Monocular Localization of a moving person onboard a Quadrotor MAV

Hyon Lim and Sudipta N. Sinha

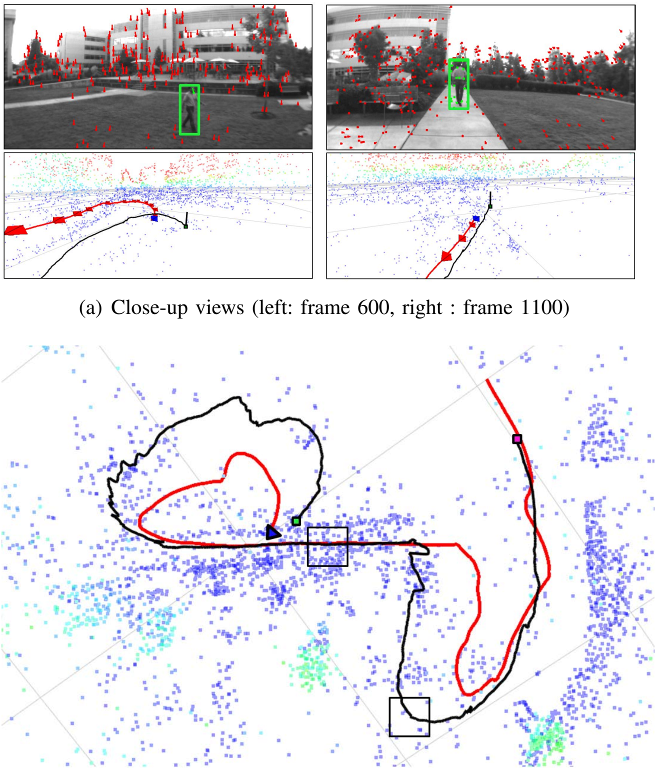
***Abstract*— In this paper, we propose a novel method to recover the 3D trajectory of a moving person from a monocular camera mounted on a quadrotor micro aerial vehicle (MAV). The key contribution is an integrated approach that simultaneously performs visual odometry (VO) and persistent tracking of a person automatically detected in the scene. All computation pertaining to VO, detection and tracking runs onboard the MAV from a front-facing monocular RGB camera. Given the gravity direction from an inertial sensor and the knowledge of the individual's height, a complete 3D trajectory of the person within the reconstructed scene can be estimated. When the ground plane is detected from the triangulated 3D points, the absolute metric scale of the trajectory and the 3D map is also recovered. Our extensive indoor and outdoor experiments show that the system can localize a person moving naturally within a large area. The system runs at 17 frames per second on the onboard computer. A walking person was successfully tracked for two minutes and an accurate trajectory was recovered over a distance of 140 meters with our system running onboard.**

I. INTRODUCTION

Rapid growth in off-the-shelf micro aerial vehicle (MAV) technology has created unprecedented interest in aerial photography and filming in recent years. Quadrotor MAVs with gimballed cameras are being used for first-person view (FPV) flying and photography by both professionals and amateurs to record sports action footage from unique viewpoints that was impossible a few years ago. However, safely manoeuvring a quadrotor MAV to film a moving person is a challenging task. Currently, this is done by a skilled radio control pilot trained to navigate the MAV often by viewing FPV video from an onboard camera streaming over a radio communication link.

Recently, various methods for user-friendly person-following MAVs are being developed (e.g., Airdog, Hexo+, 3D Robotics IRIS). These systems rely on GPS, inertial sensors on the person's body to transmit the person's location to the MAV over a wireless link. However, such methods are infeasible in GPS-denied environments (e.g., indoors) and can be error prone. Furthermore, wearable accessories can sometimes be an inconvenience to the person.

Autonomous navigation for MAVs using vision as the primary sensor [1], [2] is currently an active area of research which is building upon advances in visual odometry (VO) [3]–[6], visual SLAM [7], [8] and visual-inertial navigation systems [9], [10] for monocular and stereo cameras. These advances have led to significant progress towards real-time onboard flight stabilization and control [11], environment-mapping and obstacle avoidance [12] and online 3D reconstruction [13]. However, tasks concerning the automatic detection and 3D reconstruction of moving objects in natural scenes and interaction between autonomous MAVs and humans have not been extensively addressed so far.



(a) Close-up views (left: frame 600, right : frame 1100)

(b) Estimated trajectories and sparse 3D point cloud reconstruction

Fig. 1. Our system estimates the 3D trajectory of a moving person from a monocular camera onboard a quadrotor MAV. This result is from the MFLY-09 sequence (1800 frames) captured from the MAV in flight. (a) Close-ups of two selected frames. (b) Top view of the camera and person trajectories (in red and black respectively) and 3D points obtained from visual odometry (transition from blue to red indicates greater height above ground).

Appearance-based object detection and tracking [14]–[16] in video is a well studied problem in computer vision [17]. Progress in this area has led to applications in real-time systems for autonomous driving [18] and visual servoing for person-following on MAVs and UAVs [19], [20]. However, the topic of estimating full 3D trajectories of targets tracked from moving cameras is less explored. Existing methods either require RGB-D sensors [21] or laser range finders [22] which are suitable for autonomous vehicles [23] but not quadrotor MAVs which have limited payload capacity.

In this paper, we present a pure vision-based technique to localize a person in real-time from a monocular camera

[1] Hyon Lim is with the Dept. of Mechanical and Aerospace Engineering, Seoul National University, Seoul, South Korea, hyonlim@snu.ac.kr

[2] Sudipta N. Sinha is with the Interactive Visual Media group, Microsoft Research, Redmond, USA. sudipsin@microsoft.com

onboard a quadrotor MAV (see Fig. 1). Our method does not require sensors, artificial fiducial markers, color patches or wearable devices on the person. Instead, the proposed system uses a feature-based VO algorithm to recover the 6-DoF pose of the camera and simultaneously performs appearance-based tracking of a moving person. Using the estimated camera poses, the 2D detections, the gravity direction from the inertial sensor and the knowledge of the person's height, the full 3D trajectory of the person within the sparse 3D map is estimated at true metric scale. Despite the general advantages of the stereo approach, a monocular approach for VO and tracking is more desirable for MAVs due to the smaller payload and lower power budget. Moreover, compact (i.e., small baseline) stereo cameras are unlikely to be effective at large depths in outdoor scenes.

Localizing a moving person from a moving monocular camera has several challenges. First, feature-based VO can suffer from scale drift due to fast camera motion, moving objects in the scene or the lack of stable features with sufficient parallax. Detection and persistent tracking of moving objects is also considerably more difficult in the monocular case as only appearance cues must be used. Moreover, monocular reconstruction of the 3D trajectory of a moving target is an ill-posed problem. We show that the person's trajectory can be recovered from their 2D detections in every frame when their height is known a priori. We propose a novel approach that combines the complementary strengths of an existing person detector trained using supervised learning [14], [24] and an adaptive tracking algorithm [15] in order to persistently track the person over long periods of time.

The main contribution of this paper is the novel system for reconstructing a 3D trajectory of a moving person from a moving camera. This involves integrating real-time techniques for feature-based VO and extending tracking algorithms for persistent tracking of a person in video. To the best of our knowledge, this is the first work describing such a system for a low-flying MAV that performs all vision processing onboard from a front-facing monocular RGB camera. Our indoor and outdoor experiments confirm that the approach is robust and can accurately estimate long trajectories of a person moving within a large scene.

II. RELATED WORK

Our visual odometry (VO) sub-system relies on recovering the 6-DoF pose of the moving camera from vision and is closely related to [3], [4] and feature-based visual SLAM [7]. More recently, direct methods for VO [5], [6] have been proposed but the feature-based approach is more suited to our setting since the feature tracks from VO can also be reused for person tracking. Image-based localization from structure from motion point clouds [25] as well as visual-inertial navigation systems [9] have also been used for state estimation on MAVs [26]. MAVs relying on onboard real-time vision-based stabilization, control and navigation [1], [10]–[12] for autonomous flight have also been demonstrated.

There is a long history of tracking-by-detection approaches that have been developed for tracking pedestrians using appearance-based features [14] and cascaded classifiers [24]. In parallel, there has been progress on algorithms for adaptive tracking of arbitrary objects in video given an object template. Struck [15] and TLD [16] are amongst the top performing algorithms on a popular tracking benchmark [17]. TLD interleaves tracking, learning and detection tasks whereas Struck incorporate margin-based online learning techniques for higher robustness. However, most trackers do not address the issue of recovering from failure. Combined tracking and detection methods have been proposed for vehicle detection for autonomous driving [18].

Joint SLAM and moving object tracking (SLAMMOT) has been explored in the area of autonomous driving using range sensors [22]. Closely related techniques have been developed for detection and tracking of moving object (DATMO) [27] that have been extended to stereo cameras [28]. Motion segmentation provides additional cues to detect moving objects. In [23], joint monocular VO and tracking was used to reconstruct moving cars in the scene.

Reliably detecting and tracking objects using appearance has been applied to persistant target localization from small UAVs [20], [29], [30]. However, these methods either involve UAVs at high altitude or require downward-facing cameras, whereas our work focuses on the situation involving quadrotor MAVs with a front-facing camera flying a few meters above the ground which is more challenging.

Visual Servoing is the most common approach for target-following. Efficient onboard approaches for person-following have been proposed using TLD on an ARDrone [19]. However, flight manoeuvres may be limited in such systems due to lack of enough information about the 3D target location. In contrast, our technique estimates the full 6-DoF pose estimate of the MAV as well as the 3D position of the target. This can be used to plan more complex manoeuvres in real-time and gives more flexibility for recording aerial video.

III. VISUAL ODOMETRY

In this section, we describe our monocular feature-based visual odometry approach. Although, similar to existing methods [3], [4], [7], [8], the specific algorithms for feature tracking, relative pose estimation, keyframe selection and bundle adjustment differ from common VO methods especially in the way the individual components are combined. Recently, semi-direct and dense methods for VO have been proposed [5], [6]. However, feature-based VO is more suited for our task for two reasons. First, the feature tracks are reused for person tracking. Second, the computational burden of simultaneous VO and tracking in our system leads to frame-rates that vary between 15–20Hz. This produces large inter-frame motion which is not suitable for direct methods.

A. Keypoint detection and tracking

The first step in our feature-based VO approach involves detecting features and tracking them in consecutive frames. Keypoints are extracted using a Harris corner detector on every frame. Given a feature budget of 2000 features, candidates are selected in decreasing order of cornerness scores and grid-based heuristics are also used to ensure a good spatial distribution of features in the image. Each keypoint is described by a 256-bit BRIEF descriptor [31]. On our platform, these binary descriptors can be compared very efficiently since Hamming distances between binary strings can be computed using *popcnt*, a dedicated CPU instruction.

On the first frame, all features are added into the active feature table. On the next frame, each active feature is compared to feature candidates that lie within a $k \times k$ pixel region around the active feature's location in the previous frame. We set $k = 64$ pixels. Given the typical frame-rate of our system, this search range is wide enough to handle large interframe 2D motion caused by fast camera movement. A coarse grid is used as a search index to speed up the 2D range search. Next, for each active feature, two nearest neighbors are computed in descriptor space and the Hamming distance ratio of the best to the second-best match is computed. The motion vectors corresponding to match candidates with a ratio smaller than 0.8 are accumulated into a 2D motion histogram. This 2D histogram image is morphologically filtered to remove spurious peaks and holes and thresholded to obtain a 2D motion bitmap.

Next, in a second pass, each potential match candidate for each active feature is tested again. If the corresponding motion vector maps to a non-empty bin in the 2D motion histogram, the candidate is selected provided it also passes a mutual consistency check [3]. The active features for which valid matches are found in this way, remain active and their tracks are updated with the new feature location and descriptor while the rest are invalidated. Finally, whenever a new keyframe is selected, the unmatched keypoint candidates in that frame are added as active features to the track table.

The idea of preemptively filtering outliers using 2D motion histograms in our work differs from RANSAC-based outlier removal used in [3], [7], [8], [25]. Feature tracking in our approach is completely independent of geometric motion estimation. This has two advantages. First, a significant amount of outliers are removed very efficiently. Second, feature matches on moving objects are retained and later reused in our system when tracking the moving person. In contrast, these matches would have been removed by RANSAC-based methods along with the random outliers.

B. Camera motion estimation

Next, we describe our camera ego-motion estimation algorithm. After the initialization step, it alternates between two states. In the first state, there is a single reference keyframe and associated 3D points. The camera pose for the current frame is estimated from those 3D points using a standard camera resection method. Occasionally, the system enters the second state where the current frame becomes a new keyframe and the camera pair for the current and previous keyframes are optimized along with the triangulated 3D points. After this the system returns to the first state.

Initialization and Model Selection. In our method, the first frame automatically becomes a keyframe with canonical pose. For each subsequent frame, the camera pose relative to the first keyframe is computed after retrieving the respective two-view matches from the track table. The 5-point algorithm [32] is used with RANSAC to estimate the essential matrix. Let I_E denote the set of epipolar inliers. A homography is then robustly fitted to those inliers. The inliers to the homography are denoted by I_H. Next, a model selection score $S = 100(|I_E| - |I_H|)/|I_E| + \ln(|I_E|)$ is computed. When the number of epipolar inliner is small, the first term dominates whereas when the number of inliers is larger (e.g. 500+), then the second $\ln$ term boosts the score even when the first term is small (i.e. the numerator is small). When S exceeds a threshold T_S $(= 55)$, we proceed with the initial 3D reconstruction. A full bundle adjustment (BA) over the two views is then performed, producing an initial reconstruction of the two cameras and triangulated 3D points in a metric coordinate frame where the camera baseline is scaled to unit length. The current frame is then turned into the new reference keyframe.

If the active feature count falls below a threshold $(= 500)$ before the initial reconstruction succeeds, a new keyframe is created. In that case, the camera motion relative to the previous keyframe is considered a pure rotation and no 3D points are initialized. Our system continues to remain in the bootstrapping stage until the model selection test succeeds.

Absolute camera pose estimation. After the system is bootstrapped and a reference keyframe with 3D points is available, the camera pose for the next few frames is obtained using standard camera resectioning. First, 2D–3D matches are retrieved by querying the track table with the feature-ids of the 3D points in the reference keyframe. Next, the 3-point algorithm is used within a RANSAC framework to estimate the camera's pose. Finally, the pose parameters are refined in a final nonlinear optimization step.

Keyframe selection and scale propagation. Feature tracking and pose estimation continues as described above until either the number of 2D–3D match inliers falls below a minimum count (=25) or the inter-camera distance between the current and reference keyframe exceeds a threshold (=1.0). In both cases, a new keyframe is added. Then, bundle adjustment is performed on the current and previous keyframes, followed by the model selection test described above. If the test is successful, the relative scale between the 3D points common between the previous keyframe and the new set of 3D points is estimated. This is done using a 1D RANSAC to estimate the scale factor for which the pixel reprojection error of the transformed old 3D points in the current frame is minimized. Using image measurements reduces the effect of depth uncertainty in the triangulated 3D points. Using the recovered scale, the new 3D points and camera are transformed into the coordinate frame of the previous keyframe using the estimated similarity transform. The current frame is then turned into the reference keyframe.

System reset. We build resilience to failure using an approach similar to the firewall strategy discussed in [3]. During scale propagation, the scale estimate can be inaccurate when enough common 3D points between the previous and current keyframe are not present. We ensure that the inlier-count and the inlier percentage are both above acceptable thresholds and that the estimated scale factor is within an acceptable range (0.02 – 50). When scale propagation fails, the system resets and starts bootstrapping from scratch. After this reset, the new 3D points have an arbitrary scale and this can cause drift. However, in practice the reset occurs quickly enough that the depth distribution in the scene before and after the reset are similar. In such cases, the model selection test during initialization provides a weak normalization and that prevents large scale drift.

Recovering Absolute Scale. A monocular system cannot recover the true scale of the camera trajectory. Since the camera is calibrated, by assuming that the person with known height is in an upright position, a 2D person detection in the image can be used to recover the true camera height above the ground plane (see section V for details). We also estimate the ground plane from the triangulated 3D points. Using the known gravity vector g, the ground plane is detected by building a histogram of the 1D projections of the 3D points along g and finding the bin that received the most votes. Since it is safe to assume that the camera is higher than the ground plane, we only use the 3D points lower than the camera center during this step. When both the ground plane and the person is detected, the VO coordinate frame can be rescaled to the true dimensions. Currently, we only perform this scaling during the VO initialization step and after the system resets. However, incorporating these constraints into a global bundle adjustment during an optional post-processing step could improve the accuracy of the estimated trajectories.

IV. TRACKING THE PERSON

Persistent tracking over long duration from a moving MAV has several challenges – the scale of the target in the image or the scene illumination may change drastically, for example, when the person walks from a sunlit to a shaded area. Our system also needs to be robust to occlusions, pose deformations and large inter-frame motion caused by fast camera or subject movement. We tested two existing alternatives – tracking-by-detection and adaptive tracking, neither of which alone could meet the accuracy and runtime requirements in our experiments. Next, we describe our hybrid approach that combines the strengths of the two approaches and further optimizations that helped to reduce the running time.

Person Detector. For detecting a person independently in each frame we use multiscale channel features [14] and a boosted object detection method. The weak classifiers are 4096 two-level decision trees. Adaboost is used for supervised learning on the CalTech Pedestrian benchmark [33]. Cascade classifier are commonly used in fast object detectors. We use the *soft cascade* variant [24] that allows a tradeoff between accuracy and speed. Instead of using multiple cascade layers, a threshold is used after evaluating each decision tree and unpromising patches are rejected early after evaluating a small number of trees.

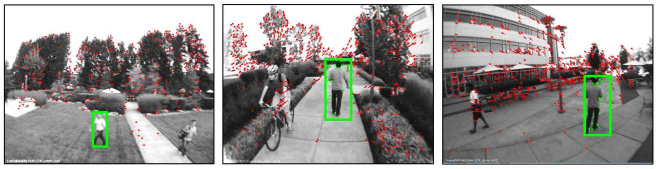


Fig. 2. Our system is robust and continues to track the same person even when other individuals appear in the scene. Results on selected frames from the WALK-02 and MFLY-08 sequences are shown.

Adaptive Tracking. Unlike detectors that require offline supervised learning on labeled data, adaptive trackers [15], [16] can track arbitrary objects using online learning on a small number of training examples. Struck [15] and TLD [16] are two such algorithms with excellent performance on tracking benchmarks [17]. However, it is worth noting that most of their test sequences are short and recorded from static cameras. We used the Struck tracker in our pipeline but another adaptive tracking method could also have been used. In Struck, 2D translational tracking is posed as a structured output prediction problem. Unlike prior template tracking methods, it incorporates background appearance and updates a discriminative model during online learning which gives it higher robustness. It also uses a budgeting scheme to maintain a small number of support vectors crucial for good runtime performance. However, Struck and other adaptive trackers requires good initialization and re-initialization, often obtained from object detection.

Our Hybrid Approach. Due to its ability to recover from failure, tracking-by-detection is often better for systems that need long duration tracking. However, running detection independently on every frame gives noisy results with jitter in 2D position and scale. To address this, we use dynamic programming commonly used in 1D sequence labeling tasks.

Let $\{b_i^t\}$ denote multiple detection hypotheses in frame t and $U(b_i^t)$ be the cost of selecting the i-th hypothesis in the t-th frame. Let $V(b_i^t, b_j^{t+1})$ be the cost of selecting the i-th and j-th hypotheses in frames t and $t+1$. Dynamic programming (DP) efficiently finds the optimal labeling $\mathcal{L}^*$ that minimizes the energy function $E(\mathcal{L})$ defined as follows.

$$E(\mathcal{L}) = \sum_{t=1}^{N} U(b_i^t) + \sum_{t=1}^{N} V(b_i^t, b_j^{t+1}) \tag{1}$$

$$V(b, b') = \lambda_1 V_1(b, b') + \lambda_2 V_2(b, b') + \lambda_3 V_3(b, b') \tag{2}$$

where selecting hypotheses b and b' in consecutive frames incurs different pairwise costs: $V_1(b, b') = 1 - \frac{b \cap b'}{b \cup b'}$ penalizes b and b' with low overlap, V_2 penalizes distances between their centers whereas V_3 penalizes a difference in their sizes. λ_1, λ_2 and λ_3 are weights that are chosen empirically. Our system runs online by caching intermediate results computed on the previous frame, the solution for the current frame is obtained recursively from the cached results.

Our DP-based detection approach suppresses false positives but fails to address the issue of false negatives. For instance, our detector often had low recall when the person was farther away or when the camera was tilted sideways.

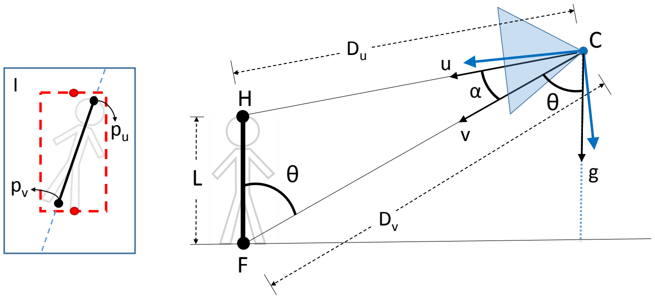


Fig. 3. (Left) Camera roll induces a tilt which can be corrected to calculate p_u and p_v the head and foot pixels in the image. This assumes that the person is in an upright position. (Right) The known gravity direction (g) and the person height L allows the D_v to be computed in closed form.

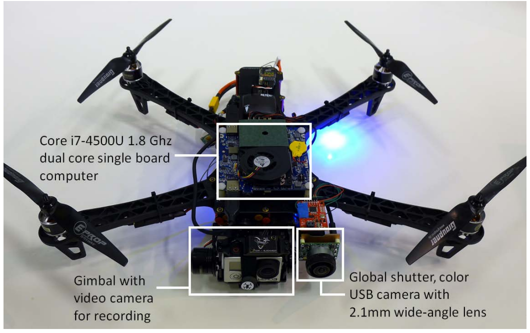

Core i7-4500U 1.8 Ghz dual core single board computer
Gimbal with video camera for recording
Global shutter, color USB camera with 2.1mm wide-angle lens

Fig. 4. Our Quadrotor MAV platform with additional payload – an USB camera and a dual-core single board computer for onboard processing. Separate camera on a gimbal can record high resolution video.

To address this issue, we developed a heuristic that lets our system temporarily switch to adaptive tracking (using Struck) when the total number of detection hypotheses falls below a threshold. This threshold is chosen conservatively to ensure that Struck is initialized with a good detection result. Setting the threshold too low increases the risk that the last detection is inaccurate preventing the Struck tracker from being initialized with an accurate template. While the adaptive tracking is enabled, we continue to speculatively run the detector on every frame. When the number of detection hypotheses increases above the threshold, we again enable the detector after disabling the Struck tracker.

Optimizations. We propose three optimizations to improve runtime performance. First, when detection is enabled, we use the previous detection's bounding box b to predict a smaller region by enlarging b by a factor of two and run the detector on this region in the current frame rather than in the whole frame. This produces a 3× speedup on average.

Although the detector mostly runs on cropped regions, after a failure, it needs to be run on the whole frame. Our second optimization involves scheduling this computation over successive frames. Rather than processing the whole frame at once, we schedule the detection over four overlapping image bands 200 × 480 pixels each, and cycle through them on four consecutive frames. Typically in these situations, the system has adaptive tracking enabled and detection is only being run in speculative mode. Therefore, a little delay in obtaining a good detection does not hurt the system's accuracy.

Finally, frames where bundle adjustment is performed can be bottlenecks in our system. To reduce the latency, we avoid running our hybrid person tracker on these frames. Instead, we predict the detection in the current frame based on the detection in the previous frame and the mean translation vector computed from the tracked keypoints that lie within the person bounding box. This is possible because our keypoint tracking approach can recover short but accurate keypoint tracks even on non-rigid objects. This optimization does require that the previous frame had a valid detection and the tracked keypoints within the person bounding box exceeds a minimum count (= 20).

V. PERSON TRAJECTORY RECONSTRUCTION

We propose a simple closed form method to estimate the distance from the camera C at an unknown height above the ground to the person's foot by assuming the person is upright and that the person's height L, the camera intrinsics and the gravity vector g are known. As shown in Fig. 3, let u and v denote the backprojected rays corresponding to p_u and p_v, the roll-corrected head and foot pixels in the image respectively and let D_u and D_v denote the distances from the camera C to the person's head H and foot F respectively. Using the law of cosines on $\triangle CHF$ in Fig. 3, we get

$$L^2 = D_u^2 + D_v^2 + 2D_u D_v cos(\alpha). \qquad (3)$$

Since the camera is calibrated and g is known, angles α and θ are both known. From basic trigonometry, we can obtain a related between D_u and D_v as follows.

$$D_u cos(\alpha) + Lcos(\theta) = D_v \qquad (4)$$

Substituting D_u from (4) into (3) gives a quadratic equation in D_v which has two real roots but only one that is positive. The negative root is simply discarded.

The final trajectory is estimated by an extended Kalman filter (EKF). The state vector of the EKF consists of 3D position and velocity of the camera and the person's foot. The input measurements to the EKF are the estimated camera position from VO, the normalized image coordinates of the person's foot and estimated value of D_v. Details are provided in the supplementary material [34].

VI. SYSTEM AND IMPLEMENTATION

For our experiments, we use a custom built quadrotor MAV shown in Fig. 4. Our system is implemented in C++ and runs on the single board computer equipped with a Core i7-4500U CPU (1.8 Ghz dual-core), 16GB RAM and a 240GB mSATA SSD drive. The board runs 64-bit Windows 8.1, weighs 183 grams without the protective case and runs on a 12V power source. A global shutter color camera with a wide-angle lens (2.1mm focal length) is mounted front-facing on the MAV and connected via USB to the computer. The maximum framerate is 30Hz at a resolution of 640 × 480 pixels. The camera intrinsics were calculated offline using the omnidirectional camera calibration toolbox [35].

Name	F	V (m/sec.)	$Succ$ (%)	T_{track} (msec.)	T_{total} (msec.)
WALK-O1	1400	3.7 ± 1.6	85.4	25 ± 10	63 ± 9
WALK-O2	1590	3.6 ± 0.9	92.8	18 ± 14	55 ± 16
WALK-O3	2205	5.3 ± 2.3	93.1	17 ± 12	60 ± 14
WALK-I4	1180	2.6 ± 0.9	82.4	21 ± 7	56 ± 9
WALK-I5	650	6.8 ± 4.0	86.6	31 ± 18	69 ± 18
MFLY-O6	1160	0.9 ± 0.4	73.9	12 ± 5	46 ± 10
MFLY-O7	1200	1.7 ± 0.9	98.5	19 ± 18	56 ± 17
MFLY-O8	1370	2.2 ± 0.9	96.1	16 ± 12	56 ± 15
MFLY-O9	1800	2.1 ± 0.5	82.2	24 ± 21	59 ± 22
ONBRD1	816	5.4 ± 1.6	–	26 ± 11	68 ± 11
ONBRD2	2110	3.5 ± 1.2	–	18 ± 8	57 ± 12

TABLE I

#FRAMES (F), PERSON'S ESTIMATED VELOCITY (V), OUR TRACKER'S SUCCESS-RATE ($Succ$), PER-FRAME TIMINGS FOR TRACKING ONLY (T_{track}) AND PER-FRAME TIMINGS (T_{total}).

VII. EXPERIMENTS

We first performed an independent evaluation of our VO and tracking algorithms focusing on robustness and runtime performance and then evaluated the system end-to-end on nine sequences acquired onboard the MAV. Finally, we tested our real-time system onboard the flying MAV in two different sessions. Selected results are described in the paper. The complete set of results can be seen on the website [34].

Datasets. The nine sequences used for offline evaluation are denoted WALK-[o1–o3], WALK-[i4, i5] and MFLY-[o6–o9] in the paper. The letters {o,i} indicate scene-type (outdoors vs. indoors). The WALK sequences were acquired from a hand-held quadrotor carried around to simulate person-following, whereas the MFLY sequences were captured from the quadrotor in flight. The image resolution is 640 × 480 pixels and synchronized attitude readings are obtained from an onboard IMU. These sequences were manually annotated to obtain the ground truth person detections in every frame and can be found on the project website [34].

Qualitative Results. The average framerate of our system is 17 frames per second. Table I summarizes timings on individual sequences and reports our tracker's success rate, i.e. the percentage of frames where the 2D detections had an overlap error less than 0.5. Fig. 6 and 7 show results on the WALK-o1 and WALK-i4 sequences. Here, the MAV was carried hand-held 1.5–2 meters above the ground while following the person at close range. In WALK-O1 the camera follows behind the person walking on a steady path whereas in WALK-I4, the person changes direction frequently and is observed in the camera from many different directions. Fig. 1 shows results on the MFLY-o9 sequence where the MAV flies 1–4 meters above the ground causing higher variance in the size of the person in the image.

A. Evaluation of Visual Odometry

We compared our VO system with the authors' implementation of SVO [6] on one of their 30 fps sequences with small inter-frame camera motion[1]. The result from SVO and our system are comparable as shown in Fig. 5(a) and Fig. 5(b) respectively. Next, we simulated larger camera motion by skipping frames and selecting every 8-th frame of this sequence. This time SVO failed to update the camera pose after 832 frame while our system processed the complete input and generated a result (Fig. 5(c)) visually similar to the original result (Fig. 5(b)). The robustness of our VO system was tested on all nine offline sequences where it produced visually accurate camera trajectories. For example in the WALK-o1 sequence, the person returns approximately to the same location he started from. Fig. 6 shows the low drift in our result despite the long trajectory (87 meters).

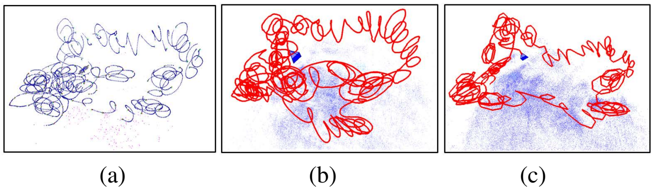


Fig. 5. (a) Camera path estimated by SVO [6]. (b) by our method on all 4872 frames. (c) The result of our method when every 8-th input frame is used. SVO failed to compute a valid camera pose after 832 frame.

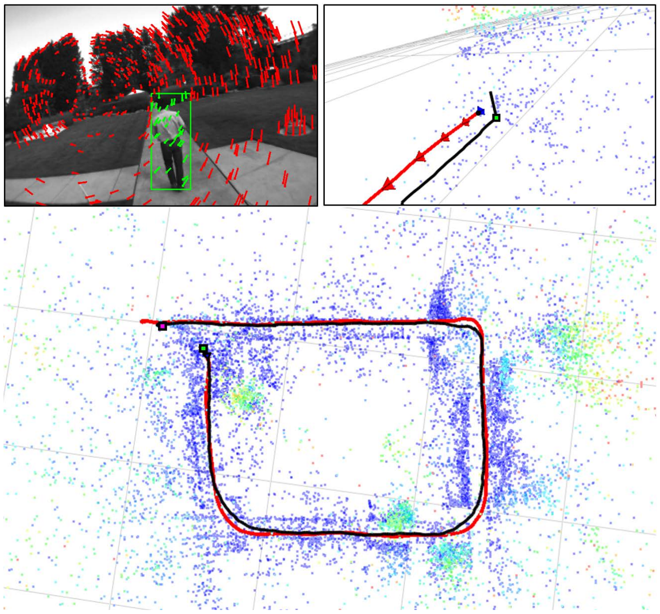


Fig. 6. Results on WALK-o1 (1400 frames): [Top row] Feature tracks on a selected frame with large inter-frame camera rotation. Top view of reconstructed camera and person trajectories (shown in red and black respectively). Here, the person returns to the starting location. The trajectory recovered by our open-loop VO system shows fairly low drift in this case.

B. Evaluation of Tracking and Detection

To evaluate the tracking performance, we used the standard overlap error metric [17], $O = 1 - \frac{b \,\cap\, b_{gt}}{b \,\cup\, b_{gt}}$, where b and b_{gt} denote the detected and the ground truth rectangles respectively. We compare our method with the detector [14], [24] and Struck [15] (initialized using the detector) running at full and half image resolution. Fig. 8(a) shows the cumulative histogram of the overlap errors for the four methods on the MFLY-o9 sequence. The percentage of detections with

[1] the airground_rig_s3_2013-03-18-21-38-48 sequence (4872 frames)

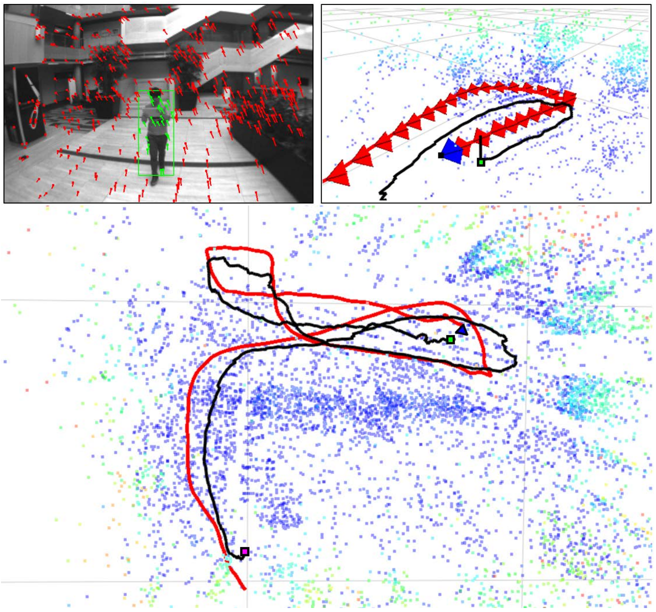


Fig. 7. Results on (WALK-i4) (1180 frames): [Top] Selected frame and close-up view of the trajectory in a large indoor scene. [Bottom] Top view of camera and person trajectories (in red and black respectively).

error less than 0.5 was 88% for the detector, 80% for our method and much lower for Struck. Fig. 8(c) compares the cumulative histograms of the *tracking only* per-frame timings. The percentage of frames where timing was less than 40 msec. was 100% for Struck, 70% for our method and 20% for the detector. Thus, our method achieves a good tradeoff between speed and accuracy. Using thresholds for error = 0.5 and timing = 40 msec., the error and timing metrics for all nine datasets are summarized in Fig. 8(b) and Fig. 8(d). Our method is the most accurate on four out of the nine sequences whereas the baseline detector is the most accurate method on the other five (our method is a close second). However, the baseline detector is consistently very slow. While Struck is the fastest, it fails catastrophically on four of the sequences. In summary, our hybrid approach achieves the best compromise between speed and accuracy.

C. Onboard system evaluation

Finally, we tested our system running onboard our flying quadrotor MAV during two outdoor flight sessions – ONBOARD-1,2 lasting approximately one and two minutes respectively. The average framerate during these sessions was 15 and 17 frames per second respectively. Fig. 9 shows results from the ONBOARD-2 experiment where the person walked a distance of 140 meters. The estimated trajectories and 3d reconstruction are logged onboard and visualized later on. Fig. 9(a) and 9(b) shows close-ups and Fig. 9(c) shows the top-view of the trajectories and the reconstruction. The supplementary video shows our system in action. During this experiment, the person also carried an accurate INS device (GPS1) and an ordinary GPS logger recording at 1Hz (GPS2). We treat the GPS1 track as ground truth. We globally aligned the GPS2 track and our estimated person trajectory to GPS1 in a least square sense. The mean 2D Euclidean distance error for GPS2 and our system was 2.57 ± 1.87 meters and 3.60 ± 3.22 respectively.

VIII. CONCLUSIONS

We have presented a system for estimating the trajectory of a moving person from a monocular camera onboard a low-flying quadrotor MAV. We perform VO and persistent person tracking by building on top of existing algorithms in both areas. The system runs at 17 frames per second on an onboard computer that consumes low power. Extensive evaluation on long sequences in indoor and outdoor scenes demonstrates that the system is robust and effective.

However, our system has some limitations. Our tracking approach currently requires the detector to succeed on a majority of frames. It can be inaccurate if the person is difficult to recognize depending on his pose. Like all VO system, ours assumes that the background scene is mostly static and dynamic scenes with water cannot be handled.

As future work, building a controller that uses the real-time camera and person position estimates from our method will enable autonomous navigation and person-following quadrotor MAV in both indoor and outdoor scenes. We expect that the trajectories obtained using our proposed method can be exploited for real-time path planning and will make it more flexible to spatially position and control aerial cameras for recording interesting action footage.

REFERENCES

[1] L. Meier, P. Tanskanen, L. Heng, G. H. Lee, F. Fraundorfer, and M. Pollefeys, “Pixhawk: A micro aerial vehicle design for autonomous flight using onboard computer vision,” *Autonomous Robots*, 2012.

[2] J. Engel, J. Sturm, and D. Cremers, “Accurate figure flying with a quadrocopter using onboard visual and inertial sensing,” in *Workshop on Visual Control of Mobile Robots (ViCoMoR) at IROS*, Oct. 2012.

[3] D. Nistér, O. Naroditsky, and J. R. Bergen, “Visual odometry,” in *CVPR*, 2004, pp. 652–659.

[4] D. Scaramuzza and F. Fraundorfer, “Visual odometry [tutorial],” *IEEE Robot. Automat. Mag.*, vol. 18, no. 4, pp. 80–92, 2011.

[5] J. Engel, J. Sturm, and D. Cremers, “Semi-dense visual odometry for a monocular camera,” in *ICCV*, December 2013.

[6] C. Forster, M. Pizzoli, and D. Scaramuzza, “SVO: Fast Semi-Direct Monocular Visual Odometry,” in *ICRA*, 2014.

[7] G. Klein and D. Murray, “Parallel tracking and mapping for small AR workspaces,” in *ISMAR*, Nara, Japan, November 2007.

[8] H. Lim, J. Lim, and H. J. Kim, “Real-time 6-dof monocular visual SLAM in a large-scale environment,” in *ICRA*, 2014, pp. 1532–1539.

[9] A. Mourikis and S. Roumeliotis, “A multi-state constraint kalman filter for vision-aided inertial navigation,” in *ICRA*, 2007, pp. 3565–3572.

[10] S. Weiss, M. Achtelik, S. Lynen, M. Chli, and R. Siegwart, “Real-time onboard visual-inertial state estimation and self-calibration of MAVs in unknown environments,” in *ICRA*, 2012, pp. 957–964.

[11] M. Achtelik, M. Achtelik, S. Weiss, and R. Siegwart, “Onboard IMU and monocular vision based control for MAVs in unknown in- and outdoor environments,” in *ICRA*, 2011, pp. 3056–3063.

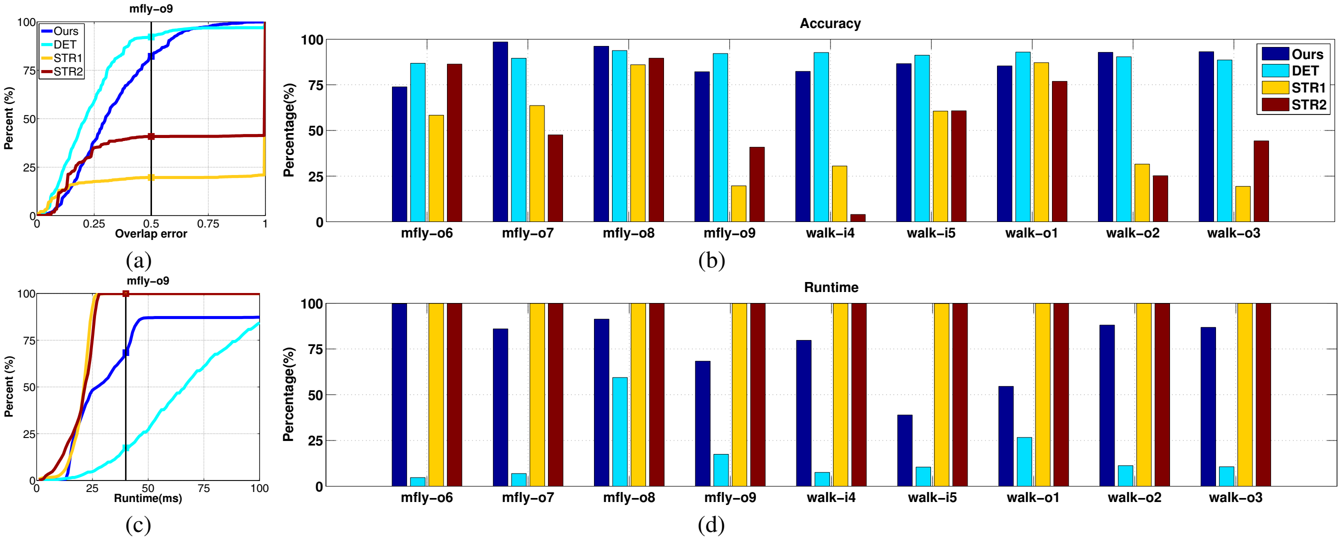
[12] L. Heng, D. Honegger, G. H. Lee, L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys, “Autonomous visual mapping and exploration with a micro aerial vehicle,” *J. Field Robotics*, vol. 31, no. 4, 2014.

[13] A. Wendel, M. Maurer, G. Graber, T. Pock, and H. Bischof, “Dense reconstruction on-the-fly,” in *CVPR*, 2012.

[14] P. Dollár, Z. Tu, P. Perona, and S. Belongie, “Integral channel features,” in *BMVC*, 2009, pp. 1–11.

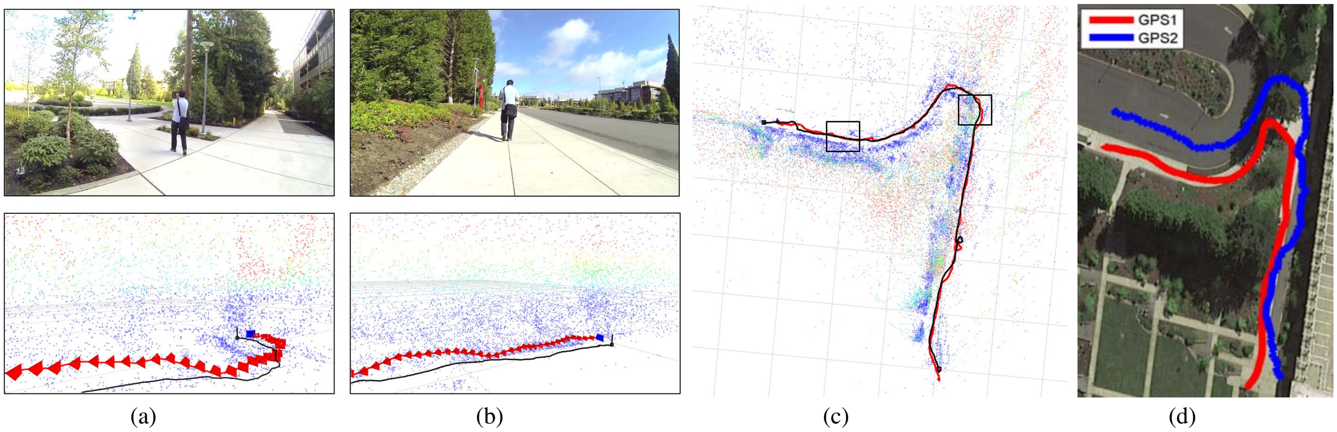
[15] S. Hare, A. Saffari, and P. H. S. Torr, “Struck: Structured output tracking with kernels,” in *ICCV*, 2011, pp. 263–270.

[16] Z. Kalal, K. Mikolajczyk, and J. Matas, “Tracking-learning-detection,” *IEEE TPAMI*, pp. 1409–1422, 2012.



(a) (b) (c) (d)

Fig. 8. Quantitative evaluation for tracking: (a) Cumulative histogram plot of overlap error for the four methods (DET: Tracking-by-detection STR1: Struck on full resolution image, STR2: Struck on half resolution image) on the MFLY-o9 sequence. (b) The percentage of frames where tracking was successful (the overlap error was less than 0.5) shown for the four methods. (c) A similar cumulative histogram plot for tracking-only per-frame timings for MFLY-o9. (d) The percentage of frames where tracking took less than 40 msec. shown for the four methods.



(a) (b) (c) (d)

Fig. 9. ONBOARD EXPERIMENT: (a,b) Two selected frames from the camera on gimbal and close-up views of the trajectories and reconstruction. The camera and person trajectories are shown in red and black respectively. (c) Top view of the full reconstruction and trajectories. The black squares correspond to the areas shown in the close-up views. (d) GPS tracks from an accurate INS (GPS1) and an off-the-shelf consumer-grade GPS tracker (GPS2).

[17] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *CVPR*, June 2013, pp. 2411–2418.

[18] C. Caraffi, T. Vojir, J. Trefny, J. Sochman, and J. Matas, "A system for real-time detection and tracking of vehicles from a single car-mounted camera," in *Intelligent Transportation Sys. (ITSC)*, 2012, pp. 975–982.

[19] J. Pestana, J. Sanchez-Lopez, S. Saripalli, and P. Campoy, "Computer vision based general object following for GPS-denied multirotor unmanned vehicles," in *ACC*, 2014, pp. 1886–1891.

[20] V. Dobrokhodov, I. Kaminer, K. Jones, and R. Ghabcheloo, "Vision-based tracking and motion estimation for moving targets using small UAVs," in *ACC*, June 2006.

[21] T. Naseer, J. Sturm, and D. Cremers, "Followme: Person following and gesture recognition with a quadrocopter," in *IROS*, 2013.

[22] C.-C. Wang, C. Thorpe, M. Hebert, S. Thrun, and H. Durrant-Whyte, "Simultaneous localization, mapping and moving object tracking," *IJRR*, vol. 26, no. 6, June 2007.

[23] A. Kundu, K. M. Krishna, and C. Jawahar, "Realtime multibody visual slam with a smoothly moving monocular camera," in *ICCV*, 2011.

[24] L. Bourdev and J. Brandt, "Robust object detection via soft cascade," in *CVPR*, vol. 2, 2005, pp. 236–243.

[25] H. Lim, S. N. Sinha, M. F. Cohen, and M. Uyttendaele, "Real-time image-based 6-dof localization in large-scale environments," in *CVPR*, 2012, pp. 1043–1050.

[26] S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar, "Vision-based state estimation for autonomous rotorcraft MAVs in complex environments," in *ICRA*, 2013.

[27] T.-D. Vu, O. Aycard, and N. Appenrodt, "Online localization and mapping with moving object tracking in dynamic outdoor environments," in *Intelligent Vehicles Symposium, 2007 IEEE*, 2007, pp. 190–195.

[28] K.-H. Lin and C.-C. Wang, "Stereo-based simultaneous localization, mapping and moving object tracking," in *IROS*, 2010, pp. 3975–3980.

[29] J. E. Gomez-Balderas, G. Flores, L. R. García Carrillo, and R. Lozano, "Tracking a ground moving target with a quadrotor using switching control," *J. Intell. Robotics Syst.*, vol. 70, no. 1-4, pp. 65–78, 2013.

[30] E. M. Céline Teuliere, Laurent Eck, "Chasing a moving target from a flying uav," in *IROS*, 2011, pp. 4929–4934.

[31] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary robust independent elementary features," in *ECCV*, 2010, pp. 778–792.

[32] D. Nistér, "An efficient solution to the five-point relative pose problem," *TPAMI*, vol. 26, no. 6, pp. 756–777, June 2004.

[33] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *TPAMI*, vol. 34, no. 4, 2012.

[34] H. Lim and S. N. Sinha, "Project website of Monocular Localization of a moving person onboard a Quadrotor MAV," http://research.microsoft.com/en-us/um/redmond/groups/ivm/mavloc/, 2014, [Online; accessed 26-Feb-2015].

[35] D. Scaramuzza, A. Martinelli, and R. Siegwart, "A toolbox for easily calibrating omnidirectional cameras," in *IROS*, 2006, pp. 5695–5701.