Heads Up and Camera Down: A Vision-Based Tracking Modality for Mobile Mixed Reality

Stephen DiVerdi, *Student Member*, *IEEE*, and Tobias Höllerer, *Member*, *IEEE*

Abstract—Anywhere Augmentation pursues the goal of lowering the initial investment of time and money necessary to participate in mixed reality work, bridging the gap between researchers in the field, and regular computer users. Our paper contributes to this goal by introducing the GroundCam, a cheap tracking modality with no significant setup necessary. By itself, the GroundCam provides high frequency and high resolution relative position information similar to an inertial navigation system but with significantly less drift. We present the design and implementation of the GroundCam, analyze the impact of several design and runtime factors on tracking accuracy and consider the implications of extending our GroundCam to different hardware configurations. Motivated by the performance analysis, we developed a hybrid tracker that couples the GroundCam with a wide area tracking modality via a complementary Kalman filter, resulting in a powerful base for indoor and outdoor mobile mixed reality work. To conclude, the performance of the hybrid tracker and its utility within mixed reality applications is discussed.

Index Terms—Anywhere augmentation, vision-based tracking, motion sensing, tracker fusion, mobile augmented reality, mixed reality, wearable computing.

1 Introduction

Traditional mixed reality applications are built on a series of assumptions about the environment they will operate in, often requiring time-consuming offline measurement and calibration for model construction purposes or instrumentation of the environment for tracking. This high start-up cost limits the general appeal of mixed reality applications, creating a barrier to entry that discourages potential casual mixed reality users. The goal of *Anywhere Augmentation* [1] is to create a class of mixed reality technologies and applications that require a minimum of setup using cheap commonly available hardware, bringing the field of mixed reality within the realm of an average computer user.

The choice of tracking technology used in a mixed reality application is heavily dependent on the environment and its setup, and calibration is often one of the time consuming initial steps of application deployment. An overview of the commonly available technologies is presented in Table 1. It is apparent that no single tracking solution exists for the interesting and increasingly common case of wide area and high resolution applications such as outdoor architectural visualizations. The prevailing solution is to couple a *global* tracker such as GPS, which provides wide area, absolute, low-resolution data, with *local* tracking, for example, from inertial sensors, which provides high-resolution relative and drift prone positioning.

In this paper, we introduce the GroundCam (consisting of a camera and an orientation tracker—see Fig. 1), a local tracking technology for both indoor and outdoor applications. We use the optical flow of a video of the ground to determine velocity, inspired by the workings of an optical mouse. This is related to the visual odometry work done in the robotics community [14], [15], but here, we apply it to the much less constrained world of human tracking. By itself, the GroundCam provides high-resolution relative position information, but is subject to drift due to the integration of error over time. In Table 1, it is clear that the GroundCam most similarly resembles an inertial tracker, which measures acceleration and integrates twice to get position. The GroundCam is a significant improvement over inertial tracking because its single integration accumulates error much more slowly, maintaining similar small-scale accuracy for a longer time.

To address the GroundCam's long term drift, we use a complementary Kalman filter to combine the GroundCam with a wide area sensor such as a GPS receiver (see Fig. 1), providing better accuracy over large environments. For wide area indoor operation, we simulate the signal from a beacon-based tracker such as the Cricket [12] or Locust Swarm [11] to demonstrate the hybrid performance. These wide area trackers provide periodic stable corrections to compensate for the GroundCam's drift while maintaining its fast and high-resolution data.

The advantages of the GroundCam include its favorable performance compared to other local tracking technologies, as well as its general applicability to a variety of mixed reality applications, including outdoor mobile augmented reality and indoor virtual reality. Hybrid indoor/outdoor applications can also use the GroundCam, as it handles large changes in illumination gracefully. Finally, the low cost and ease of construction of the GroundCam make it suitable toward our goal of Anywhere Augmentation by reducing the barriers to entry for mixed reality applications.

The rest of this paper is organized as follows: Previous results are surveyed in Section 2. Section 3 details the

- *S. DiVerdi is with Adobe Systems Inc. E-mail: stephen.diverdi@gmail.com.*
- *T. Höllerer is with the Department of Computer Science, University of California, Santa Barbara, Santa Barbara, CA 93106-5110. E-mail: holl@cs.ucsb.edu.*

Manuscript received 15 July 2007; revised 7 Nov. 2007; accepted 18 Dec. 2007; published online 16 Jan. 2008.
Recommended for acceptance by B. Sherman, A. Steed, and M.C. Lin.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number TVCGSI-2007-07-0085.
Digital Object Identifier no. 10.1109/TVCG.2008.26.

1077-2626/08/$25.00 © 2008 IEEE Published by the IEEE Computer Society

TABLE 1
A Brief Comparison of Tracking Technologies for Typical Setups

technology	range (m)	setup (hr)	resolution (mm)	time (s)	environ
magnetic [2]	1	1	1	∞	in/out
ultrasound [3]	10	1	10	∞	in
inertial [4]	1	0	1	10	in/out
pedometer [5]	1000	0	100	1000	in/out
optical,					
beacons [6]	10	1	1	∞	in
passive [7]	10	10	10	∞	in
markerless [8]	10	0	10	∞	in/out
hybrid [9]	10	10	1	∞	in
GPS [10]	∞	0	1000	∞	out
beacons [11], [12]	100	10	1000	∞	in/out
WiFi [13]	100	10	1000	∞	in/out
GroundCam	10	0	1	1000	in/out

Range is the size of the region that can be tracked within. Setup is the amount of time for instrumentation and calibration. Resolution is the granularity of a single output position. Time is the duration for which useful tracking data is returned (before it drifts too much). Environ is the where the tracker can be used, indoors or outdoors. All values are expressed accurate to orders of magnitude.

implementation of the GroundCam, and its performance is carefully analyzed in Section 4. The extension of the GroundCam to work with wide field of view cameras is detailed in Section 5. Motivated by the GroundCam analysis, Section 6 describes the implementation of the hybrid tracker, and its results are presented in Section 7. Concluding remarks and avenues for future work are in Section 8.

2 Related Work

This paper is an extended version of work we presented at the Ninth International IEEE Conference on Virtual Reality [16].

Related work falls into the following main categories: optical flow-based tracking techniques and hybrid tracking approaches with a focus on pedestrian navigation.

2.1 Optical Flow-Based Tracking

Using optical flow for camera tracking has been explored in many applications. The widest commercial distribution was reached by the modern optical mouse, which uses an LED to illuminate the mouse pad surface for a small camera that tracks texture motion across the visual field, generating a translation vector. For optical mice, the problem is drastically simplified by the assumption that the entire visual field will exhibit a single coherent translation. A similar concept is implemented as part of Haro et al.'s mobile UI work [17], which uses optical flow from a cell phone camera as a 2D input to mobile application user interfaces. Many concessions must be made due to the phone's limited processing power—most importantly, the motion estimation is limited to one of four cardinal directions and very approximate measures of motion magnitude are used. Given our interest in accurate and robust tracking, we cannot make similar simplifying assumptions for the GroundCam.

Much of the previous work in camera tracking via optical flow is in the field of visual odometry for mobile robotics. A straightforward approach is taken by Lee and Song [18], mounting an optical mouse near the ground on a wheeled robot. While the mouse does provide high-quality optical flow information, the fact that it needs to be within a few millimeters of the tracked surface inherently restricts the

Fig. 1. A wearable computer setup for GroundCam tracking. We use a Unibrain Fire-i 400 camera, an InterSense InertiaCube2 orientation tracker, and a Garmin GPS 18 receiver. Inside the backpack is a Dell Precision M50 laptop.

robot to a very smooth terrain. A more sophisticated solution is to mount a camera horizontally on the robot to provide an eyelike view of the world, as in Campbell et al.'s visual odometry evaluation [14]. While their work tests the performance of visual odometry in terrain that is difficult for robots such as ice and grass the ground is still required to be flat and free of distracting influences. The GroundCam's design allows it to be used in complex terrain including obstacles and debris, significant changes in height, and other moving agents.

Se et al.'s robot [15] handles complex environments by using SIFT features and a three camera stereo system. SIFT features are matched across images from the three cameras to build a 3D map, against which features in subsequent frames are matched. While the results are impressive, the algorithm is also very demanding computationally, operating at 2 Hz and restricting their robot to a speed of 0.4 m/s. Nistér et al. [19] use stereo imagery for visual odometry for ground vehicles, with good accuracy and ability to handle distractions. However, the updates are limited to 10 Hz, and necessary temporal filtering takes advantage of the low frequency of accelerations for a ground vehicle. Human tracking requires low-latency high-frequency updates for interactive applications.

2.2 High-Quality Wide Area Tracking

None of the methods discussed so far are sufficiently robust and/or precise to work for arbitrary wide area mixed reality applications. Therefore, tracking approaches for such environments are typically of a hybrid nature.

Foxlin and Naimark [9] propose the coupling of inertial sensors with vision-based tracking of fiducial markers that have to be attached to the ceiling or walls around the

tracking area. By tracking natural features the GroundCam does not require instrumentation of the environment.

A common approach for tracking pedestrians in the outdoors is to couple GPS tracking with inertial-based dead reckoning to improve update rates and to bridge areas where GPS is unreliable [20]. Relying on inertial sensors as a direct position tracking modality is widely deemed to be of limited use, however, because of the rapid propagation of drift errors due to double integration [4]. Instead, many systems employ inertial sensors as a pedometer, detecting the event of the user taking a step [5], [20], [21]. For indoor navigation, Lee and Mase couple step-detection via inertial sensors with infrared beacons for absolute measurements [22]. For all these hybrid tracking techniques, the GroundCam provides an additional dead-reckoning sensor that could improve accuracy and reliability of the position tracking.

As hybrid tracking systems are often used to address the limitations of individual tracking modalities, there has been extensive research into techniques for optimally coupling these sensors. Foxlin [23] originally used a complementary separate-bias Kalman filter to combine gyroscopes, inclinometers, and a compass, while You and Neumann [24] use an extended Kalman filter with separate correction steps for vision and gyroscope updates. Jiang et al. [25] combine vision and gyroscope sensors in a more heuristic manner—the gyroscope measurement is used as an initial estimate to limit the vision feature search, and the vision measurement is used to limit the gyroscope drift. Finally, while coupling between sensors is often loose, there is also work in tightly coupled sensors such as GPS/inertial hybrids [26]. For our hybrid tracker, we loosely couple the GroundCam and GPS units for modularity and simplicity of design.

The GroundCam as a novel local tracking modality provides continuous high-frequency information, complementing any sporadic or low-frequency absolute position tracking in an advantageous fashion. It is very well suited for improving position tracking for mixed reality applications, which particularly rely on fast update rates and high-resolution tracking [27], [28], [29]. As a sourceless tracking modality that works both indoors and outdoors, it is a valuable supporting and enabling technology component for the long-term goal of Anywhere Augmentation, making mixed reality possible in unprepared environments without incurring high start-up costs.

3 GroundCam Implementation

The inspiration for the GroundCam is a desktop optical mouse. A camera is pointed directly at the ground from just above waist height, and the video of the ground moving in front of the camera is used to determine how the camera is moving in the plane of the ground. The result is a 2D position tracker. Depending on the environment the GroundCam is being used in, it could be more useful if it is directed at the ceiling, for example, an indoor location with a featureless floor but a textured ceiling. Operation is the same in either case.

The GroundCam takes a few straightforward steps to compute user motion. The pseudocode of this algorithm can be found in Fig. 2. Since features are lost and must be added again each frame, there is no explicit initialization step—instead, the first frame is treated as the case where all the features were lost in the previous frame. This means the GroundCam can recover even in cases of total image loss such as sudden extreme dark or bright conditions, without any user intervention. For the algorithms used in the GroundCam, standard implementations from the OpenCV image-processing library [30] are used, unless stated otherwise.

```
  loop:
1   get frame
2   undistort frame
3   if( num_features < max_features )
4     find new features
5   find optical flow
6   find inliers
7   get orientation
8   compute motion
9   report position
```

Fig. 2. Pseudocode for the GroundCam algorithm.

3.1.1 Undistortion

(line 2 in Fig. 2*)* Offline intrinsic camera calibration is done using Zhang's procedure [31]. The distortion coefficients from this process are used to correct the resulting artifacts in the video frames by creating a corresponding undistortion image warp that is applied to each frame—this allows us to use image distances as direct measurements of distances in the scene. However, for cameras with a narrow field of view, the distortion effect is small enough that it does not produce a significant effect, and undistortion is unnecessary, saving CPU cycles—for example, we do not undistort the video for the camera in Fig. 1, which has a field of view of 12.2 degrees.

3.1.2 Feature Detection

(line 4 in Fig. 2*)* In our system, features are small regions of image texture. Good features for tracking are selected from the video frames by Shi and Tomasi's algorithm [32], which finds a set of all features of a certain quality and then greedily selects features from the set that are not within a minimum distance of the already selected features. After the initial set of features are found, new features are introduced with the same technique as features are lost.

3.1.3 Feature Tracking

(line 5 in Fig. 2*)* Features are tracked frame to frame using the image pyramid-based optical flow algorithm by Lucas and Kanade [33]. A hierarchy of images at different resolutions are used to efficiently match texture features from one frame with the most similar region in another frame. If the similarity between these two regions is below a threshold, the feature is considered lost and is removed from the set. This can happen when a feature goes outside the field of view or when changes in illumination or occlusion occur. Each feature is tracked independently of one another, so their motion may not be (and in most cases is not) uniform. This is a strength of the technique, as distractors can be accounted for so long as, overall, they represent a minority of the viewable scene.

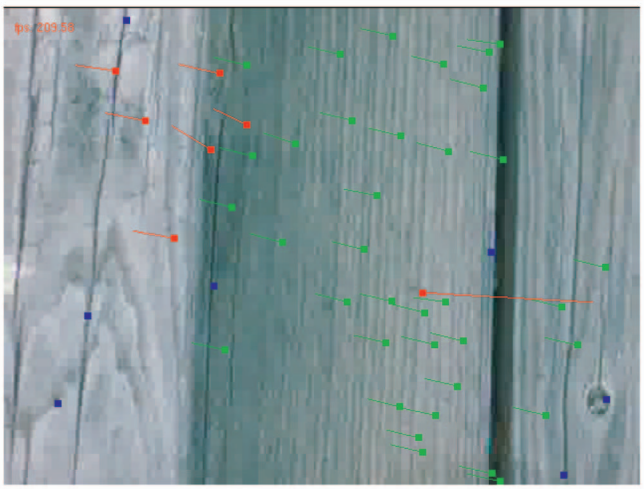

Fig. 3. Features tracked in a camera frame from walking on wood: Green features represent the RANSAC consensus set, red features are outliers, and blue feature points were newly introduced this frame.

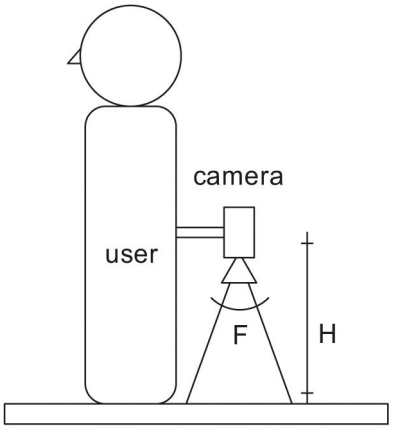

Fig. 4. Model of the camera setup. F is the camera's horizontal field of view, and H is the height of the CCD from the ground. It is assumed that the camera is oriented perpendicular to the ground.

3.1.4 Coherent Motion Estimation

(line 6 in Fig. 2*)* Coherent motion must be extracted from the set of features successfully found in consecutive frames, discarding the influence of outliers. We implemented the RANSAC algorithm [34] to accomplish this task. Only one sample is necessary to estimate the image's 2D translation. Other samples are tested against this estimate by separately thresholding the differences in magnitude and orientation. Once the final set of inliers is found, the image motion estimate is computed by taking the average of all the good samples. In the event that a consensus is not reached, a fall-back estimate is computed as the average of all the samples.

Fig. 3 shows debug information for our feature tracking, overlaid on a camera frame obtained when walking on a wood patio. Features are shown as dots with attached line segments visualizing the estimated motion since the last frame. The green features represent the consensus set, red features represent outliers, and blue feature points are newly added features to make up for lost features and features that left the view frustum.

The computation to get world motion in real units from the image motion in pixels is straightforward. The camera (Fig. 4) is assumed to be perpendicular to the ground at some uniform height (measured offline). For a known height in meters H, camera horizontal field of view F, and camera width in pixels P, the relationship between image motion Δx in pixels and camera motion T in mm is given as

$$T = \frac{2H}{P}\tan\left(\frac{F}{2}\right)\Delta x. \quad (1)$$

A 640 × 480 image from our camera with a field of view of 12.2 degrees, mounted at 1.1 m (just above waist height), yields a factor of 0.37 mm per pixel.

Our implicit assumption that the conversion between image distance and physical distance can be represented by a single scale factor is not actually correct. For different regions of the image, the distance from the camera to the ground varies, even assuming a flat ground and perfectly orthogonal viewing direction. The scale factor we computed is therefore not correct outside of the center of the field of view. Based on (1), we find that the relationship between the actual camera motion T, camera field of view Θ, and camera motion measured near the edges of the image T' is

$$T' = T\cos\left(\frac{\Theta}{2}\right). \quad (2)$$

For our camera with a 12.2 degree field of view, (2) shows that a 0.5 percent error is introduced between computations at the center and the perimeter of the image. This is small enough that we can safely ignore it for our purposes.

3.1.5 World Coordinate Transformation

(lines 7-8 in Fig. 2*)* Our motion estimate is computed in the camera's frame of reference. In order to convert it to the world's coordinate system, we need to know the absolute orientation of the camera. An InterSense InertiaCube2 orientation tracker is used to obtain this information. A quick offline calibration is done to orient the InertiaCube2's output by obtaining angles for north, east, south, and west. During operation, the detected angle is linearly interpolated between these computed values to get the world stabilized camera orientation. The motion vector is then transformed by this orientation to yield the final world stabilized motion estimate.

4 Analysis and Discussion

Numerous experiments using the GroundCam have yielded the following insights into its setup and operation in real-world conditions.

4.1 Performance

The output of the GroundCam is essentially a linear velocity measurement—some distance traveled over a small unit time. Therefore, integration is necessary to use it as a position tracker. However, it compares favorably to the primary alternative, a linear accelerometer, as the acceleration data requires double integration to yield position and so accumulates error much faster. Single integration means our drift

TABLE 2
Average Times (in ms) of the Various Stages of the GroundCam, on a Desktop 2.1-GHz Athlon

line	stage	ms
1	video decoding	9.9
2	undistortion	23.8
	preprocessing total	33.8
4	find new points	9.2
5	optical flow	21.4
	tracking total	35.1
6	RANSAC	0.1
	total	69.0

The preprocessing and tracking are broken up into their component stages, and timings are presented for each stage, as well as the frame total. The final total is the start to finish for each frame of the test application. Line numbers refer to pseudocode in Fig. 2.

over time is drastically reduced. Also similar is the use of a pedometer to track walking motion, which uses a known stride length and counts steps to arrive at a position estimate. However, pedometers are limited to the resolution of a stride and can drift significantly when the user walks with steps of unusual stride (for example, due to terrain considerations like stairs, or from repeated small steps when carefully adjusting ones position).

The limiting factors in the GroundCam's feature tracking are the camera's image quality and frame rate and the size of the visible ground region. Good lighting and good optics improve the performance of the optical flow algorithm significantly—optics improve image quality, and bright light lowers the necessary exposure time, reducing image noise and motion blur (daylight or bright office illumination is generally sufficient). For our setup, we use a Unibrain Fire-i 400 camera with a 12.2 degree field of view lens mounted at 1.1 m, which yields a ground section of 0.24 m by 0.18 m. For half the features to still be visible, the ground can move at most half of this region, 0.09 m along the y-axis, in the time of a single frame. At 10 fps, that is, equivalent to a speed of 0.88 m/s, at 15 fps, 1.18 m/s, and at 20 fps, 1.76 m/s. Since forward motion is most common, mounting the camera rotated at 90 degrees (portrait versus landscape) gives 0.24 m of visible ground along the walking dimension and results in a trackable speed of 1.32 m/s at 10 fps, 1.76 m/s at 15 fps, or 2.35 m/s at 20 fps. Average walking speed is 3 mph or 1.34 m/s, and we consistently get between 15 fps and 20 fps; so, this is sufficient for basic walking behavior. Fast walking or running cause sufficient jitter of the camera's orientation, resulting in significant motion blur and noisy apparent motion, such that they cannot be accurately tracked in any case.

Detailed performance timings are in Table 2. In general, we experience around 15 frames per second in the testing application, which, for testing convenience, runs off of a prerecorded MPEG encoded video and accompanying metadata file. For narrow field of view cameras, the cost of undistortion can be eliminated by not carrying out that step, which saves an additional 23 ms per frame. The performance can be further improved by distributing the load of finding new feature points across many frames. Since features are not likely to be lost in a single frame, they can be searched for periodically rather than every frame. How aggressively this can be done depends on how frequently features are lost and need to be reacquired, which is a function of the field of view of the camera and the speed of motion. Faster camera motion means the features will be in the field of view for fewer frames and so must be reacquired more often. If the video includes significant amounts of distractions, more features will be necessary consistently for better robustness to noise. These considerations are important on a per-application basis.

4.2 Feature Selection

Our choice of 50 tracked features bears justification. A manual comparison was done of the GroundCam's coherent motion estimate output for different target numbers of tracked features, from 25 to 200 (since some number of features are lost each frame, the actual set of features present in two consecutive frames is less than the target). At 25, there were few enough points that a coherent estimate often was not possible, or else, it was very likely to get distracted by random noise over low texture terrain. At 100 and above, the probability of achieving a coherent estimate was very high, but it was necessary to increase the number of inliers required for RANSAC to succeed; so, the overall gain in detection was small. However, the additional CPU drain in tracking and replenishing lost features was significant. Having 50 features is a compromise between CPU cost and likelihood of detecting coherent motion. It may be possible to fine-tune this parameter for particular known types of terrain, but we preferred a single static value.

4.3 Orientation Estimation

The need for an orientation tracker is not necessarily clear in light of research such as Davison's single camera SLAM [8]. In his work, the camera's 6DOF pose is completely determinable from the video stream. However, the dependable high contrast of the texture in his work makes the tracking much more reliable than for the GroundCam. Over high-contrast terrain such as well-lit grass or gravel, there may be enough information to extract the camera's orientation as well, but on terrain such as concrete, asphalt, or hard dirt, there is enough optical flow noise that it is difficult to reliably extract the translation motion estimate. Techniques to improve the quality of feature tracking for these types of terrain could make the individual feature motion information reliable enough for full 6DOF pose estimation.

4.4 Tracking Distractions

There are a number of possible distractions that can reduce the accuracy of the tracking result. The user's lower legs and feet may appear in the camera's field of view, which can create a strong enough optical flow to influence the motion estimate (see Fig. 5a). It would be possible to create a color model of the lower legs and feet, which is likely different from the ground terrain, and use it to mask them out of the ground image before tracking features on it. Alternately, proper mounting of the camera (for example, on the back of a backpack containing the wearable computer) with a narrow field of view alleviates this problem by keeping the feet out of the video.

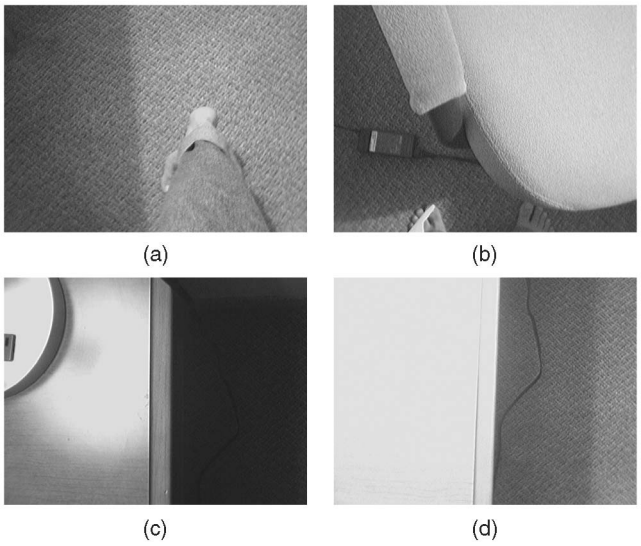

Fig. 5. Example sources of error in tracking. (a) User's feet and legs entering the image. (b) Obstacles not in the ground plane. (c) Under exposure when moving from light to dark. (d) Over exposure when moving from dark to light.

Motion of the user's shadow can have a similar effect, if the leg shadows are moving across the camera's view, by creating many strong features on the shadow boundary that move separately from the ground. Mounting the camera on the front or back diminishes the effect, as the forward-backward motion of legs creates shadows with much less motion in those cases. It would also be possible to use image processing to remove large-scale illumination changes while keeping small local texture at additional CPU cost.

Changing illumination when moving from a well-lit area into a poorly lit one, such as crossing into the shadow of a building can create temporary confusion as the camera's exposure setting automatically adjusts, depending on the contrast and sharpness of the shadow (see Figs. 5c and 5d). This is only a serious problem when the contrast creates under or over saturation of regions of the image, which then do not have trackable texture. As soon as the camera's exposure adjustment compensates (up to one second), tracking resumes as normal.

Finally, changes in the height of the ground plane, either from structures like stairs or random debris on the ground (for example, rocks), may introduce error in the conversion between motion in pixels to meters (see Fig. 5b). Potentially, the use of SIFT features would allow determination of changing heights such as when going up or down stairs, which could then be used to improve the coherent motion estimate. For objects or debris of static height above the ground plane, more reliable feature tracking would be necessary to identify coherent patches of motion of different velocity than the majority of the image and then extract a height estimate.

However, all of these possible distractions combined do not, in general, significantly impact the quality of the tracking result. They are all short temporary effects that introduce small amounts of random noise. The result is that they cause the integrated position to drift slightly faster than it would under ideal conditions, but this has not shown to be a problem.

4.5 Camera Parameters

The camera's height, angle, and field of view, are all important considerations. Since the GroundCam aims to track the position of the user while maintaining its position relative to the ground, it must be mounted on the hips or torso. It should not restrain the user from moving their arms or doing their work; so, the back is best. For wearable systems that include a backpacklike computer, the back of such a device is an ideal location (see Fig. 1).

The height of the camera is important in conjunction with the field of view—the resulting size of the viewable ground region affects the maximum speed that can be tracked, as discussed earlier. It also affects the size of texture features that can be used for tracking. Keeping the viewable ground region small has the advantage of reducing the potential for distractors such as feet to interfere with the tracking. On the other hand, it increases the potential for distractors to take over the majority of the field of view and significantly confound the tracking. These trade-offs must be considered per-application.

We chose to point the camera straight down, perpendicular to the ground. This choice has a few nice properties—first, it makes the motion estimation easy to compute, and the matching of samples to an estimate in the RANSAC algorithm is similarly easy. Second, pointing straight down minimizes the total volume of the viewing frustum of the camera, which means there is less volume for distractors to intrude. Third, this orientation makes tracking easier as features exhibit the smallest change in appearance moving across the field of view. Finally, since the camera is not rigidly held with respect to the ground, its orientation is likely to change slightly during operation—small changes from this orientation will have a smaller impact on the assumptions made than from other orientations. Based on (1), the relationship between the actual camera motion T, camera field of view Θ, angular offset Φ, and measured camera motion T' is

$$T' = T\cos\left(\Phi + \frac{\Theta}{2}\right). \tag{3}$$

For example, if the camera were to become misaligned by 5 degrees, (3) shows the error in the motion estimate would be ≤ 2 percent, which is small enough that it can be ignored.

Vertical motion that occurs as a natural part of the user's walking gait will also introduce error to the motion estimate, as it introduces an offset to the measured height of the camera from the ground. During an average walking motion, the camera will undergo a vertical displacement of approximately 4.5 cm [35], [36]. From (1), we can see that for actual camera motion T, with the camera's height measured as H with vertical displacement d, the measured camera motion T' will be

$$T' = T\frac{H}{H+d}. \tag{4}$$

For our setup with a camera height of 1.1 m and a maximum vertical displacement of 4.5 cm, the error in the motion estimation is ≤ 4 percent. This effect is more consistent than error from the angular motion of the camera, as each step the user takes will exhibit this displacement. The

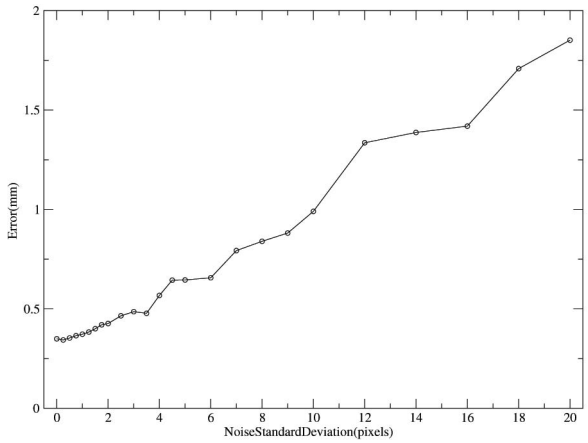

Fig. 6. Average error in the motion computation during translation at 1.0 m/s versus pixel noise added to feature positions.

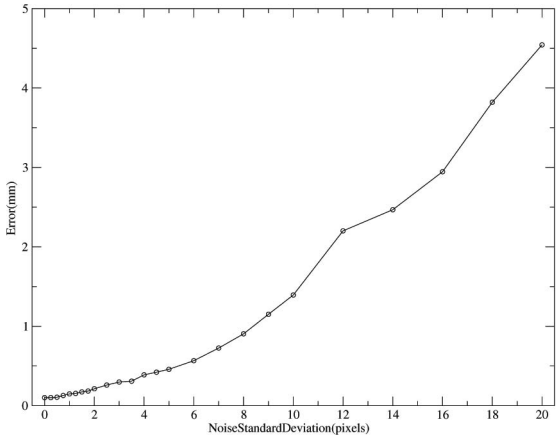

Fig. 7. Average error in the coherent RANSAC motion estimate during translation at 1.0 m/s versus pixel noise added to feature positions, simulating noisy or blurred images.

error can be reduced by adjusting the measured camera height to be the median height rather than the height while the user is standing still. In that case, the maximum vertical displacement would be ± 2.25 cm, and the error to the motion estimate would be ≤ 2 percent, which is once again small enough to be ignored.

4.6 Error Analysis

To more carefully quantify the effects of error on the GroundCam's tracking, we artificially added different types of noise to the stages of the algorithm in a simulated tracking situation and measured the resulting computed motion versus the ground truth.

The tracking simulation was carried out by constructing an artificial ground plane in OpenGL and rendering it into the image buffer normally filled with camera images. The scene consists of a grid of alphabetic characters spaced 3 cm apart for good corner features, on top of a low-frequency black to white gradient to show large-scale motion of the scene. The virtual camera was setup with a 12.2 degrees field of view, 1.1 m above the ground. For a single trial simulation, a sequence of 100 frames was tracked, during which the camera was translated by 5 cm to the right each frame at 20 fps, for a velocity of 1.0 m/s. The average difference between each measurement and the known translation was reported for each trial.

4.6.1 Image Noise

Noisy image data and motion blur both have the effect of adding random perturbations to the detected positions of features, so to test the impact of these effects, random noise is added to the positions of each feature. First, this effect is tested on just the motion computation to see how noise effects its accuracy, and second, the noise is added to the features before RANSAC and motion estimation together to see how the actual GroundCam will perform in light of noise.

Fig. 6 shows how the motion estimation is affected by increasing pixel noise. The nonzero error value for zero input noise is because in this test, RANSAC is not being used for robustness against outliers, and in the simulated environment, the optical flow still sometimes incorrectly detects feature motion. As the amount of noise increases, the resulting error increases roughly linearly, which is reasonable since the motion estimation will simply take the average translation and convert it to real-world units. At 20 pixels of noise, that is, 7.4 mm of error (for 0.37 mm/pixel), whereas the measured error is approximately 1.75 mm. This is because the noise direction is random; so, it tends to cancel itself out over many features. For fewer features, the size of the error will increase until it directly matches the size of the input noise for a single feature. Because these errors are random, they will not cause systematic errors in the tracking but will cause it to drift over time.

Fig. 7 shows the effect of noise on the complete RANSAC and motion estimation computation, as implemented in the GroundCam. Because RANSAC provides robustness to outliers, the error for zero input noise is very small, as the optical flow outliers in the simulated environment can be ignored. As the amount of noise increases, the error increases roughly linearly, but at a much faster rate than without RANSAC. This is because RANSAC finds a subset of the features that are roughly in agreement with one another to use for its estimate, so rather than averaging out the noise added to all the features, it selects a subset that has coherent noise and computes an estimate from those features. The expected amount of noise is an input parameter to the RANSAC implementation that determines when features are roughly in agreement with one another—in situations when larger amounts of image noise are expected, this parameter can be tuned to alleviate the effect and get performance similar to that shown in Fig. 6. The error still increases linearly with more noise, for the same reason as in the test without RANSAC.

4.6.2 Distractions

To test the effect of distractions within the camera's field of view, coherent noise was applied to a subset of the tracked features within the test framework. This simulates the effect of a single large object that moves separately from the ground and should be ignored as outlier data in the motion estimation. Fig. 8 shows the error measured for adding significant coherent noise (standard deviation of 20 pixels, computed once per frame and applied to each feature) to growing portions of the set of features. Since RANSAC will cause a single coherent subset of features to be used for

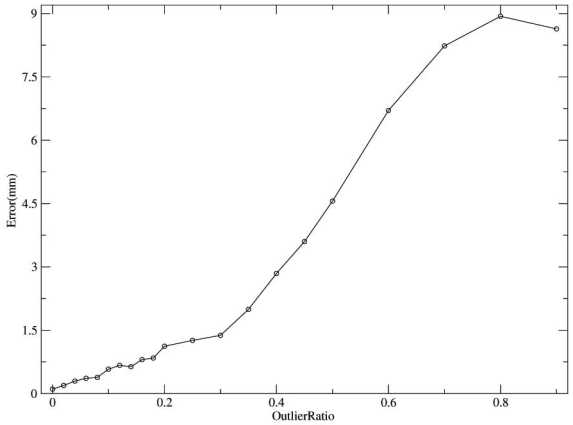

Fig. 8. Average error in the motion estimate during translation at 1.0 m/s versus portions of the features with significant coherent noise added, simulating the effect of large distractions in the image.

motion computation, in this case, there are two possible subsets, one with no noise and one with noise of 20-pixel standard deviation. Therefore, the error is more of a measure of how many out of the 100 frames in each trial sequence did RANSAC select the distraction features. Between 0.4 and 0.6, the error increases dramatically, because this threshold is where RANSAC becomes more likely to select the distraction (due to it taking up half the image) than the ground. Depending on the application, the RANSAC parameters can be tuned to perform optimally for a given expected size and frequency of distractions in the field of view.

5 Wide Field of View

There are advantages to wide field of view cameras that make them desirable instead of narrow field of view cameras. Therefore, it is worthwhile to detail the considerations necessary in order to provide support for their use.

5.1 Undistortion

First of all, for wide field of view cameras, distortion of the image is significant enough that it will lower the quality of the motion estimation, making some sort of correction necessary. The optional image undistortion step described in Section 3 is one option, but applying a per-pixel warp to a 640 × 480 image on the CPU impacts performance significantly (see Table 2). An alternative is to perform the optical flow computation on the distorted image and then to undistort each feature point's position individually. This can be done much faster and for 50 features takes a few tenths of a millisecond, removing the cost of image undistortion.

5.2 Motion Estimation

The error of the motion estimation computation becomes significant for a wide field of view camera as well. Equation (2) shows that while a 12.2-degree field of view camera has only 0.5 percent error, a 50-degree field of view camera has 10.3 percent, and a 90 degree has 41.4 percent. The source of the error is that the distance between the camera and the ground is no longer accurately approximated by the height H for pixels that are far away from the camera's center (that is, a large number of degrees away from the optical center of the camera). Since the

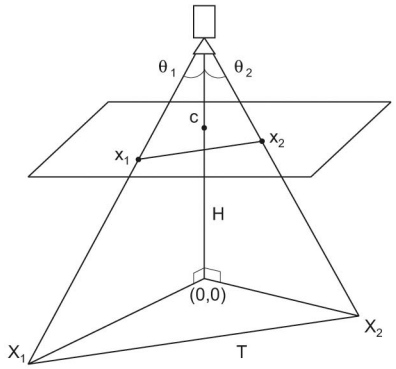

Fig. 9. An illustration of the steps to compute the ground motion T between two image features x_1 and x_2 for a wide field of view camera.

scale factor varies significantly across the image, a single conversion cannot be used as in the narrow field of view case. Instead, the correct solution is to convert each feature point's image motion to the corresponding translation on the ground plane and then find a consensus among those measurements. The math to calculate this is slightly more complex. For a pixel x and a camera center point c, both in pixel coordinates, the angle between rays cast through x and c is

$$\Theta(x) = \cos^{-1}\left(1 + \|x - c\|^2 p^2\right)^{-\frac{1}{2}}, \tag{5}$$

where p is the size of a pixel for an image plane 1 unit in front of the camera, with x-axis field of view f and w pixels wide, assuming square pixels

$$p = \frac{2}{w}\tan\left(\frac{f}{2}\right). \tag{6}$$

For a feature that moves from position x_1 to x_2 in the image, this is now enough information to find the ground translation T. See Fig. 9 for an illustration of these equations. Let the point on the ground directly below the camera's center point c be the ground coordinate origin, (0, 0). Let X_i be the ground coordinate of the image point x_i projected onto the ground. To compute X_i, we combine the magnitude m_i and direction d_i, computed separately

$$X_i = d_i m_i, \tag{7}$$

$$d_i = \frac{x_i - c}{\|x_i - c\|}, \tag{8}$$

$$m_i = H\tan(\Theta(x_i)). \tag{9}$$

Then, with X_1 and X_2, computing T is straightforward:

$$T = X_2 - X_1. \tag{10}$$

By computing T for each tracked feature point, RANSAC can then be run to find a consensus among the ground translation vectors, which is the inverse of the camera motion.

5.3 Features and Motion

The larger field of view also means a larger region of the ground is visible, and thus, a larger maximum velocity can theoretically be tracked. For a 50-degree field of view

Fig. 10. Narrow (10 degrees) and wide (40 degrees) field of view cameras in the same scene.

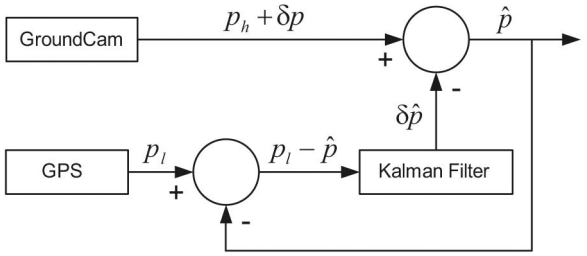

Fig. 11. System diagram of the complementary Kalman filter. A Kalman filter is used to update an error between the current GroundCam estimate and the GPS absolute position. While the Kalman filter is updated infrequently (1 Hz), a new position estimate is generated for each GroundCam update (30 Hz).

camera mounted at 1.1 m, the visible region of the ground measures 1.03 m $\times$ 0.77 m, and the maximum trackable speed is 5.1 m/s at 10 fps, 7.7 m/s at 15 fps, and 10.2 m/s at 20 fps. For 90 degrees, the ground region is 2.20 m $\times$ 1.65 m, and the maximum trackable speed is 11.0 m/s at 10 fps, 16.5 m/s at 15 fps, and 22.0 m/s at 20 fps. The same limitations regarding camera jitter and motion blur apply similarly in the wide field of view case.

Because the camera resolution does not increase with field of view, the larger the visible ground region, the larger features must be to be accurately tracked (see Fig. 10). Terrains such as carpet, asphalt, and concrete will not be trackable for wide field of view cameras, as their texture features are too fine to be tracked at the resulting low angular resolution of the video. For these environments, tracking will rely on the presence of debris on the ground (for example, cables, larger rocks, plants, breaks in the ground, etc.) to provide adequate trackable features. Terrain with larger scale features such as grass, gravel, and wood may still work for larger fields of view.

Finally, the larger field of view means that distractions such as the user's legs will definitely be in the image. To deal with the legs, as was stated earlier, segmentation could be used or that region of the image could simply be masked out. Other distractions are still likely to end up in the image as well such as other legs or objects on the floor. More careful tuning of RANSAC parameters will provide better performance in light of these distractions, but fundamentally, there still needs to be sufficient visible ground for tracking, just as in the narrow field of view case. The difference is that in highly cluttered or dynamic environments, a narrow field of view camera is more likely to be completely distracted (its entire field of view filled with a distractor), which will cause it to measure a coherent but incorrect translation, while a wide field of view camera is more likely to track many, differently moving distractors, which will cause RANSAC to fail to find a consensus. In these cases, hybridization with another sensor such as a linear accelerometer may help by providing a nonvision estimate of the motion to guide selection of ground features for measurement.

6 Hybrid Tracking

The GroundCam by itself is not a sufficient wide area tracking solution because it tends to drift over prolonged operation. Instead, it is most appropriately used in concert with a wide area tracker like a GPS receiver. This loose coupling is achieved with a complementary Kalman filter.

6.1 Complementary Kalman Filter

Our complementary Kalman filter design is inspired by Foxlin's work on orientation tracker filtering [23]. The underlying concept is to filter the error signal between two sensors, rather than filtering the actual position estimate (see Fig. 11).

The signal from the GroundCam is high frequency (30 Hz), high resolution (1 mm), and includes small random errors (10 mm) and large systematic errors (drift is unbounded over time). There are two main sources of error—random errors in the motion estimates per update, and random underestimation of motion when RANSAC fails to find a coherent estimate. These errors accumulate over time due to integration of the GroundCam signal. The signal from a standard GPS receiver is low frequency (1 Hz), medium resolution (10 cm), and includes medium random and systematic errors (5 m). The main source of the error is due to changing atmospheric conditions, which delay the signals from GPS satellites differently, creating apparent differences in position. Generally, this error is randomly distributed around the true position, but prevailing weather conditions such as cloud cover or view obstructions such as buildings can create systematic errors in the signal over extended periods of time.

Ideally, the filtered output will be available at the high frequency and high resolution, which the complementary Kalman filter achieves with minimal processor load. We can model the error between the two signals as a smoothly varying random process with a Kalman filter and then use the filtered error signal to correct the GroundCam signal on the fly.

Let p_h be the high-frequency signal from the GroundCam and p_l be the low frequency signal from a GPS receiver. p is the ground truth position, $\hat{p}$ is the estimated position, $\delta p = p - p_h$, and $\delta\hat{p}$ is the estimated error signal. Since our filter operates on 2D data, p is actually the vector $[x, y]^T$. Within the Kalman filter, there are six process dimensions and two measurement dimensions. Filter variable names are standard as used in [37]:

$$\mathbf{x} = \begin{bmatrix} \delta p \\ \delta\dot{p} \\ \delta\ddot{p} \end{bmatrix}, \tag{11}$$

$$\mathbf{z} = [\, \delta p \,], \tag{12}$$

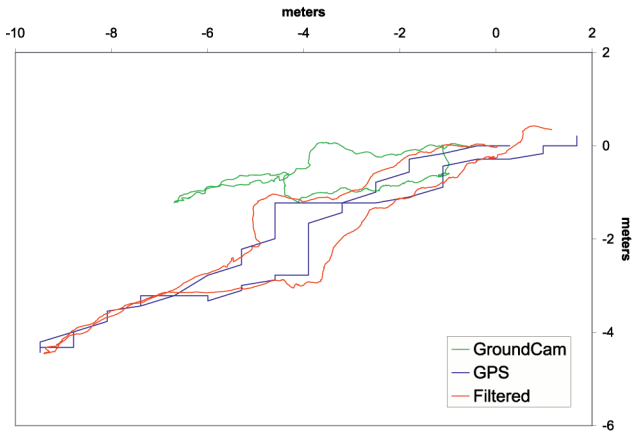

Fig. 12. A trial run of the GroundCam coupled with GPS. The run was 90 seconds in duration, over wood, gravel, and concrete terrain, and included avoiding obstacles and going up and down stairs. The slip rate was 30 percent.

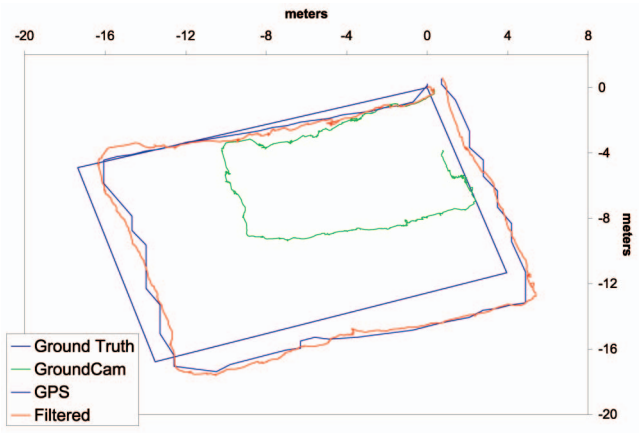

Fig. 13. A trial run of the GroundCam coupled with GPS, with hand-labeled ground truth. The run was 81 seconds long, over concrete and asphalt, along a rectangle 18 m long by 12 m wide. The slip rate was 80 percent and RMS errors for the GroundCam, GPS, and filtered signals are 5.5 m, 1.9 m, and 1.9 m, respectively.

$$A = \begin{bmatrix} 1 & \Delta t & \frac{1}{2}\Delta t^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix}, \tag{13}$$

$$H = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}. \tag{14}$$

B and $\mathbf{u}$ are both not used, and thus zero. Q and R are empirically determined depending on the particular sensor being coupled with the GroundCam, and P is initially set so measurements are preferred at start-up.

The result of a complementary filter setup such as this is that for each new high-frequency update, only a prediction and then subtraction is necessary, making the processor load very low for the frequent step. The expensive correction step is computed once per low-frequency update.

6.2 Potential for Coupling

There are a number of possible wide area trackers that could be integrated with the GroundCam in this manner, depending on the needs of the particular system. GPS is a straightforward choice for outdoor applications, as its signal is commonly available, and sensors are cheap. Applications without a clear view of the sky, however, such as dense urban environments or indoors, must consider alternative solutions. In these cases, a cheap and easily deployable beacon-based system, for example, on RF, ultrasound, or infrared basis [11], [12], [38], [39], may be more appropriate. Such systems provide position information in the sense that they identify which discrete region the user currently occupies. This information would be sufficient for applications such as audio annotations or situated content, but for visual overlays or immersive virtual content, coupling with a more accurate tracker like the GroundCam is necessary.

The coupling of the GroundCam with another sensor may be done differently as well, for different needs. For instance, a common problem with GPS signals is that while the user is standing still, error in the GPS signal will make it appear as though the user is moving slowly. This drift can make operations that require stationary actions very difficult. The GroundCam, on the other hand, is very good at determining when the user is standing still and could be used as a binary walking/standing behavior classification to selectively ignore GPS updates.

7 Results

Fig. 12 shows a typical run using the GroundCam and the GPS hybrid tracking system for approximately 90 seconds. The path includes avoiding obstacles and going up and down steps, with wood, gravel, and concrete terrain. As expected, the GroundCam exhibits some drift, partially from random errors in the motion estimate and also from updates where a coherent estimate cannot be generated. These errors cause different effects in the GroundCam path—random errors make the path less smooth, while missing coherent estimates create a shortening effect. However, the coupling with the GPS signal eliminates the effect of the GroundCam drift. Of particular importance is the much smoother quality of the filtered signal than the raw GPS signal, which makes the hybrid tracker very appropriate for mixed reality applications.

For comparison purposes, the run in Fig. 13 includes a hand-labeled ground truth—a rectangular path of approximately 18 m $\times$ 12 m over 81 seconds on a residential street. The terrain is concrete and asphalt, which have lower contrast textures and are more prone to noise in the error estimates. For this particular trial, our GPS receiver experienced very little random noise but did exhibit a significant drift overall due to our GPS unit not receiving a WAAS signal at our location. While our filtered path stays close to the GPS signal, we cannot correct for systematic errors in the GPS position, which are propagated into our tracking result. In most US locations, the presence of a WAAS signal will improve the quality of the GPS data and subsequently improve the filtered data as well.

7.1 Slip Compensation

The problem of RANSAC not reaching a consensus is analogous to the problem of slipping wheels in the odometry of wheeled vehicles and results in estimated paths that are much shorter than the ground truth. Certain types of terrain are more prone to this sort of error (see Fig. 14). Low-contrast terrains like concrete were much more prone to slipping than high-contrast terrains such as grass.

We made a simple attempt to compensate for some of this error, which we call *slip compensation*. The error is proportional to the rate at which RANSAC does not

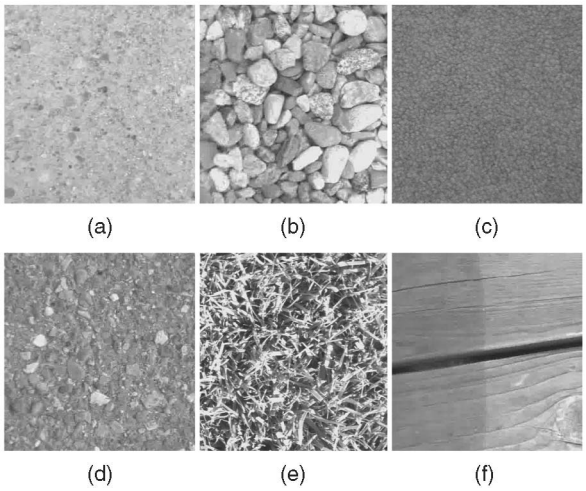

Fig. 14. Different types of terrain with example slip rates. (a) Concrete (80 percent). (b) Gravel (32 percent). (c) Carpet (48 percent). (d) Asphalt (65 percent). (e) Grass (20 percent). (f) Wood (24 percent). Slip rates depend on speed, jitter, lighting, and debris, in addition to texture contrast.

produce a coherent estimate, or the *slip rate*. Based on the slip rate over a short window of time, a successful coherent estimate is scaled to compensate for the missed estimates (for example, if the slip rate is $s = 0.8$, then a coherent estimate is scaled by $(1 - s)^{-1} = 5.0$). Fig. 15 clearly shows that slip compensation helps achieve the appropriate scale of the GroundCam signal.

7.2 Beacon-Based Wide Area Sensors

To demonstrate the usefulness of the GroundCam in concert with wide area sensors other than GPS, we simulated a discrete beacon-based wide area sensor signal (similar in concept to the Cricket [12] and Locust Swarm [11] projects). We used ground truth to trigger a periodic signal that identified which discrete region the user currently occupied on a rectilinear grid of 6-m cells (roughly room size). This coarse wide area signal was used in place of the GPS signal in the complementary Kalman filter.

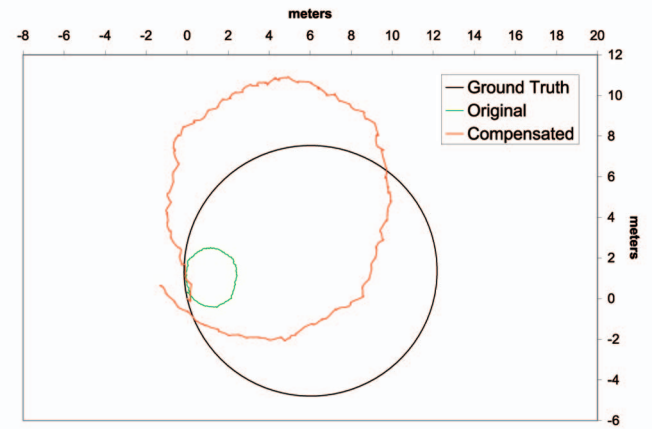

Fig. 15. A trial run of the GroundCam with and without slip compensation, with hand-labeled ground truth. The trial was 72 seconds in duration over asphalt and had a slip rate of 63 percent. Originally, the RMS error was 7.0 m; with slip compensation, the RMS error is 4.8 m.

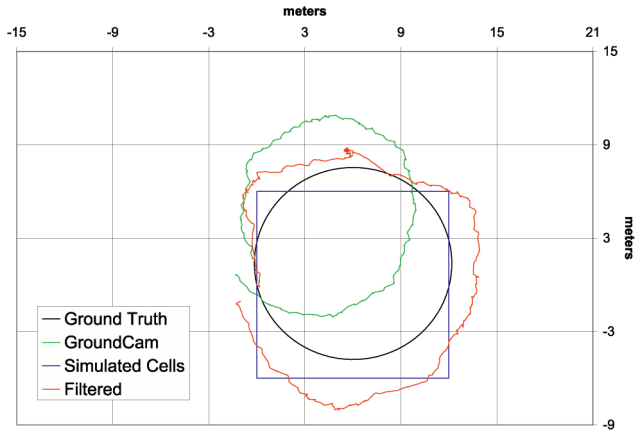

Fig. 16. A trial run of the GroundCam (with slip compensation) with a simulated beacon-based wide area sensor in place of the GPS signal. The RMS errors of the GroundCam, beacon signal, and filtered signal are 4.6 m, 2.3 m, and 1.9 m, respectively.

Fig. 16 shows that the beacon-based signal provides a measure of drift correction that improves the GroundCam's raw result. For a longer path, the GroundCam drift would result in significant divergence from ground truth, while the beacon-based signal would make sure the filtered output stays within certain bounds of the true position.

7.3 Application Performance

To evaluate the utility of the GroundCam's performance, the tracking requirements for mixed reality applications need to be examined. Azuma posited that an outdoor system using GPS with a 3-m accuracy, viewing a scene at a distance of 100 m would have a registration error of 1.7 degrees [40]. The equation that provides this relationship is

$$\delta = \tan^{-1}\left(\frac{\varepsilon}{d}\right), \tag{15}$$

where d is the distance to the scene, ϵ is the position error, and δ is the resulting registration error. For a maximum 1-degree error, the relationship between d and ϵ becomes

$$\varepsilon = d \tan 1.0 = 0.0174d. \tag{16}$$

Therefore, 1.7 m is the allowable maximum position error for a scene 100 m away or 0.17 m for a 10 m scene. As the results in Fig. 15 show, after 30 seconds (half the run), the GroundCam's error was roughly 3 m, which make it insufficient as a lone tracking modality. Even if the error were an order-of-magnitude lower, over long tracking periods, it would still accumulate to large values because of the dead-reckoning nature of the GroundCam. However, when combined with a wide-area sensor such as GPS, the periodic absolute position updates limit the amount of error accumulated by the GroundCam to 1 second worth, or roughly 0.1 m, which meets our maximum registration error criteria.

8 Conclusion and Future Work

We have presented the GroundCam tracking modality, a vision-based local tracker with high resolution, good short-term accuracy, and an update rate appropriate for interactive graphics applications. We have also demonstrated the

feasibility of a hybrid tracker, coupling the GroundCam with a GPS receiver, and a discrete beacon-based wide area sensor. In our trials, the GroundCam compares favorably to other similar tracking modalities. A recent focus in our mobile mixed reality work has been the joint use of local sensors, ubiquitously available GIS data sources such as aerial photographs of a user's environment and fast and direct human input, all in pursuit of Anywhere Augmentation [1]. Users are enabled to switch back and forth between an aerial view of their current location, and a first-person camera view, and to annotate either with simple interaction. The GroundCam significantly enhances the user experience in our application scenarios, as it gives us a means to determine robustly when a user interacts with the system from a static position as opposed to use while walking. When the data from the GroundCam indicates a static position, we can correct for GPS drift. In addition, our hybrid tracker's real-time update rate for reporting the user's position on the aerial photograph is a great improvement over the 1 Hz sample rate of our run-of-the-mill GPS unit. Toward the goal of Anywhere Augmentation, the GroundCam is cheap, readily available, and requires almost no time to setup in a new environment for high-quality tetherless tracking in mixed reality applications.

The most promising avenue for future work is to extend the GroundCam to use a wide field of view camera to measure orientation, as well as translation, removing the need for the separate orientation unit. The motion estimation algorithm must be modified to compute 2D image translation and rotation simultaneously. Because the orientation would be prone to drift just as the GroundCam position measurement is, periodic correction can be generated by the recent velocity of the GPS unit under the assumption that users generally walk forward or backward, rather than left or right. This would remove the most expensive component from the GroundCam tracking solution, making it readily available to even casual mixed reality users.

Acknowledgments

This research was in part funded by a grant from US National Science Foundation (NSF) IGERT in Interactive Digital Multimedia DGE-0221713 and a research contract with the Korea Institute of Science and Technology (KIST) through the Tangible Space Initiative Project.

References

[1] T. Höllerer, J. Wither, and S. DiVerdi, ""Anywhere Augmentation": Towards Mobile Augmented Reality in UnpreparedEnvironments," *Location Based Services and TeleCartography*, Lecture Notes in Geoinformation and Cartography, G. Gartner, M. Peterson, and W. Cartwright, eds., Springer, pp. 393-416, Feb. 2007.

[2] Polhemus, *FastTrack*, http://www.polhemus.com/, Sept. 2006.

[3] C. Randell and H. Muller, "Low Cost Indoor Positioning System," *Proc. Ubiquitous Computing*, pp. 42-48, 2001.

[4] B. Barshan and H. Durrant-Whyte, "Inertial Navigation Systems for Mobile Robots," *Trans. Robotics and Automation*, vol. 11, no. 3, pp. 328-342, 1995.

[5] C. Randell, C. Djiallis, and H. Muller, "Personal Position Measurement Using Dead Reckoning," *Proc. Int'l Symp. Wearable Computers*, pp. 166-173, 2003.

[6] WorldViz, *Precision Position Tracker*, http://www.worldviz.com/, Sept. 2006.

[7] I. Poupyrev, D. Tan, M. Billinghurst, H. Kato, H. Regenbrecht, and N. Tetsutani, "Developing a Generic Augmented-Reality Interface," *Computer*, vol. 35, no. 3, pp. 44-50, Mar. 2002.

[8] A. Davison, "Real-Time Simultaneous Localisation and Mapping with a Single Camera," *Proc. IEEE Int'l Conf. Computer Vision (ICCV'03)*, Oct. 2003.

[9] E. Foxlin and L. Naimark, "VIS-Tracker: A Wearable Vision-Inertial Self-Tracker," *Proc. Virtual Reality*, pp. 199-206, 2003.

[10] I. Getting, "The Global Positioning System," *IEEE Spectrum*, vol. 30, no. 12, pp. 36-47, Dec. 1993.

[11] T. Starner, D. Kirsch, and S. Assefa, "The Locust Swarm: An Environmentally- Powered, Networkless Location and Messaging System," *Proc. Int'l Symp. Wearable Computers*, pp. 169-170, 1997.

[12] N. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket Location-Support System," *Proc. Int'l Conf. Mobile Computing and Networking*, pp. 32-43, 2000.

[13] P. Bahl and V. Padmanabhan, *RADAR: An In-Building RF-Based User Location and Tracking System*, vol. 2, pp. 775-784, 2000.

[14] J. Campbell, R. Sukthankar, and I. Nourbakhsh, "Techniques for Evaluating Optical Flow for Visual Odometry in Extreme Terrain," *Proc. Int'l Conf. Intelligent Robots and Systems*, vol. 4, pp. 3704-3711, 2004.

[15] S. Se, D. Lowe, and J. Little, "Vision-Based Mobile Robot Localization and Mapping Using Scale-Invariant Features," *Proc. Int'l Conf. Robotics and Automation*, pp. 2051-2058, 2001.

[16] S. DiVerdi and T. Höllerer, "GroundCam: A Tracking Modality for Mobile Mixed Reality," *Proc. Int'l IEEE Conf. Virtual Reality*, pp. 75-82, 2007.

[17] A. Haro, K. Mori, T. Capin, and S. Wilkinson, "Mobile Camera-Based User Interaction," *Proc. Int'l Conf. Computer Vision Workshop Human Computer Interaction*, pp. 79-89, 2005.

[18] S. Lee and J. Song, "Mobile Robot Localization Using Optical Flow Sensors," *Int'l J. Control, Automation, and Systems*, vol. 2, no. 4, pp. 485-493, 2004.

[19] D. Nistér, O. Naroditsky, and J. Bergen, "Visual Odometry for Ground Vehicle Applications," *J. Field Robotics*, vol. 23, no. 1, 2006.

[20] L. Fang, P. Antsaklis, L. Montestruque, M. McMickell, M. Lemmon, Y. Sun, H. Fang, I. Koutroulis, M. Haenggi, M. Xie, and X. Xie, "Design of a Wireless Assisted Pedestrian Dead Reckoning System—The NavMote Experience," *Trans. Instrumentation and Measurement*, vol. 54, no. 6, pp. 2342-2358, 2005.

[21] D. Hallaway, T. Höllerer, and S. Feiner, "Bridging the Gaps: Hybrid Tracking for Adaptive Mobile Augmented Reality," *Applied Artificial Intelligence J.*, special issue on AI in mobile systems, vol. 18, no. 6, pp. 477-500, 2004.

[22] S. Lee and K. Mase, "A Personal Indoor Navigation System Using Wearable Sensors," *Proc. Int'l Symp. Mixed Reality*, pp. 147-148, 2001.

[23] E. Foxlin, "Inertial Head-Tracker Sensor Fusion by a Complementary Separate-Bias Kalman Filter," *Proc. Virtual Reality*, pp. 184-194, 1996.

[24] S. You and U. Neumann, "Fusion of Vision and Gyro Tracking for Robust Augmented Reality Registration," *Proc. Virtual Reality*, pp. 71-78, 2001.

[25] B. Jiang, U. Neumann, and S. You, "A Robust Hybrid Tracking System for Outdoor Augmented Reality," *Proc. Virtual Reality*, pp. 3-10, 2004.

[26] Y. Li and J. Wang, *Low-Cost Tightly Coupled GPS/INS Integration Based on a Nonlinear Kalman Filtering Design*, Inst. Navigation Nat'l Technical Meeting, 2006.

[27] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre, "Recent Advances in Augmented Reality," *Computer Graphics and Applications*, vol. 21, no. 6, pp. 34-47, 2001.

[28] G. Welch and E. Foxlin, "Motion Tracking: No Silver Bullet, but a Respectable Arsenal," *Computer Graphics and Applications*, vol. 22, no. 6, pp. 24-38, 2002.

[29] T. Höllerer and S. Feiner, "Mobile Augmented Reality," *Telegeoinformatics: Location-Based Computing and Services*, H. Karimi and A. Hammad, eds. Taylor and Francis Books, 2004.

[30] Intel Corporation, *Open Source Computer Vision Library Reference Manual*, Dec. 2000.

[31] Z. Zhang, "A Flexible New Technique for Camera Calibration," *Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330-1334, 2000.

[32] J. Shi and C. Tomasi, "Good Features to Track," *Proc. Conf. Computer Vision and Pattern Recognition*, 1994.

[33] B. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," *Proc. Int'l Joint Conf. Artificial Intelligence,* pp. 674-679, 1981.
[34] M. Fischler and R. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Comm. ACM,* vol. 24, pp. 381-395, 1981.
[35] R. Boulic, P. Glardon, and D. Thalmann, "From Measurements to Model: The Walk Engine," *Proc. Conf. Optical 3D Measurement Techniques,* 2003.
[36] M. Orendurff, A. Segal, G. Klute, J. Berge, E. Rohr, and N. Kadel, "The Effect of Walking Speed on Center of Mass Displacement," *J. Rehabilitation Research and Development,* vol. 41, no. 6A, pp. 829-834, Nov./Dec. 2004.
[37] G. Welch and G. Bishop, *An Introduction to the Kalman Filter,* Siggraph Course Notes, course 8, 2001.
[38] D. Hallaway, T. Höllerer, and S. Feiner, "Coarse, Inexpensive, Infrared Tracking for Wearable Computing," *Proc. Int'l Symp. Wearable Computers,* pp. 69-78, 2003.
[39] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster, "The Anatomy of a Context-Aware Application," *Proc. Int'l Conf. Mobile Computing and Networking,* pp. 59-68, 1999.
[40] R. Azuma, "The Challenge of Making Augmented Reality Work Outdoors," *Mixed Reality: Merging Real and Virtual Worlds,* Y. Ohta and H. Tamura, eds., Springer, 1999.

Stephen DiVerdi received the PhD degree in computer science from the University of California, Santa Barbara, in 2007. He is a research scientist at Adobe Systems Inc. His work is focused on applying computer graphics and computer vision to create new modes of interaction between humans and computers. He was a member of the "Four Eyes" Laboratory, University of California, Santa Barbara. He is a student member of the IEEE.

Tobias Höllerer received a graduate degree in informatics from the Technical University of Berlin and the MS and PhD degrees in computer science from Columbia University. He is an assistant professor of computer science at the University of California, Santa Barbara, where he leads the "Four Eyes" research group, conducting research in the "Four I"s of Imaging, Interaction, and Innovative Interfaces. He is a member of the IEEE.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.