, A. Geiger. 2013. Lost! leveraging the crowd for probabilistic visual self-localization. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition.*
[34] E. Maggio, F. Smerladi, and A. Cavallaro. 2007. Adaptive multifeature tracking in a particle filtering framework. *IEEE Trans. Circ. Syst. Video Technol.* 17, 10 (Oct. 2007), 1348–1359. DOI: http://dx.doi.org/10.1109/TCSVT.2007.903781
[35] M. Hossein Mirabdollah and Bärbel Mertsching. 2014. On the second order statistics of essential matrix elements. In *Proceedings of the German Conference on Pattern Recognition.* Springer, 547–557.
[36] M. Hossein Mirabdollah and Bärbel Mertsching. 2015. Fast techniques for monocular visual odometry. In *Proceedings of the German Conference on Pattern Recognition.* Springer, 297–307.
[37] Mahmoud A. Mohamed, Hatem A. Rashwan, Bärbel Mertsching, Miguel Angel García, and Domenec Puig. 2014. Illumination-robust optical flow using a local directional pattern. *IEEE Trans. Circ. Syst. Video Technol.* 24, 9 (2014), 1499–1508.
[38] Raul Mur-Artal and Juan D. Tardós. 2017. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Trans. Robot.* 33, 5 (2017), 1255–1262.
[39] M. E. El Najjar and P. Bonnifait. 2005. A road-matching method for precise vehicle localization using belief theory and Kalman filtering. *Auton. Robots* 19, 2 (2005), 173–191.
[40] M. Perdoch P. Doubek, J. Matas and O. Chum. 2010. Image matching and retrieval by repetitive patterns. In *Proceedings of the IEEE Conference on Pattern Recognition.*
[41] Ignacio Parra, Miguel Angel Sotelo, David F. Llorca, C. Fernandez, A. Llamazares, N. Hernandez, and I. Garcfa. 2011. Visual odometry and map fusion for GPS navigation assistance. In *Proceedings of the IEEE International Symposium on Industrial Electronics.*
[42] C. Qian and Z. Xu. 2016. Robust visual tracking via sparse representation under subclass discriminant constraint. *IEEE Trans. Circ. Syst. Video Technol.* 26, 7 (July 2016), 1293–1307. DOI: http://dx.doi.org/10.1109/TCSVT.2015.2424091
[43] J. Rehder, K. Upta, and S. Nuske, and S. Singh. 2008. Global pose estimation with limited GPS and long range visual odometry. In *Proceedings of the International Conference on Robotics and Automation.*
[44] G. Schindler, M. Brown, and R. Szeliski. 2007. City-scale location recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*

[45] T. Senst, V. Eiselein, and T. Sikora. 2012. Robust local optical flow for feature tracking. *IEEE Trans. Circ. Syst. Video Technol.* 22, 9 (Sept. 2012), 1377–1387. DOI: http://dx.doi.org/10.1109/TCSVT.2012.2202070

[46] K. Shafique and M. Shah. 2005. A noniterative greedy algorithm for multiframe point correspondence. *IEEE Trans. Pattern Anal. Machine Intell.* 27, 1 (2005), 51–65.

[47] S. Song and M. Chandraker. 2014. Robust scale estimation in real-time monocular SFM for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*

[48] S. Sukkarieh, E. M. Nebot, and H. F. Durrant-Whyte. 1999. A high integrity IMU/GPS navigation loop for autonomous land vehicle applications. *IEEE Trans. Robot. Automat.* 15, 3 (1999), 572–578.

[49] Jean-Philippe Tardif, Yanis Pavlidis, and Kostas Daniilidis. 2008. Monocular visual odometry in urban environments using an omnidirectional camera. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'08).* IEEE, 2531–2538.

[50] A. Torii, J. Sivic, T. Pajdla, and M. Okutomi. 2013. Visual place recognition with repetitive structures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'13).*

[51] G. Vaca-Castano, A. Zamir, and M. Shah. 2012. City scale geo-spatial trajectory estimation of a moving camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'12).*

[52] J. Vermaak, A. Doucet, and P. Perez. 2003. Maintaining multimodality through mixture tracking. In *Proceedings of the IEEE Conference on Computer Vision (CV'03).*

[53] Lijun Wei, Cindy Cappelle, Yassine Ruichek, and Frederick Zann. 2011. GPS and stereovision-based visual odometry: Application to urban scene mapping and intelligent vehicle localization. *Int. J. Vehic. Technol.* 2011, Article 439074 (2011), 17.

[54] B. Williams, G. Klein, and I. Reid. 2011. Automatic relocalization and loop closing for real-time monocular SLAM. *IEEE Trans. Pattern Recogn. Mach. Ingell.* 33, 9 (2011), 1699–1712.

[55] Michael G. Wing, Aaron Eklund, and Loren D. Kellogg. 2005. Consumer-grade global positioning system (GPS) accuracy and reliability. *J. Forest.* 103, 4 (2005), 169–173.

[56] Ying Nian Wu, Cheng-En Guo, and Song-Chun Zhu. 2008. From information scaling of natural images to regimes of statistical models. *Quart. Appl. Math.* (2008), 81–122.

[57] Q. Yu, G. Medioni, and I. Cohen. 2007. Multiple target tracking using spatio-temporal Markov chain Monte Carlo data association. In *Proceedings of the IEEE Conference on Computer Vision.*

[58] A. Zamir and M. Shah. 2010. Accurate image localization based on Google maps street view. In *Proceedings of the European Conference on Computer Vision.*

[59] J. Zhang and S. Singh. 2014. LOAM: Lidar odometry and mapping in real-time. In *Proceedings of the Robotics: Science and Systems Conference.*

[60] K. Zhang, L. Zhang, M. H. Yang, and Q. Hu. 2013. Robust object tracking via active feature selection. *IEEE Trans. Circ. Syst. Video Technol.* 23, 11 (Nov. 2013), 1957–1967. DOI: http://dx.doi.org/10.1109/TCSVT.2013.2269772

[61] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja. 2013. Robust visual tracking via structured multi-task sparse learning. *Int. J. Comput. Vision* 101, 2 (2013), 367–383.

[62] T. Zhou, Y. Lu, and H. Di. 2015. Locality-constrained collaborative model for robust visual tracking. *IEEE Trans. Circ. Syst. Video Technol.*99 (2015), 1–1. DOI: http://dx.doi.org/10.1109/TCSVT.2015.2493498

[63] B. Zitova and J. Flusser. 2003. Image registration methods: A survey. In *Image and Vision Computing* 21, 11 (2003), 977–1000.

Received October 2017; revised April 2018; accepted May 2018