Sensor Fusion to Detect Scale and Direction of Gravity in Monocular Slam Systems

Seth Tucker and Mohamed El-Sharkawy
IoT Collaboratory, Department of Electrical and Computer Engineering
Purdue School of Engineering and Technology, IUPUI
seth.c.tucker@gmail.com and melshark@purdue.edu

***Abstract*—Monocular simultaneous localization and mapping (SLAM) is an important technique that enables very inexpensive environment mapping and pose estimation in small systems such as smart phones and unmanned aerial vehicles. However, the information generated by monocular SLAM is in an arbitrary and unobservable scale, leading to drift and making it difficult to use with other sources of odometry for control or navigation. To correct this, the odometry needs to be aligned with metric scale odometry from another device, or else scale must be recovered from known features in the environment. Typically known environmental features are not available, and for systems such as cellphones or unmanned aerial vehicles (UAV), which may experience sustained, small scale, irregular motion, an IMU is often the only practical option. Because accelerometers measure acceleration and gravity, an inertial measurement unit (IMU) must filter out gravity and track orientation with complex algorithms in order to provide a linear acceleration measurement that can be used to recover SLAM scale. This paper will explore an alternative method, which detects and removes gravity from the accelerometer measurement by using the unscaled direction of acceleration derived from the SLAM odometry.**

I. Introduction

Monocular simultaneous localization and mapping (SLAM) algorithms are capable of inexpensively generating highly accurate maps of their environment and telemetry for agents within that environment. Smartphones and drones are just two important examples of emerging platforms on which developers and researchers will be able to leverage improvements to monocular SLAM methods. Unfortunately, monocular SLAM is incapable of observing scale. Monocular SLAM can be far more useful if its arbitrary internal scale can be related to objective units of measure, such as meters. Given sufficiently accurate odometry from corroborating sensors, this is easily achieved; a scaling constant can be determined by dividing odometry measurements generated by the monocular SLAM algorithm by corroborating sensor measurements of known metric scale. However, if extremely accurate redundant telemetry is available, than the telemetry obtained from monocular SLAM is unnecessary. The goal, therefore, is to robustly find a scale factor with small, inexpensive sensors in order to take advantage of the relatively inexpensive and compact implementations of monocular SLAM.

Even where monocular SLAM is capable of accurate, self-consistent mapping and pose estimation, accurately orienting the result with earth's gravity vector is challenging, particularly without an known initialization state. In order to do this, it is necessary to use other sensors to align the SLAM map and pose to the real world. In many cases, only noisy, low cost inertial sensors are available for this task. Inertial sensors can not directly measure linear motion or absolute orientation, because the accelerometer measures the superposition of the gravity vector and the vector of linear acceleration. Commonly, this is dealt with by averaging the accelerometer over some timeframe during which the net linear acceleration is assumed to be approximately zero. In this paper, a method will be proposed that couples the gravity vector detection for both the internal sensors and the SLAM coordinate frame the scale estimation, which provides a potentially better estimate of both quantities. This will allow for metric scale navigation and sensor fusion in autonomous aircraft, and improved image stability for smartphone augmented reality apps.

II. Previous Work

There are two basic approaches that are used to determine scale. One approach is to use known features or landmarks in the environment to establish scale. A printed grid of known size can be used to infer scale as well as orientation and relative position with a single camera. This implementation is common in marker based augmented reality systems such as ARToolkit [1]. Marker based augmented reality isn't really SLAM, but markers can be used in conjunction with SLAM to provide scale. Unfortunately, most real world applications do not involve environments that contain known markers to detect scale. As image recognition improves, it will become possible to infer scale from a wider set of objects found in the environment, rather than relying on artificial markers, and some efforts have been made to do this [2]. However, these techniques are still limited to environments which have features of approximately known size.

The other approach is to compare the pose trajectory indicated by other sensors with the pose trajectory indicated by the monocular SLAM system. For example, in a large scale outdoor environment, the path of travel indicated by a GPS system can be aligned with the path indicated by the monocular SLAM system. This approach is effective for systems undergoing large scale motion outdoors, but other

This is the author's manuscript of the article published in final edited form as:
Tucker, S., & El-Sharkawy, M. (2018). Sensor Fusion to Detect Scale and Direction of Gravity in Monocular Slam Systems. IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society, 2447–2452. https://doi.org/10.1109/IECON.2018.8591296

sensors must be used for smaller scale or indoor motion. Usually an inertial measurement unit (IMU) is most convenient, due to its small size and low cost. Typically short positional estimates are taken from the IMU and compared to positional changes indicated by the monocular SLAM system using some data fitting optimization technique [3] [4]. The IMU processes accelerometer and gyroscope data over time to estimate the gravity vector and then infer positional changes. We will propose an approach that uses raw accelerometer data directly in conjunction with monocular SLAM odometery to simultaneously estimate gravity and scale with minimized requirements for a priori information.

III. Notation

Monocular SLAM odometery reports the camera's position with each new camera frame in a fixed coordinate system whose origin is usually referenced to the position and orientation of the first camera frame. An accelerometer, on the other hand, is mounted rigidly to the camera, and it will measure a three dimensional acceleration vector in the body coordinate frame, which is rigidly attached to the camera. To work with these measurements, we will need to rotate one of them into a coordinate frame aligned with the other. If we use a known rigid transformation to express an accelerometer reading in a coordinate frame aligned with the SLAM coordinates, we will say that it is in the SLAM coordinate frame, even though technically it is only in a set of coordinate frames that share a common orthonormal basis.

We will use superscripts to denote reference frames that share a common alignment. To refer to a state variable that is expressed in a coordinate system aligned with the SLAM coordinate system, we will use the superscript 's' to represent 'SLAM'. For example, the position vector is slam aligned coordinates is $\vec{p}^s$. On the other hand, an accelerometer measurement is taken in the body frame, so this measurement can be represented by the vector $\vec{a^b}$, though it is important to remember the transformation between a body frame and a fixed frame changes with each new measurement, so not all "body" frames are aligned through time. At first, we will assume that the physical separation between the camera and the accelerometer is small enough to be ignored, so both the camera and the accelerometer will share a single body coordinate frame.

Finally, a subscript will generally refer to the instrument that the particular value was derived from. For example, $\vec{a_a^b}$ refers to the acceleration measured by the accelerometer in the body frame, whereas $\vec{a_s^s}$ refers to the acceleration according the SLAM odometery in the SLAM coordinate frame.

IV. Digital Filtering

The methods described here require the first and second derivatives of discrete signals. Numerical differentiation is difficult, because differentiation tends to strongly amplify any high frequency noise in the signal. Strong filtering may be needed to remove this noise, which can seriously degrade the bandwidth of the signal, and require significant buffering. The simplest method is to use a finite difference methods such as

$$f'(n) \approx (f(n) - f(n-1))f_s \tag{1}$$

where f_s is the sampling frequency. A second order finite difference can be obtained with at least three samples,

$$f''(n) \approx (f(n+1) - 2f(n) + f(n-1))f_s^2 \tag{2}$$

This approach was insufficient for the signals described in this paper, as they excessively amplified high frequency noise. Fortunately, better options exist. A Savitksy-Golay [5] filter produces adequate results in combination with a short FIR filter. An IIR filter may be even more helpful if real time operation is important.

V. Determining the Linear Acceleration

We will demonstrate a method for determining the linear acceleration component of an accelerometer measurement by comparing its direction to that of the linear acceleration derived from monocular SLAM odometry. The result easily leads to an estimate of the gravity vector and scale in a manner similar to the inertial solution outlined in the previous section. This solution is most robust under significant acceleration, and is well suited to initialization. It is a natural complement to traditional inertial measurement unit (IMU) gravity estimators as it has an inverse symmetry with those method's strengths and weaknesses.

Consider the unit vector of the twice differentiated SLAM odometery position vector. Recall that the superscript refers to the coordinate frame of the vector, and a letter subscript refers to the source of the vector, ie s for SLAM, or a for accelerometer.

$$\vec{a}_s^s(t) = \frac{\partial^2}{\partial t^2}\vec{p}^s(t) \tag{3}$$

$$\vec{u}^s = \frac{\vec{a}_s^s}{\|\vec{a}_s^s\|} \tag{4}$$

Note that in practice, $\vec{p}^s(t)$ is a discrete function rather than continuous, so its second derivative must be determined with a discrete approximation as discussed earlier.

Consider the $\vec{u}^s \times \vec{a}_a^s$ plane in Figure 1, where $\vec{a}_a^s$ the accelerometer's measurement vector transformed in the SLAM frame. Assuming accurate values for $\vec{u}^s$ and $\vec{a}_a^s$, the vector of scaled linear acceleration must lie somewhere along the direction of $\vec{u}^s$. If $\vec{u}^s$ and $\vec{a}_a^s$ are both accurate and consistent, we can assume the following will hold true, where $\vec{w}^s$ is the vector of scaled linear acceleration.

$$\vec{w}^s = \vec{a}_a^s - \vec{g}_n \tag{5}$$

This places that $\vec{g}_n$ somewhere in the $\vec{u}^s \times \vec{a}_a^s$ plane. If we can find $\vec{g}_n$, we can find $\vec{w}^s$, and from there get our scale. Happily, we know $||\vec{g}_n|| \approx 9.81 m/s^2$ at the earth's surface. Intuitively, it should be clear that if we trace a circle of radius $9.81 m/s^2$ from the end of the $\vec{a}_a^s$ vector, it will intersect a line extending from $\vec{u}^s$ at one or two points. These points are candidates for the linear acceleration vector $\vec{w}^s$. Because the scale value

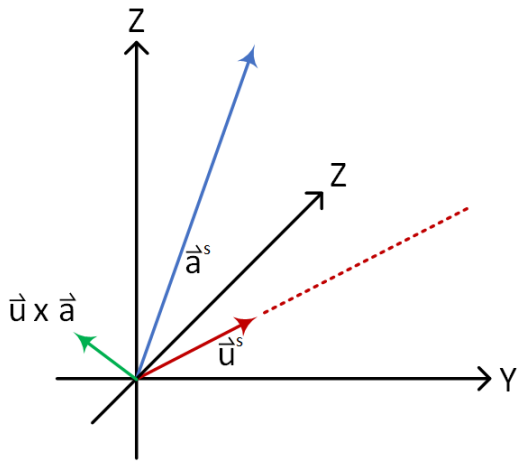


Fig. 1. $\vec{g}_n$ lies somewhere in the $\vec{u}^s \times \vec{a}_a^s$ plane.

must be positive, $\vec{u}^s$, any accelerometer measurement with a magnitude of less $1g$ will give a unique solution for $\vec{w}^s$. However, if $||\vec{a}_a^s|| \geq 1g$, there are potentially two valid candidates for $\vec{w}^s$ and $\vec{g}$.

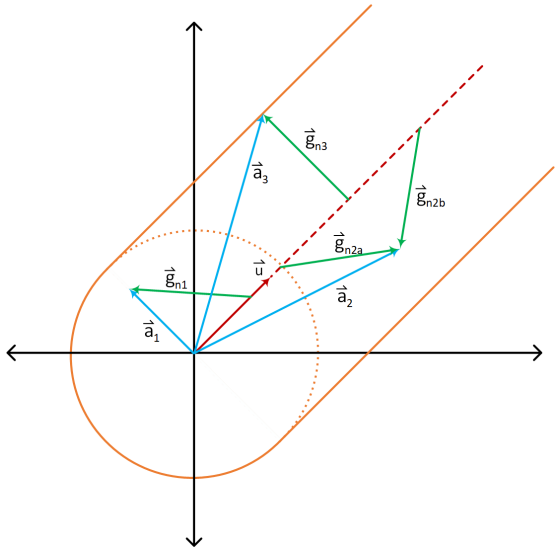


Fig. 2. For a given SLAM acceleration unit vector, $\vec{u}$, accelerometer measurement vectors can potentialy end anywhere within $9.81m/s^2$ of the positive half of the $\vec{u}$ line. Depending on where $\vec{a}_a$ lies in that range, there may be either one or two possible vectors for gravity. If $\vec{a}_a$ ends inside the dash-lined sphere, like $\vec{a}_1$ or on the boundary lines, like $\vec{a}_3$, there is only one solution.

Consider the one dimensional case in Figure 3. We cannot fully determine the gravity vector with only the accelerometer and SLAM odomtery, but it's easy to see that the orientation does not have to be known to any significant degree of accuracy in discriminate between the two possibilities. In this case, we can decide as long as we have some sensor, or a priori guarantee that the system is either inverted or upright. While it may be very difficult to track the gravity vector with great accuracy using traditional methods, it is trivial to make

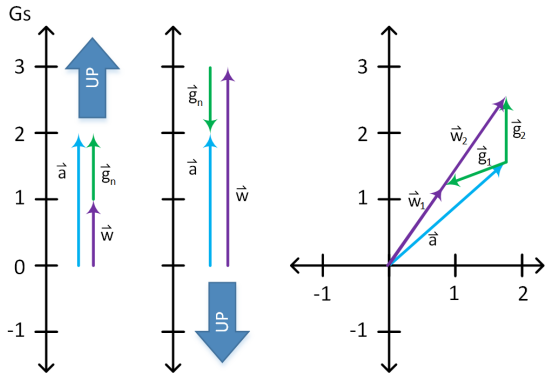
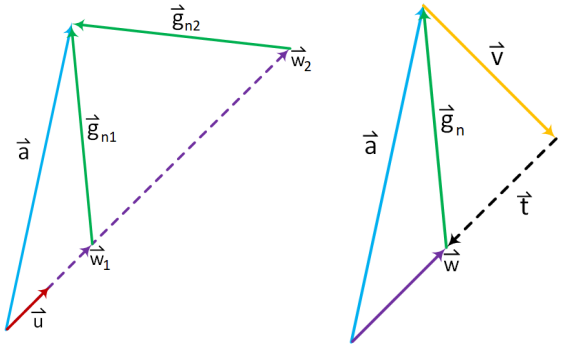


Fig. 3. The left and center diagrams represent the accelerometer reading in two different orientations.

a general distinction between a possible upright or inverted orientation.

This one dimensional perspective extended easily to two and three dimensions, as we can see from the right hand scenario in Figure 3. Because $\vec{a}_a^s$ and $\vec{u}^s$ are no longer aligned, the gravity vector candidates are no longer 180 degrees apart. As the angle between the gravity vectors decreases, it become progressively more difficult to distinguish between the two candidates. Fortunately, the error incurred by picking the wrong vector for $\vec{g}_n$ progressively decreases as the difficulty rises, until at a separation of zero degrees, the two choices are equivalent. This corresponds to $\vec{a}_3$ in Figure 2.

A. *Standard Computation of Linear Acceleration*



(a) Both Solutions (b) Detail of Single Solution

Fig. 4. This illustrates the vectors which are used to solve for $\vec{w}$, the scaled acceleration. (b) illustrates the case where $\vec{t}$ is subtracted for the vector projection.

We can now describe a general method to calculate both candidates for metric linear acceleration, $\vec{w}$, given $\vec{u}$, $\vec{a}_a$, and the fact that $||\vec{g}_n|| \approx 9.81m/s^2$. We will assume that all measurements are expressed commonly aligned coordinate frames. We have already defined $\vec{u}$ in equations 3 and 4. Next we define $\vec{v}$, the vector rejection of $\vec{a}_a$ onto $\vec{u}$,

$$\vec{v} := \vec{a}_a - (\vec{a}_a \cdot \vec{u})\vec{u} \tag{6}$$

Using the Pythagorean Theorem we find $\vec{t}$,

$$\vec{t} := \left(\sqrt{||\vec{g}|| - ||\vec{v}||}\right) \vec{u} \tag{7}$$

Finally we find the two candidates for $\vec{w}$ by subtracting and adding $\vec{t}$ from the vector projection of $\vec{a}_a$ onto $\vec{u}$,

$$\vec{w} = (\vec{a}_a \cdot \vec{u})\vec{u} \pm \vec{t} \tag{8}$$

B. Dealing with Inconsistent Measurements

Thus far, we have been operating under the assumption that acceleration measurements according to the accelerometer $\vec{a}_a$ and according to SLAM odometry, $\vec{a}$ are error free and consistent. In practice both measurements are susceptible to noise, and can sometimes become inconsistent. In Figure 2, the outside boundaries represent the region around a given $\vec{u}$ where there exist allowable terminations of the $\vec{a}_a$ vector. If $\vec{a}_a$ falls outside those boundaries, there is no possible $\vec{g}_n$ that can resolve the two measurements, and no solution exits. If we try to compute it anyway, we find that $||\vec{v}|| > ||\vec{g_n}||$, and $\vec{t}$ becomes an imaginary vector.

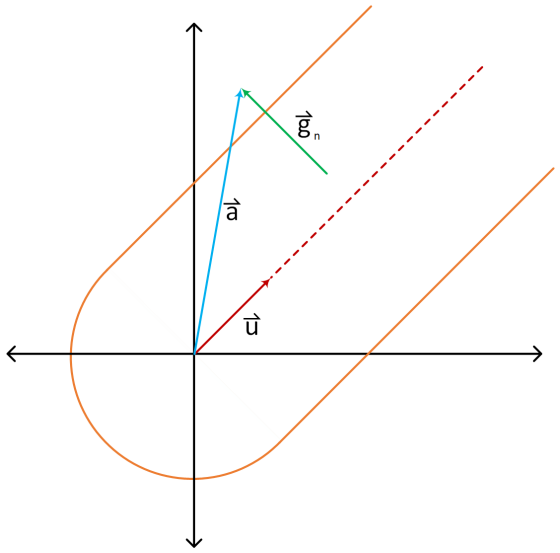


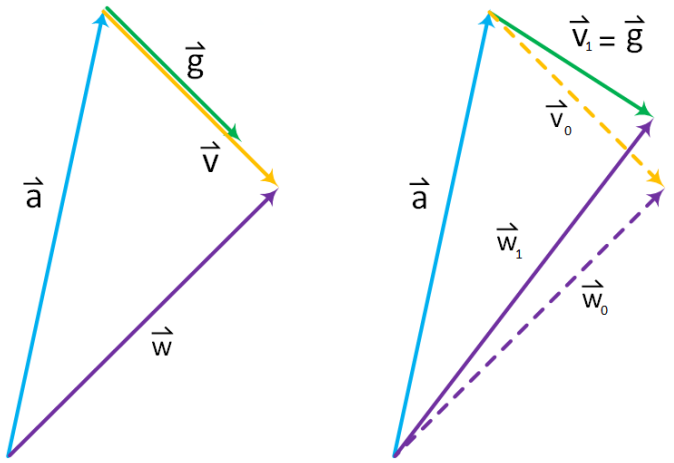
Fig. 5. When $\vec{a_a}$ lies outside the boundary, $\vec{g}$ cannot reach any scalar multiple of $\vec{u}$, and no solution exists.

In practice, significant horizontal acceleration places the end of the $\vec{a}_a$ vector close to the boundary of allowable values. When this happens, noise or offset in either the SLAM odometery or the accelerometer can frequently cause a no-solution condition. To get a solution, at least one of the parameters must change.

As $\vec{a}_a$ approaches the outer boundary of measurement consistency, $\vec{t}$ approaches $\vec{0}$. At the boundary, $\vec{t} = \vec{0}$, and Equation 8 simplifies to the vector projection of $\vec{a}_a$ onto $\vec{u}$,

$$\vec{w} = (\vec{a}_a \cdot \vec{u})\vec{u} \tag{9}$$

This is the shown by $\vec{a}_3$, in Figure 2 where $\vec{w}$, $\vec{v}$ and $\vec{a_a}$ make a right triangle, and $\vec{v} = \vec{g}$. If $\vec{a}_a$ extends into a disallowed region, the situation is similar, except $\vec{v}$ is longer than $\vec{g}$. We can get one approximate solution by using Equation 9. This approach essentially finds $\vec{w}$ with a larger value of $||\vec{g}||$. However, this is not the best option, since $||\vec{g}||$ is the one parameter that we know with the highest level of confidence. A better plan is to adjust one of the values that we have less confidence in. We can adjust either the direction or magnitude of $\vec{a}_a$, or the direction of $\vec{u}$. Since $\vec{u}$ is the least accurate measurement, this is the best parameter to change.



(a) Use larger value for $\vec{g}$ (b) Change angle of $\vec{u}s$

Fig. 6. It's easiest to just allow $\vec{g}$ to take on a larger value, but it's probably more accurate to change the angle of $\vec{u}$ and $\vec{w}$.

If $\vec{u}$ is rotated far enough towards $\vec{a}_a$, $||\vec{v}||$ will be equal to $||\vec{g}||$. This creates a new right triangle, where

$$||\vec{w_1}|| = \left(\sqrt{||\vec{a}||^2 - ||\vec{v_1}||^2}\right) \tag{10}$$

To find $\vec{w_1}$, we will use Rodrigues' rotation formula to rotate $\vec{a}_a$ into the direction of $\vec{w_1}$, and then scale the rotated vector to the known length of $\vec{w_1}$. It may seem strange that we are rotating $\vec{a}_a$ and not $\vec{u}$ or $\vec{w_0}$, since $\vec{u}$ is the parameter that is getting changed. The triangle diagrams in Figure 6 are two dimensional, but they are oriented arbitrarily in three dimensions. Using the known angle between $\vec{a}_a$ and $\vec{w_1}$, we can rotate $\vec{a}_a$ through our 2D plane in 3D space to get a vector in the vector space of $\vec{w_1}$. We obtain θ with

$$\theta = atan\left(\frac{||\vec{v_1}||}{||\vec{a}||}\right) \tag{11}$$

The rotation is then

$$\vec{l} = acos(\theta) + (k \times \vec{a}_a)sin(\theta) + k(k \cdot a)(1 - cos(\theta)) \tag{12}$$

where

$$\vec{k} = \frac{\vec{u} \times \vec{a}_a}{||\vec{u} \times \vec{a}_a||} \tag{13}$$

Because the rotation is entirely on the plane, $(k \cdot a) = \vec{0}$, and Equation 12 simplifies to

$$\vec{l} = acos(\theta) + (k \times \vec{a}_a^s)sin(\theta) \tag{14}$$

Finally, we get $\vec{w_1}$ by multiplying by the the ratio of the magnitudes of $\vec{w_1}$ and $\vec{a}_a$,

$$\vec{w_1} = \vec{l}\left(\frac{||\vec{w_1}||}{||\vec{a}_a||}\right) \tag{15}$$

C. Filtering Result

Clearly this approach requires a strong acceleration signal in order to produce a meaningful estimate or scale or the gravity vector. Typically scaling methods use filtering and machine learning techniques to refine the scale estimate over a period of time. New techniques will need to be developed in order to deal with the methods described in this paper. However, we have implemented a basic variance weighted average filter to improve the output signal. In order to roughly estimate the variance of the output, we assumed Gaussian distributed inputs and linearized all the equations around each sample. From these linearized approximations, we compute a covariance matrix for the output each step from the general form

$$\begin{aligned}\Sigma_s = \Sigma_{a_d} + B_a\Sigma_\alpha {B_a}^T + C_a\Sigma_\omega {C_a}^T \\ + B_a Cov(\alpha,\omega){C_a}^T + C_a Cov(\omega,\alpha){B_a}^T\end{aligned} \tag{16}$$

VI. Results

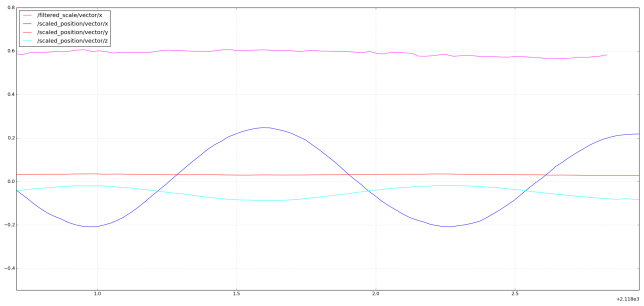
To test this method, we put together a simple device to explore the accuracy of the scale and gravity vector estimates. In order to test whether the scale factor produced is a realistic, we configured the software to produce a real time scaled graph of SLAM odometery position. We attached the camera to a drawer pull mounted to a two by four wood beam. By hand sliding the camera along this slide a known distance, we were able to compare a baseline camera displacement to the displacement indicated scaled SLAM odometery. The slide allowed movement to be constrained to a particular direction to make the displacement easier to visualize, and to allow for controlled testing of various movement angles. We do not compare these results with any other methods, because it is a proof of concept for a novel technique that has no direct comparison. Any comparison would need to be in the context of some application, with a refined hardware implementation.

Equipment was not available to directly test the accuracy of the gravity vector estimate, but the linear acceleration estimate is a good indicator of gravity vector accuracy. By moving the slide in a horizontal orientation, perpendicular to gravity, we can test to see of the method correctly shows acceleration only in the axis of movement.

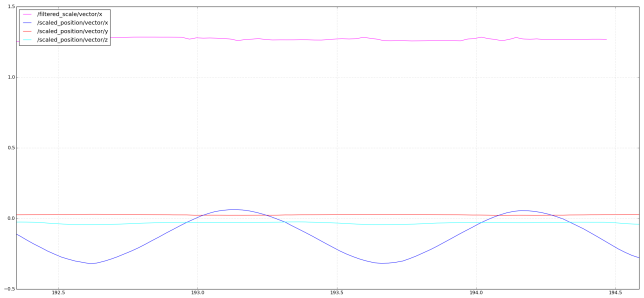
VII. Conclusion

The results of these tests indicate that the proposed scaling and gravity detection method is viable and promising. However, they do require robust, high speed pose estimation from the SLAM algorithm. The single largest limitation of performance is likely the SLAM odometery data rate. One of the most obvious applications for this method is hand held devices, but the 40hz framerate of the camera used in our implementation cannot quite capture enough high frequency detail in hand held motion. It seems likely that a framerate increase to even 60hz would make a big difference.

During testing we noticed that seemly small sources of timing error can make a surprisingly large difference in the

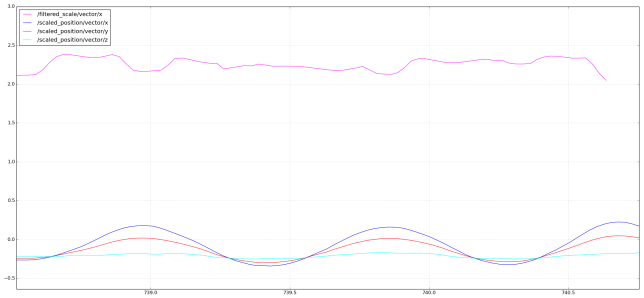


(a) Small Scale Environment

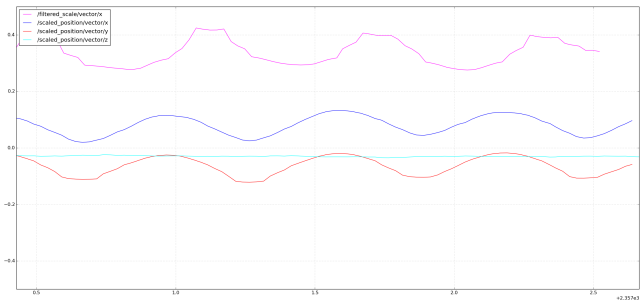


(b) Large Scale Environment

Fig. 7. Scaled horizontal position test. Shows x, y, and z displacement plus scale. Actual displacement amplitude is about 0.4 meters.



(a) Small Scale Environment



(b) Large Scale Environment

Fig. 8. Scaled angled position test. Shows x, y, and z displacement plus scale. Actual displacement is about .2 meter in the small scale and .4 meters in the large scale test.

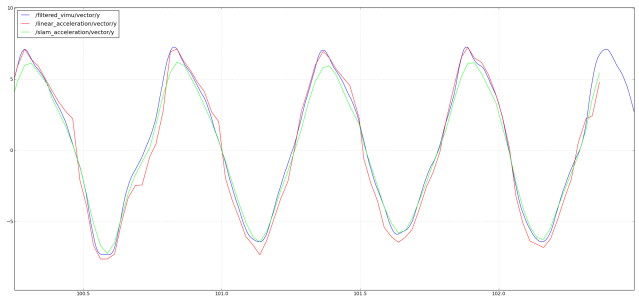


Fig. 9. Linear acceleration test.

accuracy of the output. For example, at one point it was discovered that a driver issue caused the timestamps on the IMU and the camera to be offset by about 12 milliseconds, which is only about half the period of the camera framerate. However, this offset was enough to completely destabilize the linear acceleration and scale estimates. During testing, the precise amount of delay drifted around by a few milliseconds and would cause issues unless re-calibrated. A number of other issues in the test implementation could easily corrected with a slightly higher budget. The mounting of the camera in relationship to the IMU is very imprecise, which decreases the accuracy of the VIMU. The IMU used had no calibration software, and as a result only had a very crude manual calibration. Correcting these issues, in addition to increasing the camera frame rate, could all improve performance.

It is important to remember what this method needs to accomplish. Right now, it constantly generates a new estimate for the location of gravity with each new frame. The accuracy of the gravity estimate in turn determines the accuracy of the scale. This isn't a realistic approach for a real world system. This is more of a initialization, because it does not require a-priori information, but as a-priori information becomes available, it should be used. The obvious extension of this approach would be to use the gyroscope and SLAM odometery to fuse multiple gravity estimates over time with a Kalman filter. This gravity estimate could then, in turn, be combined with estimates from other methods. A system implemented in this way could potentially offer a significant advantage in accuracy and robustness compared to current scaling schemes.

REFERENCES

[1] H. Kato and M. Billinghurst, "Marker tracking and hmd calibration for a video-based augmented reality conferencing system," in *Proceedings of the 2Nd IEEE and ACM International Workshop on Augmented Reality*, ser. IWAR '99. Washington, DC, USA: IEEE Computer Society, 1999, pp. 85–.

[2] S. B. Knorr and D. Kurz, "Leveraging the user's face for absolute scale estimation in handheld monocular slam," in *ISMAR*, 2016.

[3] G. Nützi, S. Weiss, D. Scaramuzza, and R. Siegwart, "Fusion of imu and vision for absolute scale estimation in monocular slam," *J. Intell. Robotics Syst.*, vol. 61, no. 1-4, pp. 287–299, Jan. 2011. [Online]. Available: http://dx.doi.org/10.1007/s10846-010-9490-z

[4] D. Bender, F. Rouatbi, M. Schikora, D. Cremersy, and W. Koch, "Scaling the world of monocular slam with ins-measurements for uas navigation," in *2016 19th International Conference on Information Fusion (FUSION)*, July 2016, pp. 1493–1500.

[5] A. Savitzky and M. J. Golay, "Smoothing and differentiation of data by simplified least squares procedures." *Analytical chemistry*, vol. 36, no. 8, pp. 1627–1639, 1964.