

Visual and Inertial Odometry for a Disaster Recovery Humanoid

Michael George and Alonzo Kelly

Abstract Disaster recovery robots must operate in unstructured environments where wheeled or tracked motion may not be feasible or where it may be subject to extreme slip. Many industrial disaster scenarios also preclude reliance on GNSS or other external signals as robots are deployed indoors or underground. Two of the candidates for precise positioning in these scenarios are visual odometry and inertial navigation. This paper presents some practical experience in the design and analysis of a combined visual and inertial odometry system for the Carnegie Mellon University Highly Intelligent Mobile Platform (CHIMP); a humanoid robot competing in the DARPA Robotics Challenge.

1 Introduction

Odometry is the process of measuring change in position over time without relying on external infrastructure. It is often used in combination with an absolute positioning system such as GPS [12] or to seed SLAM algorithms [8]. In certain scenarios it can also replace the need for these systems. One such scenario is tele-operation where a human is in the loop and performs the high level spatial reasoning and navigation while the odometry system works at a lower level to facilitate tasks like laser and image stitching or control feedback.

Odometry is often thought of as an open loop process, indeed, the canonical odometry techniques in 2D robotics involve one or two wheel encoders, whose measurements, over a given interval are passed through a simple kinematic model to

Michael George

National Robotics Engineering Center, Carnegie Mellon University, 10 40th St., Pittsburgh PA. U.S.A. 15221 , e-mail: mgeorge@cmu.edu

Alonzo Kelly

Robotics Institute, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh PA. U.S.A. 15213 , e-mail: alonzo@cmu.edu

produce an estimate of change in position [11]. This is sufficient in some applications but not in the presence of extreme wheel slip or terrain that is impassable to wheeled vehicles. Two alternatives are visual odometry, which is usually accomplished by directly measuring the change in position and orientation between two images and inertial navigation which involves the integration of acceleration and rotation rate signals. Both can function alone, but they have also have complementary features which make them ideal for combining. Table 1 provides a qualitative comparison of visual, inertial and wheel odometry.

Table 1 Qualitative comparisons of odometry techniques

Technique	Sensors	Characteristics
Inertial Navigation	IMU (\$15-\$100,000)	Acceleration, rotation rate measured by IMU at 100-2000Hz. Gravity model required. Position, velocity and orientation output at 100-2000Hz. Low computational complexity. Error grows as t for orientation, t^2 for velocity and t^3 for position.
Visual Odometry	Camera(s) (\$15-\$5000)	Limited by computational complexity at frame rates higher than 30 fps. Camera calibration required. Monocular cameras subject to scale uncertainty. High computational complexity. Error growth is complex function of camera intrinsics and scene characteristics but is generally not time dependent.
Wheel Odometry	Rotary encoder(s) (\$100-\$1000)	Shaft speeds measured by encoder converted to linear speeds with known wheel or track dimensions. Vehicle kinematic model (dimensions and steering mechanism) are required to convert linear speeds to vehicle velocity and heading rate. Error is a function of distance and kinematic model calibration as well as terrain and tire friction. Low computational complexity.

CHIMP contains all of these odometry systems and they are combined into a single best estimate of the robot's motion. This paper describes design and testing for the combination of inertial navigation and visual odometry on CHIMP.

2 CHIMP Robot

The Carnegie Mellon University Highly Intelligent Mobile Platform (CHIMP) is the Tartan Rescue (<http://www.rec.ri.cmu.edu/projects/tartanrescue/>) team entry in the ongoing DARPA Robotics Challenge (DRC). CHIMP is a humanoid robot that can also drive on two or four legs. The DRC event is designed to test competing robots in typical disaster recovery tasks like clearing rubble and closing valves and in ambitious locomotion strategies like climbing ladders and driving utility vehicles. CHIMP has a visual odometry system that uses a stereo camera pair from Pixim Inc. in it's head and a Honeywell Inc. HG9900 navigation grade IMU in it's torso. The cameras and IMU are separated by an articulable neck that allows CHIMP to adjust it's gaze. The visual odometry subsystem runs on a desktop grade processor

as part of CHIMP's ROS based computing platform while the inertial navigation and vision-inertial fusion algorithms run on an embedded processor in CHIMP's torso. The two subsystems communicate via user datagram protocol over CHIMP's onboard network.



Fig. 1 A rendering of CHIMP in four limb driving mode.



Fig. 2 A close-up of CHIMP's head, containing stereo cameras for visual odometry, amongst other sensors.

3 Inertial Navigation on CHIMP

Inertial navigation is a mature field in its traditional areas of application: aircraft, ships and military components. There are many excellent texts that outline the details [12], [9]. However these references typically make assumptions about the application which are not often true in field robots. Some examples include access to absolute initial heading, precise latitude information at turn on, knowledge of the IMU's height above the Earth's reference ellipsoid etc. This section discusses the IMU selected for CHIMP and the consequences associated with the above assumptions.

A summary of approximate IMU categories available in the market are given in Table 2. Most robotics applications that require an IMU will have one that falls into the Tactical grade category with cost and size being the major factor. Tactical grade IMUs are capable of tracking orientation for extended periods but cannot be used to track position for more than a few seconds in stand alone operation. Commercial grade sensors may be found in small robots where size and cost must be strictly minimized. With clever calibration they may also be capable of tracking orientation depending on the application.

Table 2 IMU categories

IMU grade	Size	Error Characteristics	Cost
Navigation	1600cm^3	< 1600 meters per hour in position, $< 1/5000$ degrees per hour in orientation.	\$70,000+
Tactical	100cm^3	$< 10^7$ meters per hour in position, < 50 degrees per hour in orientation.	\$2,000+
Commercial	1cm^3	$\sim 10^9$ meters per hour in position, < 3600 degrees per hour in orientation.	\$15+

A Monte Carlo simulation of a typical scenario for CHIMP was performed before selecting the IMU, the results were compelling enough to justify the choice of a navigation grade device, the Honeywell HG9900 [1]. Position and orientation errors for three devices that were considered in a trade study are shown in Figures 3 and 4. Considering the common uses for odometry information (3D perception, robot control etc.), the cost and size penalty of a navigation grade device were preferred to the alternative of greater reliance on vision and perception algorithms which are subject to environmental failures in low-light, low-texture, bright light etc.

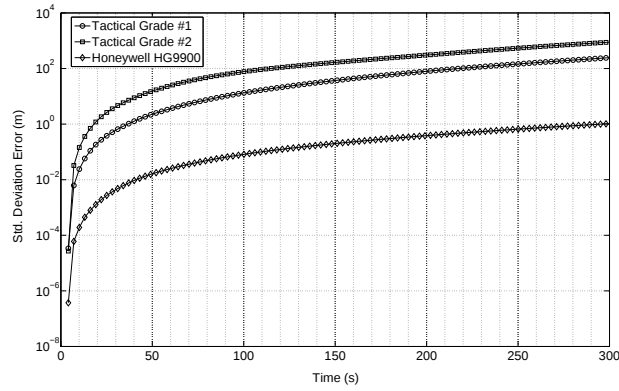


Fig. 3 Standard deviation of position errors for 5 minutes in simulated DRC scenario using an unaided IMU.

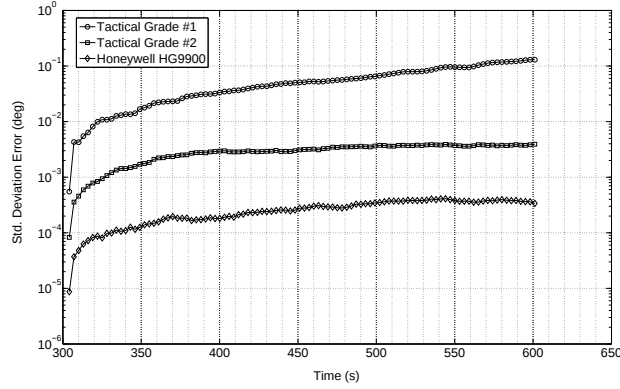


Fig. 4 Standard deviation of orientation errors for 5 minutes in simulated DRC scenario using an unaided IMU.

3.1 Inertial Navigation Equations

The fundamental equations of inertial navigation are [12]

$$\dot{\mathbf{C}}_b^n = \mathbf{C}_b^n [\omega_{ib}^b \times] - [\omega_{in}^n \times] \mathbf{C}_b^n \quad (1)$$

$$\dot{\mathbf{v}}^n = \mathbf{C}_b^n \mathbf{f}^b - (\omega_{en}^n + 2\omega_{ie}^n) \times \mathbf{v}^n + \mathbf{g}_p^n \quad (2)$$

where the following notation is used

- $\mathbf{x}^z$ Vector $\mathbf{x}$ expressed in frame z
- $\mathbf{C}_y^z$ Rotation matrix mapping frame y to frame z s.t. $\mathbf{x}^z = \mathbf{C}_y^z \mathbf{x}^y$
- $[\mathbf{x} \times]$ Cross product matrix version of $\mathbf{x}$ s.t. $[\mathbf{x} \times] \mathbf{y} = \mathbf{x} \times \mathbf{y}$
- ω_{xy}^z Angular rate vector of frame y relative to frame x expressed in frame z
- $\mathbf{g}_p^n$ Plumb-bob gravity, i.e. gravitation plus centripetal effects.
- $\mathbf{v}$ IMU velocity vector.
- $\mathbf{f}$ Non-gravitational acceleration a.k.a. specific force.
- i Inertial frame of reference
- b Body frame of reference, assumed to coincident with the IMU frame of reference
- n Navigation frame of reference, a design choice with subtle implications

It is interesting to examine the errors introduced by some common simplifications.

3.1.1 Static IMU Calibration

Most tactical grade IMUs have turn-on biases larger than Earth's rotation rate. In such a scenario, it's common to calibrate the gyroscopes at turn on by simply averaging their zero rate output and ascribing the result to turn on bias.

$$\mathbf{b}_\omega = \frac{1}{n} \sum_{i=1}^n \omega(i) \quad (3)$$

$$(4)$$

In reality all gyros, regardless of their quality, are measuring the Earth's own rotation rate (Approximately 15 degree/hour or 1 rotation per day). There are a growing number of tactical grade devices that have bias specifications under this floor [2]. The correct static calibration that takes into account Earth's rotation is given by

$$\mathbf{b}_\omega = \frac{1}{n} \sum_{i=1}^n \omega(i) - \mathbf{C}_b^{n\tau} \omega_{ie}^n \quad (5)$$

This expression now requires the constant $\mathbf{C}_b^n$ matrix that maps IMU orientation to the navigation frame during calibration. The tilt (pitch and roll) components of this are available from average accelerometer readings but the absolute heading is not available to tactical or commercial grade IMUs that cannot gyrocompass. This necessitates an additional sensor like a magnetometer or essentially you introduce a floor in the calibration accuracy that is on the order of the Earth's rate.

3.1.2 Gravity Model Error

Equation 1 requires a known gravity input. A typical model for this is [12]

$$g_p(0) = g_{equator} \frac{1.0 + 0.0019318513530 \sin^2 \lambda}{\sqrt{1.0 - 0.0066943800229 \sin^2 \lambda}} \quad (6)$$

$$g_{equator} = 9.7803267715 \quad (7)$$

which requires knowledge of the latitude of the IMU, λ . Without GPS, this latitude will be an approximation or a user input. In addition to varying with latitude, the gravity magnitude varies with altitude. Precise altitude is often unknown at the start of a robotic mission. There are local variations to the gravity field too [6], which are typically mapped by national and international standard's bodies but are mostly unavailable to robots in the field.

$$g_p(h) = \frac{g_p(0)}{(1.0 + \frac{h}{R_e})^2} \quad (8)$$

$$g_{equator} = 9.7803267715 \quad (9)$$

Figure 5 demonstrates the altitude accuracy attainable with and without correct initial altitude and local variations compensated for.

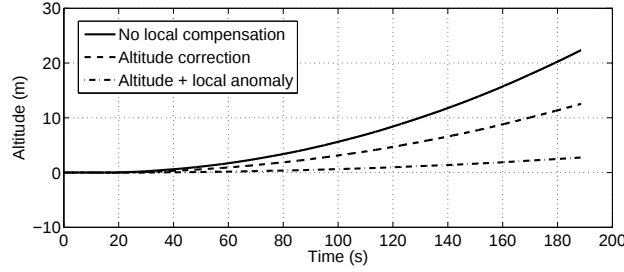


Fig. 5 Altitude errors with various levels of local compensation to the standard gravity model. Knowing the initial altitude of the test and the local gravity variations reduces total vertical error on CHIMP by 20 meters during a 3 minute task.

3.2 Initial Heading

With precise enough gyroscopes, initial absolute heading is attainable by gyrocompassing [12]. This involves averaging the gyroscopes, levelling the the resulting average and then taking the arctan of the expected measurement (Earth's rate) in each of the x, y axes

$$\bar{\omega} = \frac{1}{n} \sum_{i=1}^n \omega(i) \quad (10)$$

$$\bar{\omega}_{level} = \mathbf{C}_{\theta}^T \mathbf{C}_{\phi}^T \bar{\omega} \quad (11)$$

$$\psi = \arctan \frac{-\bar{\omega}_{level,y}}{\bar{\omega}_{level,x}} \quad (12)$$

Not precisely aligning to absolute North has an effect similar to that of calibrating the gyroscopes while neglecting Earth's rotation rate. The rotation rate of Earth will be incorrectly projected in the solution to the inertial equations which will quickly accumulate into position and orientation error. Figures 6 and 7 demonstrate the effect of assuming initial heading to be zero (North) instead of the correct value, in this case, of 33 degrees.

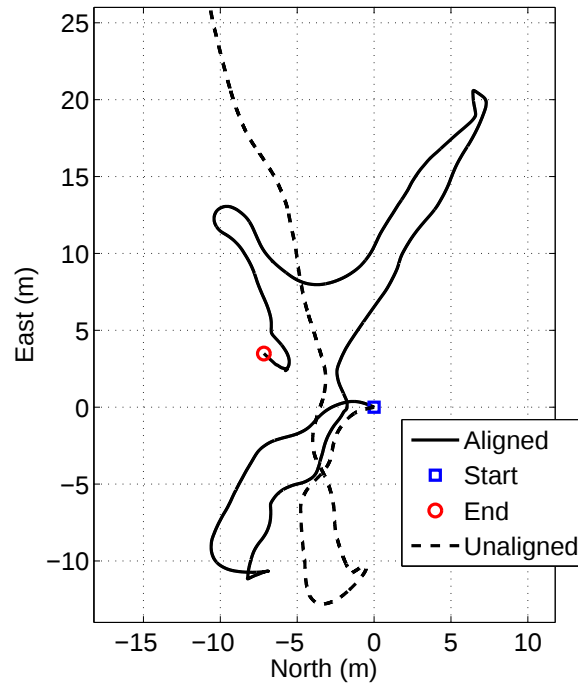


Fig. 6 INS only result with and without initial alignment to absolute North. Without precise initial alignment, incorrectly resolved components of the Earth's rotation rate quickly accumulate into position errors.

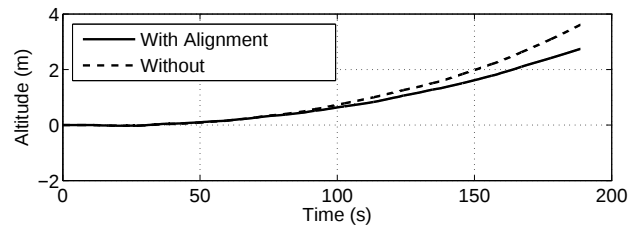


Fig. 7 Altitude error with and without alignment to absolute North. Since the misaligned components of Earth's rotation rate are in the local plane there is little effect on altitude error.

4 Visual Odometry

CHIMP's visual odometry system produces a solution based on incremental structure from motion with optional key frame selection and sparse local bundle adjustment. Details can be found in [10] and the references therein. The algorithm can produce relative pose solutions at frame rate or between opportunistically chosen key frames. It makes use of a stereo pair to eliminate scale ambiguity in the resulting solution.

Like most visual odometry systems, it solves consecutive relative pose problems from image correspondences and scene structure estimates. Each pose increment is relative to the previous camera location, denoted $c(k-1)$. At each frame k , the following measurements are computed

$$\mathbf{z}_{translation} = \Delta \mathbf{p}_{c(k-1) \rightarrow c(k)}^{c(k-1)} \quad (13)$$

$$\mathbf{z}_{rotation} = \mathbf{C}_{c(k)}^{c(k-1)} \quad (14)$$

$$= \mathbf{C}_{c(k-1)}^{c(k)\top} \quad (15)$$

Given initial conditions $\mathbf{p}_{vo}^n(0)$ and $\mathbf{C}_{c(0)}^n$ A global pose at increment k can be tracked via

$$\mathbf{p}_{c(k)}^n = \mathbf{p}_{c(0)}^n + \sum_{i=1}^k \mathbf{C}_{c(i)}^n \Delta \mathbf{p}^{c(i)} \quad (16)$$

$$\mathbf{C}_{c(k)}^n = \mathbf{C}_{c(0)}^n \prod_{i=1}^k \mathbf{C}_{c(i)}^{c(i-1)} \quad (17)$$

Alternatively when key frames are not in use and the algorithm is running at frame rate (30fps) the translation component can be numerically differentiated to produce a velocity signal

$$\mathbf{z}_{velocity} = \frac{\Delta \mathbf{p}_{c(k-1) \rightarrow c(k)}^{c(k-1)}}{t(k) - t(k-1)} \quad (18)$$

Each of these three forms is available to a fusion algorithm that blends the visual odometry data with the inertial navigation system data in real-time to produce an optimal estimate of CHIMP's pose.

5 Kalman Filter for Visual and Inertial Odometry

Visual and inertial estimates are blended in an extended Kalman filter. The Kalman filter is a well understood algorithm and has a massive body of literature in regards to aiding inertial navigation in particular. In addition it can be easily embedded for robust field applications.

The filter uses the common indirect inertial navigation state model [12]. In it's traditional form the Kalman filter equations are [4]

Prediction:

$$\mathbf{x}_{k+1} = \Phi_k \mathbf{x}_k + \mathbf{G}_k \mathbf{w}_k \quad (19)$$

$$\mathbf{P}_{k+1} = \Phi_k \mathbf{P}_k \Phi_k^\top + \mathbf{G}_k \mathbf{Q} \mathbf{d}_k \mathbf{G}_k^\top \quad (20)$$

$$(21)$$

Measurement:

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) \quad (22)$$

$$(23)$$

Update:

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_k \mathbf{H}_k^\top + \mathbf{R}_k \quad (24)$$

$$\mathbf{H}_k = \frac{\partial \mathbf{h}(\mathbf{x}_k)}{\partial \mathbf{x}_k} \quad (25)$$

$$\mathbf{K}_k = \mathbf{P}_k \mathbf{H}_k^\top \mathbf{S}_k^{-1} \quad (26)$$

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^\top + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^\top \quad (27)$$

$$\mathbf{x}_k^+ = \mathbf{x}_k + \mathbf{K}_k (\mathbf{z}_k - \mathbf{h}(\mathbf{x}_k)) \quad (28)$$

Where the following definitions apply

- $\mathbf{x}_k$ State mean at time k
- Φ State transition matrix mapping state evolution over time
- $\mathbf{G}$ Noise input matrix mapping model uncertainty to states
- $\mathbf{P}$ State covariance matrix
- $\mathbf{Q} \mathbf{d}$ Process noise matrix accounting for uncertainty inputs
- $\mathbf{z}$ Measurement vector
- $\mathbf{h}$ Measurement function mapping states to measurements
- $\mathbf{H}$ Measurement matrix, jacobian of $\mathbf{h}$ w.r.t $\mathbf{x}$
- $\mathbf{K}$ Kalman gain matrix

These equations are suitable for incorporating an absolute visual odometry pose or a velocity measurement (Equations 16 and 18). They are not suitable for incorporating the fundamental measurement from the visual odometry system, Equation 13, the relative state measurement. It has the following modified measurement form

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k, \mathbf{x}_{k-1}) \quad (29)$$

The standard filter equations have been modified to incorporate this measurement form in [4]. The modified equations take the following form

$$\mathbf{L}_k = \mathbf{H}_k \mathbf{P}_k \mathbf{H}_k^\top + \mathbf{R}_k + \mathbf{J}_k \mathbf{P}_{k-1} \boldsymbol{\Phi}_{k-1}^\top \mathbf{H}_k + \mathbf{H}_k \boldsymbol{\Phi}_{k-1} \mathbf{P}_{k-1} \mathbf{J}_k^\top + \mathbf{J}_k \mathbf{P}_{k-1} \mathbf{J}_k^\top \quad (30)$$

$$\mathbf{J}_k = \frac{\partial \mathbf{h}(\mathbf{x}_k, \mathbf{x}_{k-1})}{\partial \mathbf{x}_{k-1}} \quad (31)$$

$$\mathbf{K}_k = (\mathbf{P}_k \mathbf{H}_k^\top + \boldsymbol{\Phi}_{k-1} \mathbf{P}_{k-1} \mathbf{J}_k^\top) \mathbf{L}_k^{-1} \quad (32)$$

$$\mathbf{P}_k^+ = \mathbf{P}_k - \mathbf{K}_k \mathbf{L}_k \mathbf{K}_k^\top \quad (33)$$

$$\mathbf{x}_k^+ = \mathbf{x}_k + \mathbf{K}_k (\mathbf{z}_k - \mathbf{h}(\mathbf{x}_k, \mathbf{x}_{k-1})) \quad (34)$$

Where the $k-1$ parameters must be stored in a memory buffer or retrodicted explicitly using the $\mathbf{x}$ and $\mathbf{P}$ variables with saved versions of $\boldsymbol{\Phi}_{k-1}$ and $\mathbf{Qd}_{k-1}$ terms [3].

6 Delayed Measurements

Visual odometry is a costly process and is performed in it's own computing cores separate from the embedded navigation processor. The complexity of the computation (30 fps maximum throughput) and the additional communications delay introduce a non-trivial time synchronization problem. Delays of up to 100ms can occur between capturing an image and that data being available to the Kalman filter. In this case the previous relative state update relations can be modified to use the velocity form of the visual odometry (Equation 18) by setting the current measurement jacobian to zero, giving

$$\mathbf{H}_k = \mathbf{0} \quad (35)$$

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_{k-1}) \quad (36)$$

$$\mathbf{L}_k = \mathbf{R}_k + \mathbf{J}_k \mathbf{P}_{k-1} \mathbf{J}_k^\top \quad (37)$$

$$\mathbf{K}_k = (\boldsymbol{\Phi}_{k-1} \mathbf{P}_{k-1} \mathbf{J}_k^\top) \mathbf{L}_k^{-1} \quad (38)$$

$$\mathbf{P}_k^+ = \mathbf{P}_k - \mathbf{K}_k \mathbf{L}_k \mathbf{K}_k^\top \quad (39)$$

$$\mathbf{x}_k^+ = \mathbf{x}_k + \mathbf{K}_k (\mathbf{z}_k - \mathbf{h}(\mathbf{x}_{k-1})) \quad (40)$$

These now represent the equations necessary to apply a delayed measurement that was valid at time $k-1$ but arrived at the filter at time k . It should be noted that the interval between ks is not required to be constant. Given the form of the measurement function here, it is only suitable for velocity or absolute pose versions of the visual odometry data (Equations 16 and 18).

7 Calibration

It is necessary to calibrate the relative position and orientation of the visual odometry cameras and IMU on CHIMP. Assuming a rigid link the following relation holds, where k and j are time indices and b represents the IMU frame while c is the camera frame

$$\mathbf{C}_{b_k}^{bj} = \mathbf{C}_c^b \mathbf{C}_{c_k}^{c_j} \mathbf{C}_c^{b\top} \quad (41)$$

$$(42)$$

This can be converted to a quaternion form and solved for the relative rotation matrix between camera and IMU $\mathbf{C}_c^b$. Details can be found in [5]. For the translation, most methods involve a calibration Kalman filter [7]. These methods are complex but are necessary since they are designed for low-cost IMUs whose position estimates drift rapidly when uncorrected. The following relation maps the relative pose increments generated by the visual odometry system with similar increments calculated from the CHIMPs inertial navigation system

$$\Delta \mathbf{p}_{c_{k-1} \rightarrow c_k}^{c_{k-1}} = \mathbf{C}_b^c \mathbf{C}_n^{b_{k-1}} (\mathbf{p}_{b_k}^n + \mathbf{C}_{b_k}^n \mathbf{p}_{b \rightarrow c}^b - \mathbf{p}_{b_{k-1}}^n - \mathbf{C}_{b_{k-1}}^n \mathbf{p}_{b \rightarrow c}^b) \quad (43)$$

$$(44)$$

Re-arranging for $\mathbf{p}_{b \rightarrow c}^b$, the position offset between IMU (b) and camera (c)

$$\mathbf{C}_b^c (\mathbf{C}_{b_k}^{b_{k-1}} - \mathbf{I}) \mathbf{p}_{b \rightarrow c}^b = \Delta \mathbf{p}_{c_{k-1} \rightarrow c_k}^{c_{k-1}} + \mathbf{C}_b^c \mathbf{C}_n^{b_{k-1}} (\mathbf{p}_{b_{k-1}}^n - \mathbf{p}_{b_k}^n) \quad (45)$$

$$(46)$$

which stacked over many j, k intervals has the form

$$\mathbf{A} \mathbf{p}_{b \rightarrow c}^b = \mathbf{b} \quad (47)$$

A least squares solution is given by $\mathbf{p}_{b \rightarrow c}^b = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b}$. This can be transformed into a weighted least squares with the standard INS error model from the Kalman filter prediction steps used to generate covariances as the IMU solution drifts. Given these covariances the weighted least squares solution then becomes

$$\mathbf{p}_{b \rightarrow c}^b = (\mathbf{A}^\top \mathbf{P}^{-1} \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{P}^{-1} \mathbf{b} \quad (48)$$

8 Results

The preceeding algorithms were implemented and tested on a test rig consisting of a partially constructed CHIMP, see Figure 8.



Fig. 8 A partially constructed CHIMP on a test-rig. The sensor head containing visual odometry sensors and a single arm are shown. The IMU is located in a temporary enclosure at the base of a mobile frame (not shown).

Testing is still underway but initial results in an indoor environment where the rig was manually pushed are shown in Figures 9 and 10. The rig was moved through a variety of motions (sweeping arcs, point turns etc.) and returned to its original position and orientation to measure error. Future testing will use a motion capture system for more precise quantification. The initial use for this data is in generating 3D perception to transmit to CHIMP's operator station for situational awareness and manipulation tasks.

Acknowledgements Michel Laverne and Dane Bennington designed and built much of the positioning hardware. JP Tardif provided the visual odometry subsystem.

References

1. Honeywell Aerospace (2009) HG9900 IMU
<http://www51.honeywell.com/aero/common/documents/myaerospacecatalog-documents/MilitaryAC/HG9900.IMU.pdf>
2. KVH Industries (2011) CG-5100 KVH's Commercial IMU Solution.
<http://www.kvh.com/ViewAttachment.aspx?guidID=230CBC8B-2733-4FE7-B860-DD271A6E4476>. Cited 16 Sep 2013
3. Bar-Shalom Y (2002) Update with Out-of-Sequence Measurements in Tracking: Exact Solution. *IEEE Trans Aerosp Electron Syst* vol 38 no 3 769–778
4. Brown R, Hwang P (1997) *Introduction to Random Signals and Applied Kalman Filtering*, 3rd edn. John Wiley & Sons, New York.

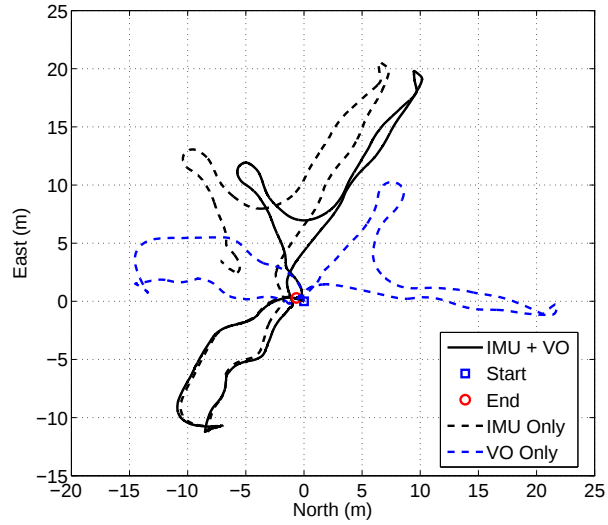


Fig. 9 Indoor path of 105m, with error assessed as closure discrepancy when returning to the origin. 2D error is 72cm or 0.6% of distance. VO Only has an arbitrary initial alignment. IMU only results drift 8m.

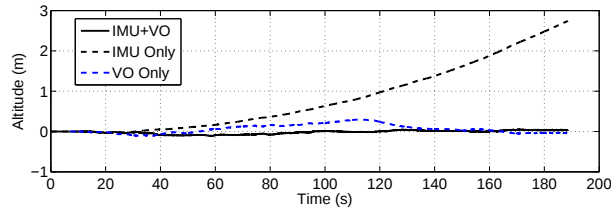


Fig. 10 Indoor path of 105m, with error assessed as closure discrepancy when returning to the origin. Altitude error is 3cm at loop closure. Ground truth altitude is not available in the middle of the segment.

5. Dong-Si T, Mourikis A (2012) Estimator Initialization in Vision-aided Inertial Navigation with Unknown Camera-IMU Calibration. Proc. IEEE/RSJ IROS Oct 7-12 1064–1071
6. Farrell (2008) Aided Navigation: GPS with High Rate Sensors. McGraw Hill, New York.
7. Hol J, Schon T, Gustafsson F (2008) A New Algorithm for Calibrating a Combined Camera and IMU Sensor Unit. Proc. IEEE ICARCV Dec 17-20 1857–1862
8. Olson E, Agarwal P (2012) Inference on networks of mixtures for robust robot mapping. Proc. Robotics: Science and Systems July 2012.
9. Savage P (2007) Strapdown Analytics, 2nd edn. Strapdown Associates Inc. Maple Plain MN.
10. Tardif J, George M et. al. (2010) Inference on networks of mixtures for robust robot mapping. Proc. IEEE/RSJ IROS Oct 18-22 4161–4168
11. Thrun S, Burgard W, Fox D (2005) Probabilistic Robotics. MIT Press, Cambridge MA.
12. Titterton D, Weston J (2004) Strapdown Inertial Navigation Technology, 2nd edn. IEE Radar, Sonar and Navigation Series 17. IEE, Reston VA.