JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 29, No. 4, July–August 2006

Visual Positioning System for an Underwater Space Simulation Environment

Jeffrey R. Smithanik,* Ella M. Atkins,† and Robert M. Sanner‡
University of Maryland, College Park, Maryland 20742

Neutral buoyancy simulation of the three-dimensional space environment enables long-term training and evaluation for astronaut and robotic operations prior to attempting these activities in space. The underwater environment, however, does not allow use of global positioning system or pseudolite equivalents, and sonar has significant limitations when used in a small, reflective tank. This paper describes an eight-camera visual positioning system (VPS) to provide real-time three-dimensional position and velocity estimates for free-flying neutral buoyancy robots performing tasks such as spacecraft inspection and servicing. A long-baseline calibration procedure is developed to estimate accurate intrinsic and extrinsic calibration parameters. An extended Kalman filter merges camera measurements with robot telemetry to create an optimal three-degrees-of-freedom estimate of translational position and velocity. Static tests indicate VPS is capable of locating a robot with subcentimeter accuracy under ideal conditions, and more typically, three-to four-centimeter accuracy when background clutter or glare influence processed image measurements. A series of dynamic tests show centimeter-level accuracy so long as the vehicle is viewable with multiple cameras, indicating VPS is a viable navigation system for neutral buoyancy operations.

Nomenclature

C_X, C_Y	=	digital image principal point
f	=	camera focal length
K_t, K_r, K, K_A, K_B	=	radial distortion parameters
L, L_k	=	observer/estimator gain matrix
P, P_k, P_{k-1}	=	estimate error covariance matrix
${}^G\boldsymbol{P}, {}^C\boldsymbol{P}$	=	global (G), camera (C) object position
${}^C\boldsymbol{P}_{G,\text{org}}$	=	translation from camera to global frame in camera coordinates $[T_X \;\; T_Y \;\; T_Z]^T$
Q, Q_k, Q_{k-1}	=	process noise covariance matrix
${}^C_G R = [r_{11}, \ldots, r_{33}]$	=	rotation matrix (global to camera)
R_k	=	measurement noise covariance matrix
$(R_X, \;\; R_Y, \;\; R_Z)$	=	rotation about fixed camera X axis (X_C), Y axis (Y_C), and Z axis (Z_C)
s_X	=	ratio of pixel spacing in X and Y
$\boldsymbol{u}$	=	control input vector
v_i	=	velocity along $i = X, Y, Z$
$[X_C \;\; Y_C \;\; Z_C]^T$	=	camera frame coordinates
$[X_D \;\; Y_D]$	=	real distorted image coords, mm
$[X_{\text{FD}} \;\; Y_{\text{FD}}]$	=	distorted image coords, pixels
$[X_{\text{FU}} \;\; Y_{\text{FU}}]$	=	undistorted images coords, pixels
$[X_G \;\; Y_G \;\; Z_G]^T$	=	global (inertial) coordinates
$[X_U \;\; Y_U]$	=	real undistorted image coordinates, mm
$[\dot{X}_G \;\; \dot{Y}_G \;\; \dot{Z}_G]^T$	=	global (inertial) velocities
$\boldsymbol{x}$	=	estimate error vector
$\boldsymbol{x}, \dot{\boldsymbol{x}}$	=	translational state and derivatives ($\hat{}$ indicates estimated state)
z	=	measurement vector
Λ_{k-1}	=	discrete input propagation matrix
Φ_{k-1}	=	discrete state transition matrix

Received 16 February 2005; revision received 10 April 2005; accepted for publication 21 April 2005.

*Graduate Research Assistant, Aerospace Engineering Department, Space Systems Laboratory; jsmithanik@mpr.com.

†Assistant Professor, Aerospace Engineering Department, Space Systems Laboratory; atkins@eng.umd.edu. Senior Member AIAA.

‡Associate Professor, Aerospace Engineering Department, Space Systems Laboratory; rmsanner@eng.umd.edu.

Introduction

SPACE operations introduce many challenges for human and robotic explorers. To minimize risk and maximize chance of success, systems must be thoroughly tested prior to launch, but such testing is difficult because of environmental differences, most notably the ability to move freely in three dimensions. Earth-based techniques to simulate the space environment exist, such as air-bearing tables,[1] aircraft following parabolic trajectories (NASA's KC-135),[2] drop towers,[3] overhead weight-bearing cables,[4] and six-degrees-of-freedom (DOF) gantry robot manipulators.[5] Neutral buoyancy simulation (Figs. 1c and 1d) has been demonstrated as a valuable training environment for astronauts and is utilized for a variety of human and robotic system tests, including ongoing efforts to plan and validate a robotic servicing mission for Hubble Space Telescope. Although no Earth-based simulator is ideal, neutral buoyancy is uniquely capable of long-duration, three-dimensional space simulation for free-flying vehicle systems and is the simulation environment studied in this work.

Spacecraft require six-DOF navigation and control systems, three DOF for attitude, and three DOF for translation. Attitude can be determined with inertial measurement units on virtually all space simulation platforms. Estimation of translational position and motion, however, has proven challenging for environments such as neutral buoyancy where neither GPS nor pseudolite[6] signals can be propagated.

Underwater sonar-based systems have been developed for undersea navigation, but success with these systems has been limited because of the highly reflective nature of an enclosed tank and the requirement to place hydrophone or emitter hardware onboard each free-flying robot (or even astronaut) that requires navigation support.[7,8]

This paper describes a machine vision system for translational navigation in a neutral buoyancy environment. This visual positioning system (VPS) must provide three-dimensional position information with accuracy comparable to global positioning system (GPS). This vision system has been implemented in the University of Maryland's Neutral Buoyancy Research Facility (NBRF), shown in Fig. 1c, and has been evaluated during navigation of the free-flying Supplemental Camera Platform (SCAMP) space simulation vehicle (Fig. 1a), an underwater analog to NASA Johnson Space Center's (JSC) autonomous extravehicular activity (EVA) robotic Camera (AERCam) platform developed for space inspection tasks.[9]

Vision systems designed for aerospace applications typically perform relative navigation, using onboard camera(s) to keep a certain

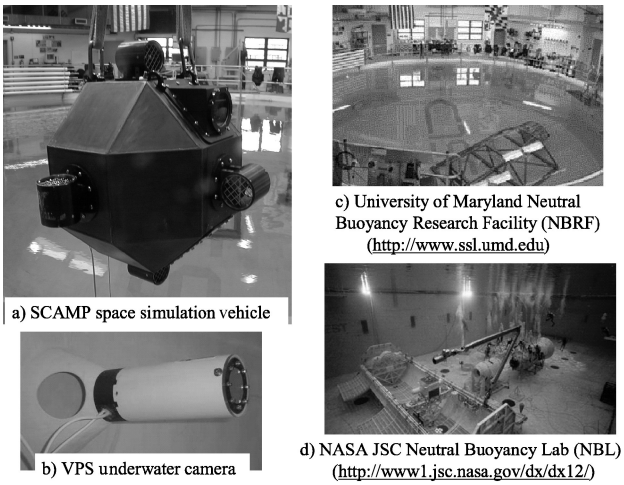


Fig. 1 Visual positioning system (VPS) for a neutral buoyancy environment.

distance and bearing from other vehicle(s)/terrain or to rendezvous and dock with another vehicle.[10–12] Although analysis is often restricted to simulation because of cost and platform availability, experimental results with integrated camera hardware are beginning to emerge.[13] For SCAMP, a color tracking system was previously implemented to follow a uniquely colored target in the NBRF,[14] but this approach required a leader to be constantly kept in view. Navigation is also possible with only onboard camera systems, provided unique markings (fiducials) exist or distinct features (landmarks) can be identified at all times. Vision-based methods have proven useful for automatic learning of landmark appearance in nonengineered environments,[15] velocity estimation from optical flow,[12,16] environment feature matching,[17] and pose estimation from known maps.[18] In our cylindrical NBRF environment, however, few landmarks exist, and those that are present are typically not unique (e.g., portholes). Because engineering the NBRF with fiducials/landmarks would itself be a substantial engineering effort, VPS was designed to use inertially fixed cameras to characterize the positions and velocities of visual targets (e.g., a free-flying robot).

The VPS implemented in this work is comprised of eight charge-coupled-device (CCD) cameras rigidly mounted to the walls of the NBRF (as shown in Fig. 1b) to track underwater robot motion. Because all VPS cameras and support hardware are external to the vehicle, this system can be used without vehicle modification and can track a robot or even astronauts (divers) that have no special equipment to support visual navigation. This paper describes the design of VPS, a robust method to compute intrinsic and extrinsic camera calibration parameters, and application of an Extended Kalman Filter (EKF) to merge camera measurements and vehicle dynamics into real-time state estimates. Accuracies of the calibration and of EKF state estimates are assessed during static and dynamic SCAMP flight tests. There are two fundamental contributions of this work: a two-step calibration procedure for accurate intrinsic parameter estimation and large-scale extrinsic calibration without a planar observation target, and an EKF derived to combine two-dimensional (x-y) camera measurements into a full three-dimensional estimate of position and velocity.

Relevant hardware and software components are outlined next, followed by a description of the camera model and its calibration parameters. The calibration procedure is described, and the EKF equations are derived. The system has been implemented and tested with the SCAMP neutral buoyancy vehicle. Both calibration and EKF output are evaluated over static and dynamic tests.

Hardware and Software Systems

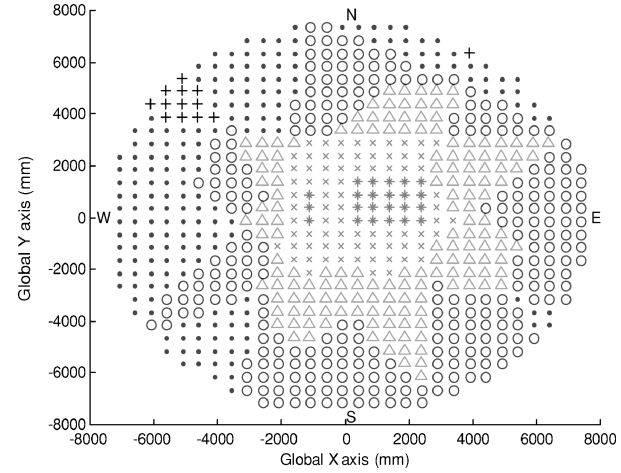
The University of Maryland's NBRF is a cylindrical, fiberglass tank of water 7.62 m (25 in.) deep and 15.24 m (50 in.) in diameter. The water is filtered for exceptional visual clarity, providing an environment well suited for machine vision. A group of 12 rigid mounting locations (hard points) is arranged in rings of four hard points spaced at 90-deg intervals at three tank depths. The eight VPS cameras are attached to the top and middle rings, with middle ring hard points offset 45 deg from the top ring stations. This choice was made because test hardware would generally block views from bottom ring mounting stations. VPS is composed of relatively low-cost cameras with 768×494 resolution. Each camera (see Fig. 1b) is sealed in a waterproof box, mounted to a hard point, and connected to power and a frame grabber on the surface. Middle ring camera boxes point radially inward, whereas top ring cameras are fitted with a single pivot axis that allow the cameras to be tilted down for maximal coverage. Camera coverage in the NBRF varies with position. Cameras were tilted and focused manually and characterized through calibration, given that mounting point imprecisions themselves preclude symmetric camera positions. Figure 2 illustrates camera coverage at the depths 7 ft (2.1 m), 12.5 ft (3.8 m), and 18 ft (5.5 m), and includes a legend indicating number of cameras that can view each tank location.

With appropriate object recognition software, VPS could track any object in the tank. However, this work was focused on navigation for the free-flying SCAMP vehicle. SCAMP has six bidirectional thrusters aligned in pairs along vehicle body axes to provide full six-DOF control authority. To simulate space, SCAMP is balanced to be neutrally buoyant in both translation and rotation, distinct from typical undersea vehicles that maintain a stable keel. Powered from onboard batteries, SCAMP contains a 100-MHz computer running VxWorks connected by Ethernet fiber optic link to a Linux-based surface control station. An onboard inertial measurement unit (IMU) has enabled attitude estimation and control,[19] and an onboard video from a single or stereo camera pair is transmitted over fiber-optic link to provide a view of the environment.

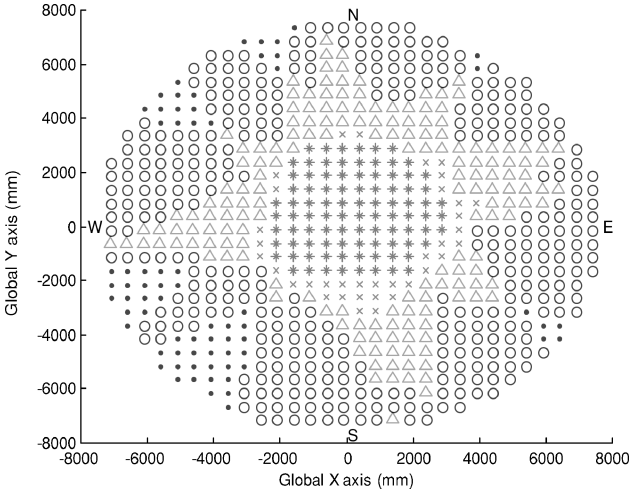
Although the long-term goal with VPS is fully autonomous flight operations, at present, an operator transmits flight commands with a pair of space shuttle hand controllers, one for translation and one for rotation. A graphical user interface (GUI) displays real-time telemetry from the robot, and EKF software on the control station integrates vehicle telemetry and camera measurements to compute vehicle position and velocity. Four computers (Vision1-Vision4) each house two FlashBus™ single-channel frame grabber boards. Figure 3 shows the system diagram of SCAMP, VPS computers, control station, and UDP Ethernet communication links. After initialization, the VPS computers run a program (VPS_client) to acquire images, process the image to find the area and centroid of SCAMP, and send this data to the control station that then computes state estimate $[X_G \;\; \dot{X}_G \;\; Y_G \;\; \dot{Y}_G \;\; Z_G \;\; \dot{Z}_G]^T$. Each VPS_client process receives an updated position estimate from the control station to determine if the vehicle is in each camera's field of view (FOV). If the vehicle is not visible to a camera, no images will be acquired from that camera until the vehicle is visible. A frame is grabbed and processed from each camera at approximate 8 Hz ($\sim$16 Hz loop rate, two cameras per computer). The main control station program includes a GUI, reads hand controllers, and manages/saves telemetry and VPS data. The VPS EKF also runs on the control station computer at regular intervals (10 Hz) controlled by a timer. SCAMP flight software has three threads: communication (50 Hz), attitude estimation (25 Hz), and controller (25 Hz). The communication thread receives pilot commands and uplinks telemetry, the attitude estimation thread reads the IMU and computes attitude/angular velocities with an onboard EKF, and the controller computes thruster forces open-loop or closed-loop (attitude).

Camera Model

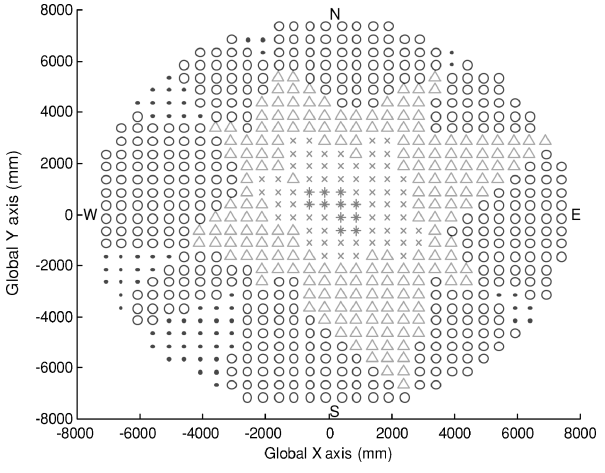
The VPS camera model was derived from Tsai.[20] Calibration parameters are divided into two sets: intrinsic and extrinsic. Intrinsic parameters model the interaction of light with optical and electronic components inside the camera and include: focal length f (mm), principal point coordinates on the image plane C_x, C_y (pixels), first-order radial lens distortion parameter K, and scaling factor s_x to account for the translation from CCD sels to pixels scanned in the horizontal x direction. The extrinsic parameters define the translation and rotation of a camera with respect to an inertial coordinate frame. With an XYZ Euler angle representation, extrinsic



a) Camera coverage, depth = 7 ft



b) Camera coverage, depth = 12.5 ft



c) Camera coverage, depth = 18 ft

Fig. 2 VPS camera coverage maps; + = 0 cams; ● = 1 cams; ○ = 2 cams; △ = 3–4 cams; × = 5–6 cams; ∗ = 7–8 cams.

parameters include rotation angles (R_x, R_y, R_z) and translation vector ${}^{C}\boldsymbol{P}_{G,\text{org}} = (T_x, T_y, T_z)$.

Consider a three-dimensional station $[X_G \;\; Y_G \;\; Z_G]^T$ in inertial (global) coordinates. Extrinsic parameters $(R_x, R_y, R_z, T_x, T_y, T_z)$ describe the conversion of $[X_G \;\; Y_G \;\; Z_G]^T$ to or from local camera frame coordinates $[X_C \;\; Y_C \;\; Z_C]^T$. Perspective projection describes the conversion from the local camera frame to image plane coordinates. Camera coordinate Z_C specifies range to an object, and

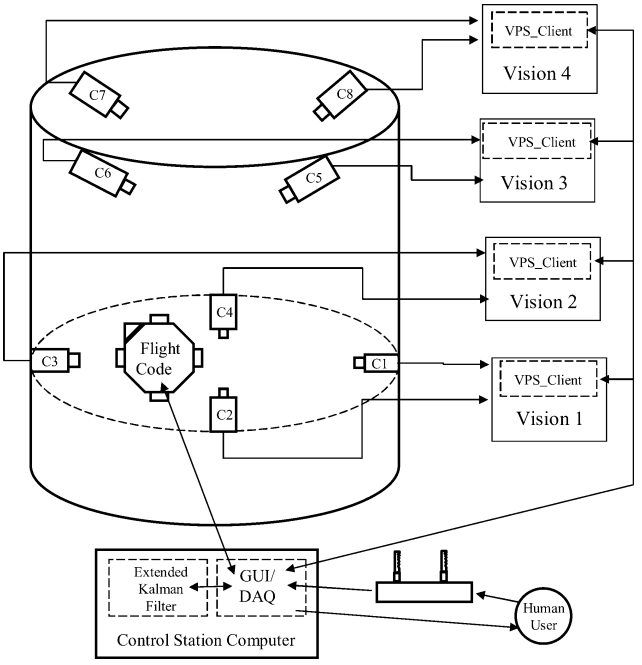


Fig. 3 VPS architecture and fixed camera locations.

assuming a pinhole camera model, the undistorted image plane coordinates are approximated by

$$\left[X_U = (f \cdot X_C)/Z_C, Y_U = (f \cdot Y_C)/Z_C, A = \pi \cdot r^2\right] \quad (1)$$

where (X_C, Y_C) are the undistorted image coordinates, A is object area, and $r = (f \cdot R)/Z_C$ is object radius. Focal length f is the only calibration parameter required for this conversion.

Camera systems composed of optical elements with spherical surfaces suffer from unavoidable geometric distortions. Image coordinates will actually be displaced farther from (pincushion distortion) or closer to (barrel distortion) the optical axis compared to the coordinates predicted by the pinhole projection model. Distortion occurs both radially and tangentially from the optical axis. Displacement caused by radial distortion is modeled by

$$\partial X_r = X_D \cdot \left(K_{r1} \cdot r^2 + K_{r2} \cdot r^4 + \cdots\right) \quad (2)$$

$$\partial Y_r = Y_D \cdot \left(K_{r1} \cdot r^2 + K_{r2} \cdot r^4 + \cdots\right) \quad (3)$$

where $r^2 = X_D^2 + Y_D^2$. Similarly, displacement caused by tangential distortion is modeled by

$$\partial X_t = -Y_D \cdot \left(K_{t1} \cdot r^2 + K_{t2} \cdot r^4 + \cdots\right) \quad (4)$$

$$\partial Y_t = X_D \cdot \left(K_{t1} \cdot r^2 + K_{t2} \cdot r^4 + \cdots\right) \quad (5)$$

Because of the physics of optical elements, radial and tangential distortion is proportional to only the even powers of r. The values K_{ri} and K_{ti} are distortion coefficients that must be estimated to calibrate for distortion. As is standard practice to facilitate calibration, tangential distortion is minor thus ignored, and only the first power series component of radial distortion is retained. This results in the following relations between undistorted U and distorted D image coordinates:

$$X_U = X_D \cdot (1 + K \cdot r^2) \quad (6)$$

$$Y_U = Y_D \cdot (1 + K \cdot r^2) \quad (7)$$

The conversion between undistorted coordinates and the real, distorted image coordinates to first-order requires only radial distortion coefficient K. Note that the object area A is not significantly affected by distortion. Distorted image coordinates $[X_D \;\; Y_D \;\; A]$ are in units

of length (e.g., mm). The next step is to transform $[X_D \;\; Y_D \;\; A]$ into digital image or frame coordinates with pixel units. Once passed through the frame grabber, the origin of an image is set to the upper-left corner, with $+x$ horizontal and $+y$ down the image plane. The conversion from distorted image coordinates to frame coordinates involves several parameters. The piercing point $[C_X, C_Y]$ describes the alignment between optical axis and image plane. Ideally, this point would be at the center of the image plane, but in general, the piercing point varies and must be determined experimentally. Next, units conversion from physical length (e.g., mm) to pixels must occur. For the y direction, this term D_Y depends on the number of CCD sels and their physical spacing:

$$D_Y = H_{\mathrm{CCD}}/N_{\mathrm{FY}} \tag{8}$$

where $H_{\mathrm{CCD}} =$ CCD height (mm) and $N_{\mathrm{FY}} =$ number of CCD sels in the y direction. The distorted computer image y coordinate in pixels is then:

$$Y_{\mathrm{FD}} = Y_D/D_Y + C_Y \tag{9}$$

The x conversion differs because the frame grabber samples the video in the x direction. The x-coordinate conversion factor $D_{X'}$ is defined by

$$D'_X = W_{\mathrm{CCD}}/N_{\mathrm{FX}} \cdot N_{\mathrm{FX}}/N_{\mathrm{CX}} \tag{10}$$

where $W_{\mathrm{CCD}} =$ CCD width (mm), $N_{\mathrm{FX}} =$ number of CCD sels along x, and $N_{\mathrm{CX}} =$ number of pixels sampled by the frame grabber along x. The x coordinate of the distorted image is then

$$X_{\mathrm{FD}} = X_D/D'_X + C_X \tag{11}$$

One final scaling parameter s_X, is introduced to account for differences between frame grabber sampling and CCD sel spacing in the x direction. This parameter scales Eq. (11) to become

$$X_{\mathrm{FD}} = s_X \cdot X_D/D'_X + C_X \tag{12}$$

VPS Calibration

A variety of camera calibration techniques has been developed by both machine vision and photogrammetry communities.[21] Each differs in details, but nearly all of the calibration techniques follow the same general procedure to find numerical parameter values. The first calibration input is a series of locations in three-dimensional space. These locations are typically targets on a calibration fixture. Some algorithms use noncoplanar three-dimensional points dispersed throughout a volume. Other more simple algorithms use coplanar calibrations points, at the price of reduced accuracy and capability when compared to noncoplanar techniques.[20–23]

Three-dimensional calibration targets must have two key characteristics. First, two-dimensional image coordinates corresponding to each three-dimensional location must be easily and accurately identifiable. Targets are typically defined as the centers of spheres in noncoplanar calibration fixtures. The classic coplanar calibration pattern is a checkerboard, which has clear, distinct edges and vertices. Second, global target coordinates $[X_G \;\; Y_G \;\; Z_G]^T$ must be known relative to one another. It is convenient to use one target to define the global coordinate frame origin. The second calibration input is the set of two-dimensional image coordinates corresponding to the three-dimensional calibration targets. These two-dimensional coordinates are captured for each camera to be calibrated and for each target not occluded or otherwise indistinguishable.

VPS calibration presented several challenges seldom encountered in other vision systems. Large volumes such as the NBRF require both long measurement distances and large calibration fixtures difficult to manufacture and characterize accurately. Because the cameras are all pointing radially inward, instead of in one general direction, simple, coplanar calibration patterns are not applicable for extrinsic calibration. For VPS, a two-step calibration process was implemented to compute intrinsic and extrinsic parameters separately. To ensure consistent, accurate data, both steps were performed with the cameras mounted in their operational underwater positions. As a first step, multiple images were acquired of a large-scale underwater checkerboard pattern (see Fig. 4) held at 1–2 m from each camera. This local calibration procedure was able to compute camera intrinsic calibration parameters f, C_X, C_Y, and s_X with the accuracy possible in a small calibration volume. Because extrinsic calibration required that all cameras simultaneously view common targets, a large calibration fixture with 20 fixed spherical targets was built (see Fig. 5), each with a known relative position. The two-dimensional image plane and three-dimensional measured target coordinates are inputs to an algorithm that computes extrinsic calibration parameters for all cameras relative to a

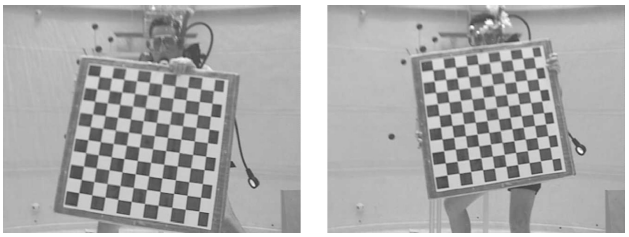
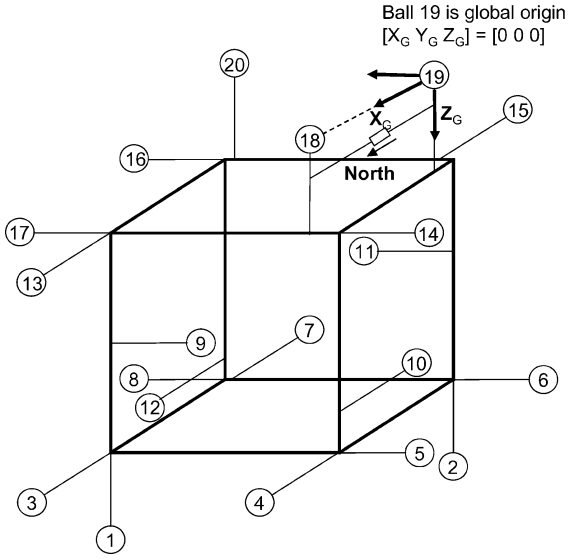
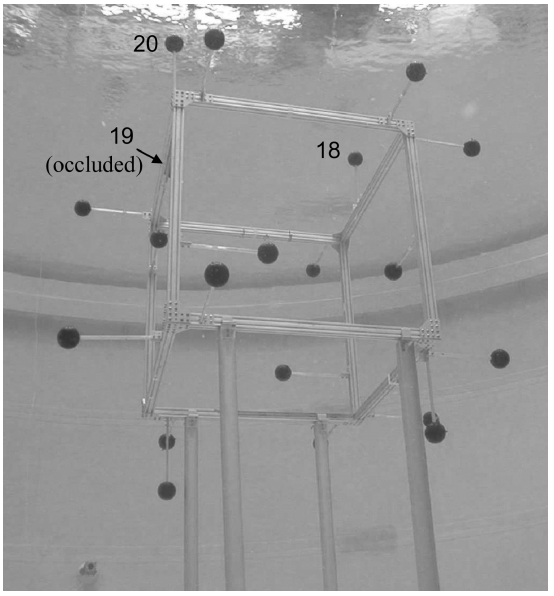


Fig. 4 Sample images from VPS intrinsic calibration.



a) Calibration frame objects and axes



b) Extrinsic calibration frame in the NBRF

Fig. 5 VPS extrinsic calibration frame.

Table 1 VPS intrinsic calibration parameters with error estimates

Camera	f, mm	C_X, pixel	C_Y, pixel	s_X	K
1	6.3301 ± 0.0061	279.19 ± 1.30	225.98 ± 1.30	$1.0251 \pm 1.54E-05$	0.007183
2	7.1428 ± 0.0065	324.20 ± 1.52	217.64 ± 1.42	$1.0268 \pm 1.40E-05$	0.005184
3	7.9902 ± 0.0092	303.38 ± 1.95	303.14 ± 1.85	$1.0274 \pm 1.83E-05$	0.002120
4	6.7967 ± 0.0051	314.00 ± 1.18	325.01 ± 1.20	$1.0265 \pm 1.18E-05$	0.008442
5	5.7769 ± 0.0066	320.03 ± 1.43	230.10 ± 1.53	$1.0266 \pm 1.83E-05$	0.009502
6	6.5814 ± 0.0058	300.19 ± 1.60	216.75 ± 1.51	$1.0257 \pm 1.39E-05$	0.005405
7	7.3099 ± 0.0072	298.95 ± 1.71	253.82 ± 1.72	$1.0255 \pm 1.59E-05$	0.005726
8	6.8419 ± 0.0045	330.62 ± 1.22	242.48 ± 1.12	$1.0264 \pm 1.04E-05$	0.005833

global reference frame origin. Although the algorithm used in the extrinsic calibration can calculate intrinsic and extrinsic parameters, greater accuracy is possible with the separate intrinsic calibration that has more data viewed at closer range. For VPS calibration, intrinsic calibration was performed first, then the extrinsic calibration algorithm was executed to initially compute all 11 intrinsic and extrinsic parameters based on the three-dimensional calibration frame. Intrinsic parameters are replaced with more accurate values from the original intrinsic calibration, then the extrinsic algorithm was reexecuted to yield the most accurate calibration possible.

Intrinsic Calibration

VPS cameras were intrinsically calibrated using a MATLAB™ calibration toolbox implementation[§] of a checkerboard-based algorithm described by Zhang.[23] Table 1 summarizes results for the eight VPS cameras, including upper-bound error estimates. As expected, manually adjusted wide-angle focal lengths are between 5–10 mm. C_X and C_Y are not far from their ideal values of 320 and 240 pixels, given the 640×480 image plane. The s_X and K values are also close to their ideal values of one and zero, respectively. The camera model used in the intrinsic parameter estimation software uses representations of focal length and distortion that differ from those used by VPS and the extrinsic calibration software[¶] that implements the Tsai algorithm.[20] A simple calculation was able to convert between focal length representations. However, radial distortion could not be equated, requiring the recomputation of K during extrinsic calibration. Thus, intrinsic calibration provided values for f, C_X, C_Y, and s_X, while the target-based Tsai algorithm estimated all extrinsic parameters plus K.

The errors reported in Table 1 were more significant after the first calibration pass. The first corrective action was to decrease uncertainty by increasing the number of images over which parameters were statistically computed to approximately 40 images per camera. Next, using built-in error analysis tools, specific images that produced large error were eliminated. Finally, checkerboard vertices that had been imprecisely marked were identified and eliminated. After these steps, accuracy goals of 0.01 mm for focal length and two pixels for C_X and C_Y were achieved. This appears to near the limit for VPS equipment and configuration and is sufficient to meet our centimeter-level positioning accuracy goal.

Extrinsic Calibration

A C implementation[**] of the Tsai method[20] was used to estimate VPS camera intrinsic and extrinsic calibration parameters simultaneously, with modification to incorporate the four intrinsic parameters f, C_X, C_Y, s_X computed with the checkerboard-based algorithm. The extrinsic parameter calibration target fixture had several requirements:

1) A minimum of 14 highly visible three-dimensional targets must be visible for each camera.

2) Target (x, y) positions must be accurately computed in all images, requiring a fixture that minimized target occlusion.

3) Physical target locations $[X_G \;\; Y_G \;\; Z_G]^T$ must be accurately characterized.

4) Targets must be rigidly mounted to guarantee measurements are always consistent.

Positions must be an order of magnitude more precise than the accuracy required for the vision task, thus millimeter-level measurement precision was required. Additionally, calibration accuracy improves as the calibration volume increases to occupy a large fraction of the image plane. To enable correlation of the calibration frame axes with the vehicle IMU, the calibration frame was affixed such that North is aligned with one axis (e.g., x) and down is aligned with another (e.g., z). Figure 5 shows the calibration fixture. The frame is a 127-cm aluminum cube to which twenty 36-cm aluminum posts are affixed. Hollow black plastic spheres are attached to the end of each post and act as calibration targets. The use of 20 targets guarantees that each camera will have an unoccluded view of more than the minimum of 14 targets. In the NBRF, the calibration frame sits on top of long stilts to be in view of all cameras. Ball 19 is defined as the global reference frame origin, and the vector from ball 19 to ball 18 defines the global x axis. A compass is affixed between these balls to align the frame with magnetic North. Figure 5 illustrates ball locations and axis definitions.

To use the calibration fixture, relative target positions must be precisely characterized. As a feasible alternative to measuring target positions in all three axes, ball-to-ball distances were measured and served as the numerical inputs to the calibration parameter optimizer. Special large-scale calipers were constructed that measured interball distances accurately to ± 2 mm. With n targets, the potential number of target-to-target measurements m is defined by

$$m = (n^2 - n)/2 \tag{13}$$

Thus, for 20 targets there are 190 potential measurements. Because the caliper was occasionally obstructed by the structure of the frame, only 180 measurements could actually be taken. With 20 targets, there were 60 unknowns to be solved, the x, y, and z coordinates for each target. The 180 measurements each created one term in a scalar objective function. If $L_{j,k}$ is the measurement from ball j to ball k and $[X_j \;\; Y_j \;\; Z_j]^T$ and $[X_k \;\; Y_k \;\; Z_k]^T$ were the global coordinates of balls j and k, then the objective function equation for that measurement is

$$e_{jk} = \left\{ \left[(X_j - X_k)^2 + (Y_j - Y_k)^2 + (Z_j - Z_k)^2 \right]^2 - L_{j,k}^2 \right\}^2 \tag{14}$$

The scalar objective function is then the sum of all errors:

$$E = \sum_{j=1}^{20} \sum_{k=1}^{20} e_{jk} \tag{15}$$

Because the distance from any ball to itself is zero, $e_{jk} = 0$ if $j = k$. A minimization with a 180-term scalar objective function in 60 unknowns will have a multitude of local minima, making the solution quite sensitive to initial conditions. Accurate initial guesses of the target locations were computed deterministically. The plane defined by the top three targets, balls 18 thru 20, was defined as the x–y plane with $z = 0$. Ball 19 is the origin, and ball 18 is defined to be on the x axis. Ball 20 is assigned coordinates

[§]Data available online at http://www.vision.caltech.edu/bouguetj/calib_doc/ [2004].

[¶]Data available online at http://www-2.cs.cmu.edu/~rgw/TsaiCode.html [2004].

[**]Data available online at http://www-2.cs.cmu.edu/~rgw/TsaiCode.html [2004].

Table 2 VPS extrinsic calibration parameters with error estimates

Camera	R_X, deg	R_Y, deg	R_Z, deg	T_X, mm	T_Y, mm	T_Z, mm	Normalized pixel error, pixels	Avg. object space error, mm	Max. object space error, mm	No. targets visible
1	94.72	−80.21	175.40	−614.66	−1208.47	6921.82	0.60	2.12	4.78	17
2	91.12	7.65	179.89	283.17	−1245.12	6792.34	0.64	1.91	5.97	18
3	−83.61	83.70	5.86	565.41	−1500.57	7413.26	1.18	3.03	8.59	18
4	−90.82	−9.15	0.08	−489.02	−1229.02	7555.31	0.76	2.32	4.49	18
5	−54.55	−61.39	−34.30	−258.17	−223.93	7391.42	0.73	3.10	6.61	19
6	66.50	−22.27	−169.92	341.59	−831.95	6914.25	1.07	3.92	11.21	20
7	41.41	56.24	132.38	588.11	−1468.69	7206.47	0.76	2.31	7.43	20
8	−68.66	20.05	6.65	−222.95	−949.48	7689.74	0.70	2.29	6.40	20

$[x_{20}\ \ y_{20}\ \ 0]^T$. The three-dimensional positions of targets 18 thru 20 can then be solved deterministically with three intertarget distances $L_{18,19}$, $L_{18,20}$, and $L_{19,20}$. Ball 19 has coordinates $[0,\ 0,\ 0]^T$, and ball 18 is at $[L_{18,19}\ \ 0\ \ 0]^T = [x_{18}\ \ 0\ \ 0]^T$. Ball 20 coordinates are then given by

$$L_{19,20}^2 = x_{20}^2 + y_{20}^2 \tag{16}$$

$$L_{18,20}^2 = y_{20}^2 + (x_{18} - x_{20})^2 \tag{17}$$

with x_{20} and y_{20} the only unknowns. The remaining 17 targets have positive z coordinates. The position of any ball k can be computed with three measurements between it and targets 18–20:

$$L_{k,18}^2 = (x_k - x_{18})^2 + y_k^2 + z_k^2 \tag{18}$$

$$L_{k,19}^2 = x_k^2 + y_k^2 + z_k^2 \tag{19}$$

$$L_{k,20}^2 = (x_k - x_{20})^2 + (y_K - y_{20})^2 + z_k^2 \tag{20}$$

This system was solved algebraically for each of the 17 target positions $[x_k\ \ y_k\ \ z_k]^T$. Using the resulting three-dimensional positions as initial conditions, a Nelder–Mead simplex optimization algorithm is then executed to adjust position estimates and find the best match for physical measurements. With the optimized three-dimensional measurements, "synthetic" distances $L_{j,k}$ were calculated:

$$L_{j,k} = \sqrt{(x_j - x_k)^2 + (y_j - y_k)^2 + (z_j - z_k)^2} \tag{21}$$

Synthetic and physical distances were then compared. Those few that exhibit large differences are discarded, and the optimization process is repeated until all measurement differences were less than 1.0 mm. Even though discarded measurements were not unreasonable (1–3 mm average difference; 5 mm maximum difference), 30 of the 180 measurements spread over most of the 20 targets were discarded. Then, presuming two-dimensional image coordinate data are of similar quality, extrinsic calibration can be computed to millimeter-level accuracy, providing sufficient calibration quality for centimeter-level VPS system accuracy.

A MATLAB® image processing program was created to facilitate target localization in each camera's two-dimensional image plane. A user identifies each target ID and graphically defines a circle to circumscribe each image plane target. This target localization strategy is somewhat time consuming but has proven quite repeatable and is estimated to have an accuracy of ∼0.3 pixels. The Tsai algorithm first computes a full set of calibration parameters based on the three-dimensional and two-dimensional calibration target locations. Because of the relatively few targets (20) and the long distances involved, this initial calibration is not highly accurate. Next, values of C_X, C_Y, f, and s_X computed from the checkerboard algorithm are specified as fixed calibration parameters for the Tsai code, which is rerun with first-iteration values for extrinsic parameters (and K) used as initial guesses.

The Tsai algorithm is based on the set of homogeneous transformation equations that relate the three-dimensional target positions with their two-dimensional image plane coordinates, including translational and rotational extrinsic and intrinsic parameters. The error (cost) function to minimize is defined as

$$J = \sum_{i=1}^{n} (x_i - x_i')^2 + \sum_{i=1}^{n} (y_i - y_i')^2 \tag{22}$$

where n = number of calibration targets used, (x_i, y_i) = observed image coordinates of target i, and $(x_i, y_i)'$ = predicted image coordinates for target i based on current calibration parameter estimates. The Levenberg–Marquardt optimization method is used, requiring a nonredundant representation for rotation, in this case, XYZ fixed Euler angles (R_X, R_Y, R_Z). Table 2 lists the extrinsic calibration results, with rotations and translations to the global frame reported in local camera coordinates. Table 2 also contains the extrinsic calibration errors and the number of usable calibration targets per camera. The Tsai algorithm computes overall process errors rather than the error for each parameter estimate. The normalized pixel error is a measure of pixel-space errors that accumulate over the optimization process, providing a measure of relative accuracy for different cameras. The average object space error (in mm) is the average difference between actual three-dimensional calibration target locations and their predicted three-dimensional global locations given the final calibration parameters. The maximum object space error is the largest discrepancy between actual and predicted target locations. These error metrics include inaccuracies of extrinsic and intrinsic calibration parameters because all are used in the camera to global coordinate transformation. In practice, the error introduced by calibration inaccuracies alone should not exceed the maximum object space error (4–11 mm). As a conservative interpretation, calibration error is also unlikely to be smaller than the average object space error (2–4 mm).

State Estimation with an Extended Kalman Filter

The goal of VPS is to provide accurate position and velocity estimates for free-flying targets such as SCAMP from the sequence of images captured by the cameras. The full state vector is $\boldsymbol{x} \equiv [X_G\ \ \dot{X}_G\ \ Y_G\ \ \dot{Y}_G\ \ Z_G\ \ \dot{Z}_G]^T$, where X_G, Y_G, and Z_G are the position coordinates of the target with respect to the fixed VPS frame and the overdots denote the corresponding velocities.

In principle, each captured image is capable of providing a full, three-dimensional global position estimate $[X_G\ \ Y_G\ \ Z_G]$ of the target; the velocity estimates could then be obtained by finite differencing the resulting sequence. However, three-dimensional position estimates from a single image require computing the distance to the camera based upon the image area, a process that is highly sensitive to measurement noise. Preliminary computations using the calibration coefficients computed in the preceding section showed that area errors of only a few percent could create position errors on the order of tens of centimeters, which would be unacceptable.

Thus, instead VPS was designed to use only the undistorted digital image plane coordinates of the target centroid $(X_{\mathrm{FU}}\ \ Y_{\mathrm{FU}})$ measured in pixels. These are related to the global position $(X_G\ \ Y_G\ \ Z_G)$, measured in meters, using the camera calibration constants determined in the previous section with r_{ij} representing elements of the

rotation matrix from camera to global G coordinates:

$$X_{\mathrm{FU}} = \left[\frac{fT_X s_X}{D'_X} + C_X T_Z + \left(C_X r_{31} + \frac{f r_{11} s_X}{D'_X} \right) X_G + \left(C_X r_{32} + \frac{f r_{12} s_X}{D'_X} \right) Y_G + \left(C_X r_{33} + \frac{f r_{13} s_X}{D'_X} \right) Z_G \right] \Big/ (T_Z + r_{31} X_G + r_{32} Y_G + r_{33} Z_G) \tag{23}$$

$$Y_{\mathrm{FU}} = \left[\frac{fT_Y}{D_Y} + C_Y T_Z + \left(C_Y r_{31} + \frac{f r_{21}}{D_Y} \right) X_G + \left(C_Y r_{32} + \frac{f r_{22}}{D_Y} \right) Y_G + \left(C_Y r_{33} + \frac{f r_{23}}{D_Y} \right) Z_G \right] \Big/ (T_Z + r_{31} X_G + r_{32} Y_G + r_{33} Z_G) \tag{24}$$

Because there are three position coordinates and only two pieces of information from each image, two or more image plane centroid measurements must be combined in order to deduce the position of the vehicle. Combining measurements is complicated by the fact that individual measurements can be taken at slightly different times, and that there can be slight discrepancies between cameras because of noise in image acquisition and centroid calculation. To circumvent these difficulties, VPS uses an extended Kalman filter to form the global position estimate. By using a model of the target dynamics, the same filter can also provide significantly more accurate velocity estimates than a naïve finite differencing of positions would yield.

The state dynamics of a translating flight vehicle are given by the differential equation

$$m \ddot{X}_G = \sum_i F_{\mathrm{ext},x,i} \tag{25}$$

where m is the vehicle mass (approx 50 kg for SCAMP), and $F_{\mathrm{ext},x,i}$ are the external forces applied to the vehicle acting in X direction of the VPS frame. Analogous expressions can be obtained for Y_G and Z_G axis dynamics. For the underwater vehicle SCAMP, the principal forces are F_{thrust}, the force applied by the vehicle thrusters, and F_{drag}, the force caused by viscous water drag along this same axis. SCAMP is balanced to be neutrally buoyant, so that its natural buoyancy approximately balances the effect of gravity; hence, these two external forces cancel and are neglected in the preceding model. The drag terms are typically quadratic in velocity, making differential equation (25) nonlinear. For the limited range of velocities SCAMP can achieve, however, a linear drag model of the form $F_{\mathrm{drag},x} = -C_{\mathrm{DT}} \dot{X}_G$, where C_{DT} is a constant drag coefficient, is sufficiently accurate and allows Eq. (25) to be analyzed as a linear differential equation. The drag constant C_{DT} was experimentally computed as 90 N/m/s for SCAMP.

The complete state vector dynamics can then be written

$$\dot{\boldsymbol{x}}(t) = A\boldsymbol{x}(t) + B\boldsymbol{u}(t) \tag{26}$$

where $\boldsymbol{u} = [F_{\mathrm{thurst},x} F_{\mathrm{thrust},y} F_{\mathrm{thrust},z}]^T$ and

$$A \equiv \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -C_{\mathrm{DT}}/m & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -C_{\mathrm{DT}}/m & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & -C_{\mathrm{DT}}/m \end{bmatrix}$$

$$B \equiv \begin{bmatrix} 0 & 0 & 0 \\ 1/m & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1/m & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1/m \end{bmatrix} \tag{27}$$

The corresponding sample/hold equivalent is $\boldsymbol{x}_k = \Phi_{k-1}\boldsymbol{x}_{k-1} + \Lambda_{k-1}\boldsymbol{u}_{k-1}$, where $\boldsymbol{x}_k$ denotes the value of the state vector at time $t = t_0 + kT_s$, with T_s the sample interval, and the governing transition matrices are given by Franklin and Powell.[24]

$$\Phi_{k-1} \equiv \begin{bmatrix} 1 & a & 0 & 0 & 0 & 0 \\ 0 & b & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & a & 0 & 0 \\ 0 & 0 & 0 & b & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & a \\ 0 & 0 & 0 & 0 & 0 & b \end{bmatrix} \qquad \Lambda_{k-1} = \begin{bmatrix} c & 0 & 0 \\ d & 0 & 0 \\ 0 & c & 0 \\ 0 & d & 0 \\ 0 & 0 & c \\ 0 & 0 & d \end{bmatrix} \tag{28}$$

where

$$a = (m/C_{\mathrm{DT}})\left(1 - e^{-C_{\mathrm{DT}} T_S/m}\right), \qquad b = e^{-C_{\mathrm{DT}} T_S/m}$$

$$c = \left[m\left(e^{-C_{\mathrm{DT}} T_S/m} - 1\right) + C_{\mathrm{DT}} T_s \right] \Big/ C_{\mathrm{DT}}^2$$

$$\text{and} \qquad d = (1/C_{\mathrm{DT}})\left(1 - e^{-C_{\mathrm{DT}} T_S/m}\right)$$

Finally the measurement model, which relates the values of the state variables at the kth sample time to the expected centroid coordinates at that time, is $z_k = h(\boldsymbol{x}_k)$ where

$$h(\boldsymbol{x}) \equiv \begin{bmatrix} X_{\mathrm{FU}} \\ Y_{\mathrm{FU}} \end{bmatrix} \tag{29}$$

and the dependence of $(X_{\mathrm{FU}}\ Y_{\mathrm{FU}})$ on position components $(X_G\ Y_G\ Z_G)$ of $\boldsymbol{x}$ are given in Eqs. (23) and (24). Note that the calibration coefficients used in this calculation at each sample time will depend upon the specific camera that takes the measurement.

To generate optimal estimates $\hat{\boldsymbol{x}}_{k-1}$ from available measurements z_k, the Kalman filter predictor/corrector structure is[25]

Predict:

$$\hat{\boldsymbol{x}}_k^- = \Phi_{k-1}\hat{\boldsymbol{x}}_{k-1} + \Lambda_{k-1}\boldsymbol{u}_{k-1}$$

$$P_k^- = \Phi_{k-1} P_{k-1} \Phi_{k-1}^T + Q_{k-1} \tag{30}$$

Correct:

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k^- + K_k[z_k - h(\hat{\boldsymbol{x}}_k^-)], \qquad P_k = (I - K_k H_k) P_k^- \tag{31}$$

where the Kalman gain matrix K is computed from

$$K_k = P_k^- H_k^T \left(H_k P_k^- H_k^T + R_k \right)^{-1} \tag{32}$$

The algorithm is initialized with $\hat{\boldsymbol{x}}_0$ representing the estimated initial position of the vehicle when the VPS system is started, and $P_0 = E[\hat{\boldsymbol{x}}_0\ \hat{\boldsymbol{x}}_0^T]$ is the covariance matrix representing the confidence in the initial estimate. The matrix H_k, which appears in the Kalman-filter equations, is the linearization of the measurement model $h(\boldsymbol{x})$ about the current estimate $\hat{\boldsymbol{x}}$. That is,

$$H_k = \left. \frac{\partial h(\boldsymbol{x})}{\partial \boldsymbol{x}} \right|_{x = \hat{x}_k} \tag{33}$$

Taking the indicated partial derivatives

$$\frac{\partial h(\boldsymbol{x})}{\partial \boldsymbol{x}} = \begin{bmatrix} H_{11} & 0 & H_{13} & 0 & H_{15} & 0 \\ H_{21} & 0 & H_{23} & 0 & H_{25} & 0 \end{bmatrix} \tag{34}$$

where

$$H_{11} = \frac{\partial X_{\mathrm{FU}}}{\partial X_G} = \frac{f \cdot s_X \cdot [-r_{31} \cdot (T_X + r_{12} \cdot Y_G + r_{13} \cdot Z_G) + r_{11} \cdot (T_Z + r_{32} \cdot Y_G + r_{33} \cdot Z_G)]}{D'_X \cdot (T_Z + r_{31} \cdot X_G + r_{32} \cdot Y_G + r_{33} \cdot Z_G)^2}$$

$$H_{13} = \frac{\partial X_{\mathrm{FU}}}{\partial Y_G} = \frac{f \cdot s_X \cdot [-r_{32} \cdot (T_X + r_{11} \cdot X_G + r_{13} \cdot Z_G) + r_{12} \cdot (T_Z + r_{31} \cdot X_G + r_{33} \cdot Z_G)]}{D'_X \cdot (T_Z + r_{31} \cdot X_G + r_{32} \cdot Y_G + r_{33} \cdot Z_G)^2}$$

$$H_{15} = \frac{\partial X_{\mathrm{FU}}}{\partial Z_G} = \frac{f \cdot s_X \cdot [-r_{33} \cdot (T_X + r_{11} \cdot X_G + r_{12} \cdot Y_G) + r_{13} \cdot (T_Z + r_{31} \cdot X_G + r_{32} \cdot Y_G)]}{D'_X \cdot (T_Z + r_{31} \cdot X_G + r_{32} \cdot Y_G + r_{33} \cdot Z_G)^2}$$

$$H_{21} = \frac{\partial Y_{\mathrm{FU}}}{\partial X_G} = \frac{f \cdot [-r_{31} \cdot (T_Y + r_{22} \cdot Y_G + r_{23} \cdot Z_G) + r_{21} \cdot (T_Z + r_{32} \cdot Y_G + r_{33} \cdot Z_G)]}{D_Y \cdot (T_Z + r_{31} \cdot X_G + r_{32} \cdot Y_G + r_{33} \cdot Z_G)^2}$$

$$H_{23} = \frac{\partial Y_{\mathrm{FU}}}{\partial Y_G} = \frac{f \cdot [-r_{32} \cdot (T_Y + r_{21} \cdot X_G + r_{23} \cdot Z_G) + r_{22} \cdot (T_Z + r_{31} \cdot X_G + r_{33} \cdot Z_G)]}{D_Y \cdot (T_Z + r_{31} \cdot X_G + r_{32} \cdot Y_G + r_{33} \cdot Z_G)^2}$$

$$H_{25} = \frac{\partial Y_{\mathrm{FU}}}{\partial Z_G} = \frac{f \cdot [-r_{33} \cdot (T_Y + r_{21} \cdot X_G + r_{22} \cdot Y_G) + r_{23} \cdot (T_Z + r_{31} \cdot X_G + r_{32} \cdot Y_G)]}{D_Y \cdot (T_Z + r_{31} \cdot X_G + r_{32} \cdot Y_G + r_{33} \cdot Z_G)^2}$$

and, again, the specific calibration parameters (f, s_X, T_X, etc.) used in computing the full matrix H_k from these equations depend upon the camera that takes the measurement at the kth sample time.

The matrices R_k and Q_k in Eqs. (30) and (32) are covariance matrices quantifying the level of noise in the measurement process and state dynamics, respectively. The measurement noise can be determined empirically from sequences of single camera images of a static target. Using a sequence of 250 such images, a value of

$$R_k = \begin{bmatrix} 4 & 0.01 \\ 0.01 & 4 \end{bmatrix} \tag{35}$$

was determined for the VPS cameras, corresponding roughly to worst-case variation of ±5–6 pixels in measurement of X_{FU} Y_{FU}, with a slight cross correlation.

The dominant source of stochastic influence on the dynamics comes from unmodeled forces that act on the vehicle. Accordingly, Q_{k-1} can be taken as

$$Q_{k-1} = \left(\Lambda_{k-1} \cdot \Lambda_{k-1}^T\right) \cdot \sigma_F^2 \tag{36}$$

where σ_F is the anticipated standard deviation of the forces that act on the vehicle, in Newtons. Based upon previous experience with SCAMP, this parameter was chosen as $\sigma_F = 2$N for the experiments reported next.

Software Implementation

As shown in Fig. 3, VPS software is distributed among a control station and four computers with frame grabbers (Vision1-Vision4). Each dedicated vision computer grabs images from two cameras and computes target centroid and area, provided the target is within the camera's FOV. The control station computer then compiles measurements and telemetry data from SCAMP into an EKF-based state estimate.

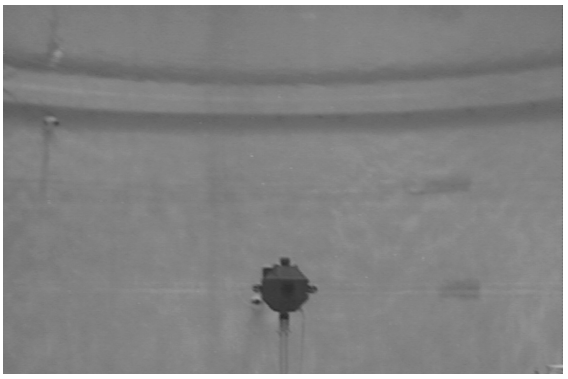
Upon startup, the vision computers acquire reference background images from all cameras and read the camera calibration parameters. Background images are acquired without SCAMP or other moving objects in the neutral buoyancy tank. Before each frame capture, the global EKF state estimate received from the control station is evaluated. If the estimated vehicle position is within the camera FOV, the frame grabber captures an image and attaches a time stamp. The image is then processed, and new measurement data are sent to the control station computer. If the tracked object is out of a camera's FOV, that camera is skipped because its data will be of no use to the EKF. Because we do not use area information in our state estimates, each VPS camera provides information for only two translational DOF. So long as a minimum of two cameras have SCAMP in their FOV, the EKF receives sufficient data to update its full translational state estimate. If the tracked object is outside the FOV of all (or all but one) VPS cameras, which can happen as shown in Fig. 2, it is possible for the global state estimate to diverge. In practice, however, given the small SCAMP size and large neutral buoyancy tank volume, there has been little challenge with commanding trajectories that keep SCAMP well within view of two or more cameras.

Figure 6 shows an example image over the three basic processing steps required to compute vehicle centroid and area. Because of the cluttered background in the NBRF, the first image processing step is to subtract the background image from the current frame (e.g., Fig. 6a), resulting in a new image composed of pixel values significantly greater than zero (e.g., light grey/white) only where the two images differ (e.g., Fig. 6b). Currently, this process is performed over grayscale pixel values (0–255), but the same static subtraction algorithm would also function with red-green-blue (RGB) frames to distinguish objects from background by color. Next, the difference image (Fig. 6b) is thresholded to create a binary image. The pixels set to black (0) are labeled part of the tracked object, and all other white pixels (255) are designated as background.

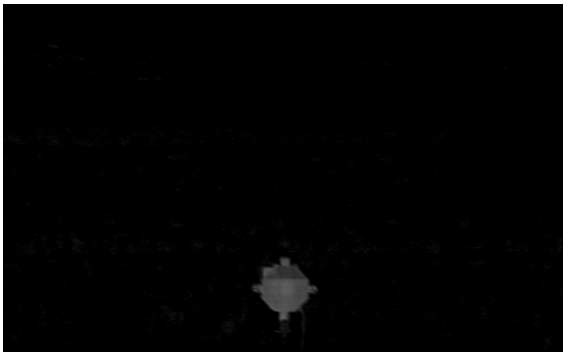
Ideally, the binary image created after thresholding would only contain one contiguous region of black pixels corresponding to the target (SCAMP). Experience shows that in addition to this region, there will be scattered black pixels known as "salt-and-pepper" noise. A linear filter is applied to remove this noise, with filter parameters ρ, threshold, and λ, region size. Values appropriate for VPS were determined experimentally. Figure 6c shows the example thresholded image after noise has been removed.

Tracked object centroid and area are computed next. Area is computed as the total number of black pixels, whereas X and Y centroid coordinates are the average black pixel distances from the image plane X and Y axes. These "raw" centroid coordinates are $[X_{\mathrm{FD}}\ Y_{\mathrm{FD}}]$, the distorted image coordinates. $[X_{\mathrm{FD}}\ Y_{\mathrm{FD}}]$ is converted into distorted real image coordinates, $[X_D\ Y_D]$ from Eqs. (12) and (9), respectively, then undistorted to form $[X_U\ Y_U]$ as described by Eqs. (6) and (7). Real undistorted coordinates $[X_U\ Y_U]$ are then converted back into undistorted digital image coordinates $[X_{\mathrm{FU}}\ Y_{\mathrm{FU}}]$, again using Eqs. (12) and (9) now applied to the undistorted (U) coordinates rather than distorted (D) coordinates. The undistorted digital centroid $[X_{\mathrm{FU}}\ Y_{\mathrm{FU}}]$ and image area are transmitted to the control station computer in local camera coordinates so the EKF can distinguish the relatively accurate centroid measurements from the less accurate area data.

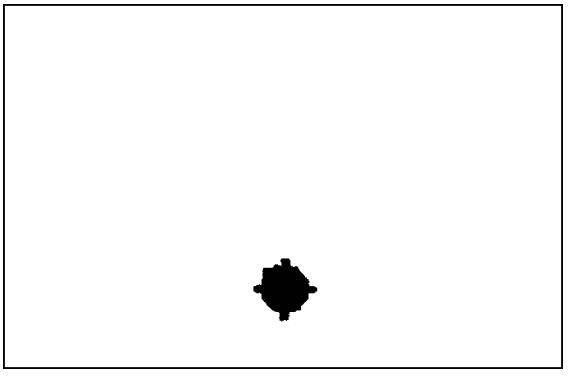
To ensure timing accuracy, the clocks on all four vision computers, the control station computer, and the SCAMP flight computer are synchronized prior to each test. Although the image processing software can be applied to any "tracked object," the EKF requires vehicle telemetry and a dynamic model, so that the control station VPS software is customized for SCAMP. The control station software is connected to SCAMP and vision computers by an Ethernet (UDP) link. Each time the control station receives new measurement data from a vision computer, it returns the current inertial position



a) Initial VPS image



b) Image after background subtraction



c) Image after thresholding and noise removal

Fig. 6 VPS image processing stages.

and velocity state vector $\hat{\boldsymbol{x}}_{\text{VPS}}$. In the standard closed-loop attitude control mode, the control station downlinks pilot hand controller inputs ${}^{B}\boldsymbol{F}_{\text{DES}}$ (desired body forces) and ${}^{B}\boldsymbol{\omega}_{\text{DES}}$ (desired body angular velocities) to SCAMP, as well as VPS state estimate $\hat{\boldsymbol{x}}_{\text{VPS}}$. The control station logs all SCAMP and VPS data to disk and manages all user commands.

Run as a control station thread, the EKF is executed at a constant 10-Hz rate, enabling Φ_{k-1} and Λ_{k-1} to be constant matrices. SCAMP uplinks the attitude quaternion, as well as the commanded body forces, to the control station as telemetry. At each iteration, the EKF converts the current SCAMP quaternion into a rotation matrix ${}^{G}_{B}R$ from body frame to inertial (global) frame. The EKF uses it to rotate the body forces from body to inertial frame ($\boldsymbol{F}^{G} = {}^{G}_{B}R \cdot \boldsymbol{F}^{B}$). The EKF propagates its state estimate $\hat{X}$ and error covariance matrix P one time step given inertial forces $\boldsymbol{F}^{G}$ from the previous time step and the SCAMP dynamic model. Next, it reads camera measurement data that have arrived since the last EKF estimation cycle. Starting at its value after the propagation step, $\hat{X}$ is repeatedly corrected, once for each valid measurement. For each new measurement processed, values of intermediate variables (K_k, P, etc.) are also updated. If there are no new camera measurements for this EKF iteration, the new state estimate will be based only on dynamic model propagation given the applied control force vector. The resulting translational state estimate $\hat{X}_{\text{VPS}} = \hat{X}$ is shared with the main control station process, and the EKF thread transitions to an idle state until the next timer event signals a new state estimate is required.

System Evaluation

A series of static and dynamic tests was conducted in the neutral buoyancy tank to determine VPS state estimate accuracies. For static tests, both a water-filled ball and the SCAMP vehicle were placed at fixed locations on in the tank, and VPS position estimates were compared to truth measurements (when available) and independent VPS measurements otherwise. For dynamic tests, VPS cameras were partitioned into groups; and the EKF was run independently on distinct camera measurement sets. Dynamic EKF estimates based on the full camera set were also computed. Note that only cameras 1–6 were available for testing because of hardware (camera and frame grabber availability) issues; use of the two additional cameras would only improve EKF results.

Static EKF Position Estimates

To initially determine static positioning accuracy, a water-filled ball covered in black fabric was suspended in the tank, providing a quick test setup that did not require vehicle support personnel. Because force inputs to the EKF were zero, the EKF computed ball position with only camera measurements, effectively using the EKF as an iterative least-squares algorithm. The six operational VPS cameras were separated into two sets: $EKF1 = \{C1, C2, C5\}$ and $EKF2 = \{C3, C4, C6\}$. This grouping was chosen to ensure each EKF could compute all three translational DOF given camera mounting locations sketched in Fig. 2. For each EKF, Table 3 lists average and standard deviation for three static positions ($1b$, $2b$, $3b$) computed over 10 s of data (100+ images per camera). $|\boldsymbol{E}|$ represents error vector magnitude between estimates from EKF1 vs EKF2. For this task, the two EKF estimates agree to within ~3 cm, providing an initial indication of system accuracy when the EKF receives data from only a few (2–3) cameras. Note that comparison of the two EKF outputs does not guarantee absolute accuracy because a systematic error could cause both EKFs to be incorrect. However, this comparison does show position estimation is repeatable and that calibrated camera measurements are consistent.

For the next static test set, SCAMP was affixed to a rigid fixture with known geometry, providing truth measurements independent of VPS. Table 4 shows estimated SCAMP (X, Y, Z) coordinates

Table 3 EKF static estimates for a suspended ball

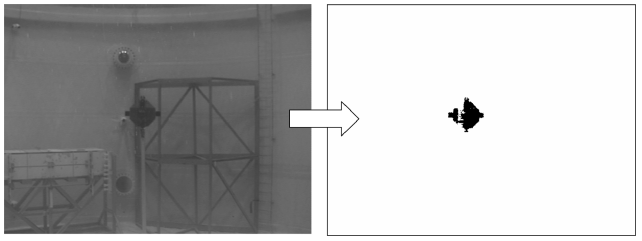
Data set	Position	$1b$	$2b$	$3b$
3-cam EKF1	$\boldsymbol{x}$, m	−0.1531	−0.7217	0.4963
	σ_X	0.019	028	0.007
	$\boldsymbol{y}$, m	0.9387	0.8007	1.0611
	σ_Y	0.008	0.009	0.016
	$\boldsymbol{z}$, m	−0.4689	−0.0634	0.475
	σ_Z	0.016	0.014	0.033
3-cam EKF2	$\boldsymbol{x}$, m	−0.1725	−0.7027	0.4734
	σ_X	0.004	0.011	0.005
	$\boldsymbol{y}$, m	0.9412	0.8196	1.072
	σ_Y	0.012	0.021	0.087
	$\boldsymbol{z}$, m	−0.4621	−0.0446	0.474
	σ_Z	0.012	0.026	0.065
Relative error	$\|\boldsymbol{E}\|$, m	**0.0207**	**0.0327**	**0.0254**

Table 4 Static SCAMP data: position estimates

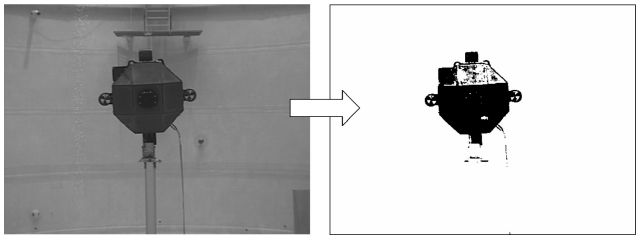
Position	$1s$	$2s$	$3s$
$\boldsymbol{x}$, m	−0.2644	0.9155	1.111
σ_X	0.022	0.004	0.053
$\boldsymbol{y}$, m	0.6944	0.8805	0.0682
σ_Y	0.060	0.003	0.008
$\boldsymbol{z}$, m	1.2113	1.2119	1.2127
σ_Z	0.104	0.006	0.058

Table 5 Static SCAMP data measured vs estimated ranges

Distance	Actual distance, m	VPS distance, m	Difference, m
1*s* to 2*s*	1.054	1.194	0.140
2*s* to 3*s*	0.819	0.836	0.017
1*s* to 3*s*	1.335	1.33	−0.005



a) Image with background interference



b) Image with glare on top vehicle panels

Fig. 7 VPS measurement error sources.

and their standard deviations for each of three static mounting positions (1*s*, 2*s*, 3*s*), as calculated by a single EKF with all six camera inputs. Although exact inertial coordinates for the static positions are not known, distances between fixture mounting locations are measurable. Table 5 compares measured (truth) and EKF-based distances between static positions (1*s*, 2*s*, 3*s*). The distances (2*s*-3*s*) and (1*s*-3*s*) agree to within 2 cm, providing confidence that the EKF is functioning properly. However, EKF distance (1*s*-2*s*) differs by 14 cm, illustrating one of the primary sources of error. Current image processing algorithms presume background color/intensity differs from that of the vehicle. Figure 7 shows two sources of error: a background truss with similar pixel value to SCAMP (Fig. 7a) and bright sunlight glare on the top SCAMP panels that make the vehicle appear too similar to its lighter background. In both cases, SCAMP pixels cannot be distinguished from the background resulting in their misclassification by VPS. These errors cause image centroid estimates to shift laterally or vertically away from misclassified regions. For example, Fig. 7a is an image acquired by camera 1 for static position 2*s*. The centroid of SCAMP is offset to the right, biasing the camera 1 X_{FU} coordinate for the 2*s* EKF estimate such that distance (1*s*-2*s*) sees 14-cm error, whereas distance (2*s*-3*s*) is mostly unaffected because it has is no significant X_{FU} component. Although the EKF mitigates single-camera errors to some extent (e.g., 14-cm error is reduced by other camera measurements), work is in progress to better characterize image plane centroid with a real-time geometry-based frequency-domain method, which is expected to correct at least for the error sources illustrated in Fig. 7.

Dynamic EKF State Estimation

For dynamic tests, SCAMP was piloted in a smooth, continuous trajectory within the neutral buoyancy tank region where most of the six operational cameras had near-continuous coverage (as shown in Fig. 3). Such coverage enabled the splitting of camera data into subsets over the same trajectory to compare independent dynamic EKF state estimates. Figure 8 shows an example trajectory, specifically the inertial $\boldsymbol{x}$ position and velocity coordinates as a function of time. And 3σ error bounds computed from EKF covariance matrix P are shown, depicting the region in which the actual state exists with 99% probability. The position varies smoothly over time, and velocities are consistent with position estimates. As is typically the

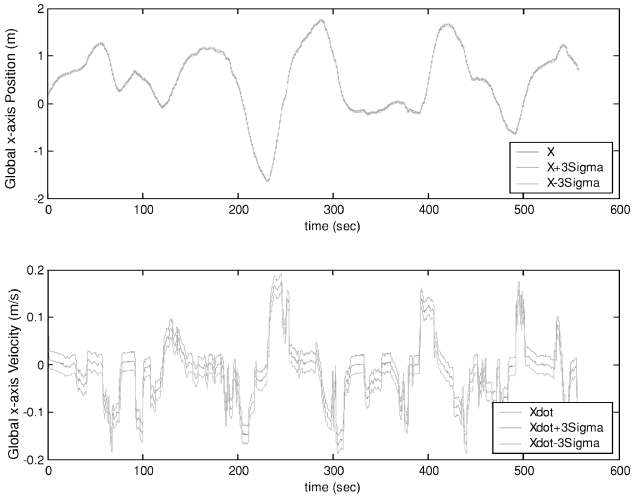
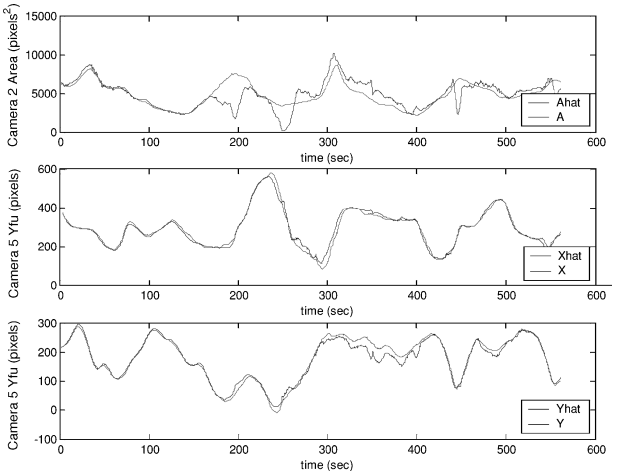
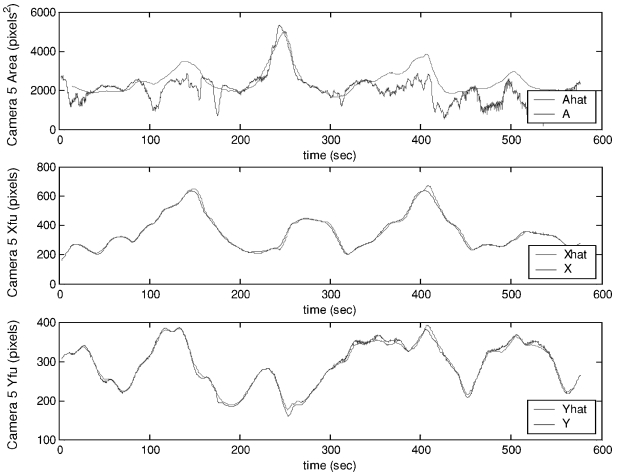


Fig. 8 x-axis EKF position and velocity estimates with 3σ error in global (i.e., neutral buoyancy tank) coordinates.



a) Camera 2 (worst-case centroid agreement)



b) Camera 5 (typical centroid agreement)

Fig. 9 Measured vs estimated centroid and area data in local camera frames.

case, 3σ error is consistently low because a sufficient set of the six cameras kept SCAMP in their FOV at all times.

Next, the six-camera EKF result was compared with the local measurements provided by the six individual cameras. The goal of these tests was to assess the agreement of each camera measurements with the averaged EKF state estimate. Figure 9 illustrates area and centroid comparisons for cameras 2 and 5. Three plots were generated for each camera, with the six-camera EKF estimates

transformed into each camera's local frame and converted to pixel units to enable correlation with individual camera measurements. The first compares SCAMP area (pixels^2) measured by the camera with the predicted image area computed from the VPS estimate $\hat{A}$. The second and third plots compare the measured X_{FU} and Y_{FU} centroid coordinates (pixels) with the predicted centroid coordinates $\hat{X}_{\mathrm{FU}}$ and $\hat{Y}_{\mathrm{FU}}$. Generally, centroid measurements agree closely with the EKF estimates, with notable exceptions. These plots also illustrate significant area discrepancies, validating our decision to use camera centroid but not area data in EKF estimates.

The most significant centroid error observed is shown in Fig. 9a (camera 2). In this case, measured and predicted Y-centroid values vary between times 290–400 s. Based on the data and locations of other objects in the tank, this phenomenon appears when SCAMP flies partially in front of a similarly colored object, as was illustrated in Fig. 7a. Part of the vehicle is indistinguishable from the background, making it appear offset from its true position. Despite such error sources, on average, each camera experienced between 2–4 pixels of noise, with camera 2 noise somewhat higher because of the background interference. As a reference, one pixel is equal to ~8 mm for all cameras when SCAMP is near the center of the neutral buoyancy tank.

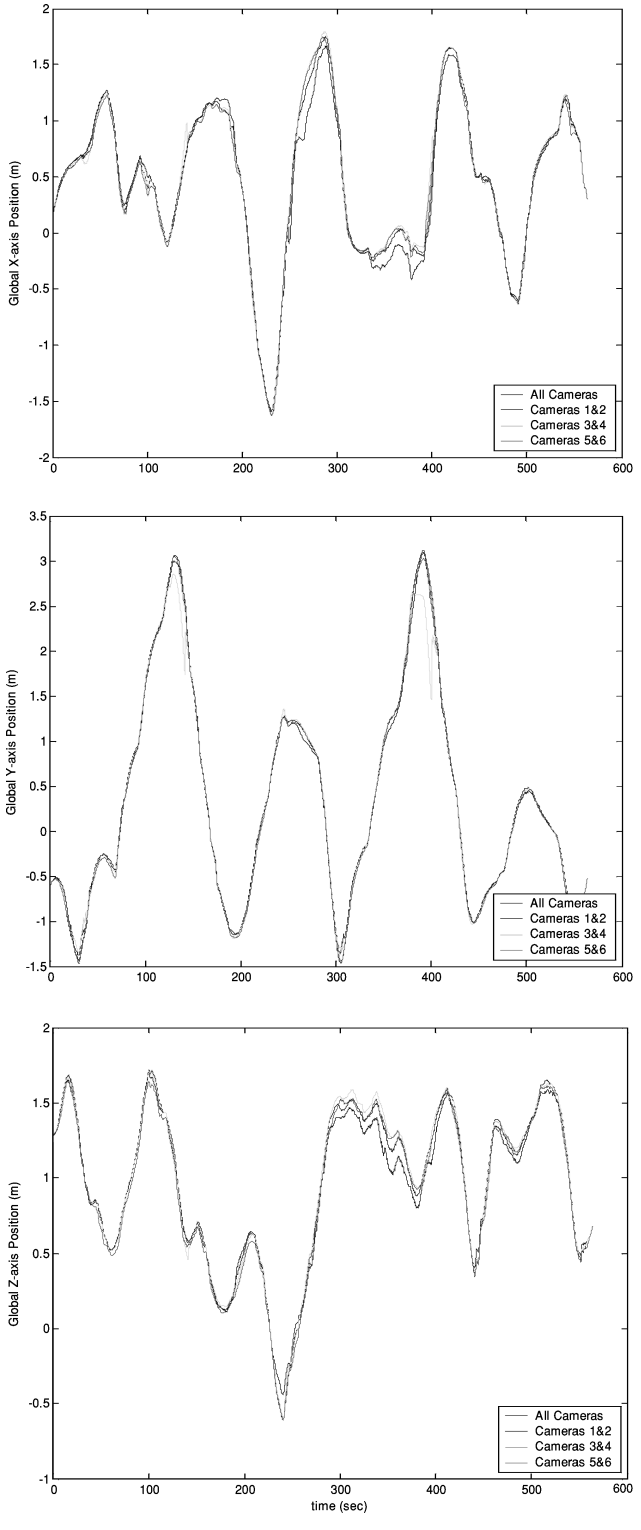


Fig. 10 GPS found with four EKF camera combinations.

Table 6 Comparison of six-camera EKF and two-camera EKF estimates

	Global x, m		Global y, m		Global z, m	
EKF	Average diff	σ	Average diff	σ	Average diff	σ
CAM12	0.054	0.061	0.0028	0.0025	0.036	0.034
CAM34	0.038	0.040	0.0030	0.0028	0.055	0.132
CAM56	0.039	0.034	0.0037	0.0030	0.031	0.024

To examine the differences between the six-camera EKF estimate and EKF estimates with fewer cameras, a simulator was constructed that developed EKF estimates from camera and vehicle thruster/attitude data acquired during the test shown in Figs. 8 and 9. For each simulation, measurements were provided for a single camera pair, resulting in three different EKF estimates: a) cameras 1-2, b) cameras 3-4, and c) cameras 5-6. Note that two cameras is the minimum set required for three-dimensional estimates given the use only of two-dimensional centroid image data; camera pairs were selected to provide data for all three dimensions.

The global X, Y, Z positions of SCAMP vs time are shown in Fig. 10 for all three two-camera EKFs as well as the original six-camera EKF. Generally, the minimal two-camera data set estimates agree with all other EKF estimates. The most notable difference is caused by the noisy camera 2 measurements just described, which causes discrepancies primarily to the camera 1-2 EKF. Because of the use of identical dynamic models, the X, Y, Z velocity plots for all four EKFs matched quite well and are not illustrated in this paper.

Table 6 summarizes differences between the six-camera EKF estimates and the corresponding two-camera estimates. As a conservative measure of error, these results include all data points, including estimates where camera 2 provided poor quality data. The results overall are promising. First, it can be seen that with good data the use of just two cameras is sufficient to yield a state estimate accurate to the centimeter level, indicating that SCAMP can be tracked almost anywhere in the tank (see Fig. 2 coverage maps). Second, it can be seen that with highly redundant measurements (from six total cameras in this case), even relatively inaccurate data (e.g., camera 2 between $t = 290$–400 s) do not significantly degrade the six-camera overall state estimate.

Conclusions

This paper describes the design and implementation of a vision-based navigation system, the vision positioning system (VPS), for a neutral buoyancy space simulation environment. A robust two-stage camera calibration technique was developed for the large-volume neutral buoyancy tank, and an extended Kalman filter combined processed image data and dynamic force/torque data to compute three-dimensional position and velocity estimates updated at 10 Hz. VPS has been fully implemented in the University of Maryland's Neutral Buoyancy Research Facility and has been shown to provide state estimates for the SCAMP free-flying robot with centimeter-level accuracy. Given the accurate (millimeter-level) calibration and statistical (EKF) compilation of measurement data, even the simple centroid computation algorithms currently implemented provide centimeter-level accuracy comparable to GPS without differential correction. VPS technology will enable motion characterization for a variety of objects ranging from free-flying robots (e.g., SCAMP) to astronauts training for EVA tasks, requiring (as in space) more accurate local navigation systems only for precision operations such as grappling

or docking. VPS also enables full six-DOF control of free-flying neutral buoyancy robots such as SCAMP, a critical capability for space inspection, assembly, or repair tasks under consideration for NASA exploration missions.

Perhaps the most significant contribution of this research is the development of procedures and equipment that allow an accurate, repeatable, and robust calibration of VPS. The two-step calibration procedure overcomes several challenges, including a large calibration volume and maintenance of centimeter-scale accuracy over long distances with inward-pointing cameras. The discrete extended Kalman filter applied to the VPS state estimation task was shown to provide estimates with centimeter-level precision in static and dynamic testing with the SCAMP vehicle.

Although the presence of background clutter and significant lighting changes degrade accuracy, the EKF's statistical averaging properties reduce this noise such that slightly larger but still centimeter-level errors were present. Currently, multiple measurements are required to reject noise for centimeter-level accuracy. With further refinement of the dynamic model and more precise definition of R and Q EKF matrices, EKF estimates are expected to improve further, moving toward a model where the vehicle enjoys centimeter-level positioning accuracy almost anywhere in the neutral buoyancy tank. More advanced image processing algorithms will also improve vehicle centroid estimates, and work is underway to window images around the expected target position to ignore far-field clutter. Ultimately, we plan to investigate object recognition algorithms that match SCAMP geometry with its two-dimensional projection in each camera's image plane based on vehicle attitude from SCAMP telemetry. This will correct for geometric asymmetry and will better reject background objects of dissimilar geometry. Once six-DOF vehicle control is robust to tank clutter, we also plan to study formation flight navigation and control technologies as well as space inspection and cooperative astronaut-robot operations, with VPS providing position estimates for closed-loop control or as truth measurements for local navigation or object motion characterization.

Acknowledgments

The authors would like to thank Cat McGhan, Rhiannon Peasco, Mike Naylor, Tim Wasserman, and the cast of University of Maryland Space Systems Lab (SSL) divers for their tireless assistance. We also very much appreciate the support and advice of Dave Akin, director of the SSL.

References

[1]Agrawal, B. N., and Rasmussen, R. E., "Air Bearing Based Satellite Attitude Dynamics Simulator for Control Software Research and Development," *Proceedings of SPIE—The International Society for Optical Engineering*, edited by R. L. Murrer, Jr., International Society for Optical Engineering, Bellingham, WA, Vol. 4366, 2001, pp. 204–214.

[2]Aircraft Operations Div., *JSC Reduced Gravity Program User's Guide*, NASA, Houston, Texas, 2000.

[3]Odenbach, S., "Drop Tower Experiments on Thermomagnetic Convection," *Microgravity Science and Technology*, Vol. 6, No. 3, 1993, pp. 161–163.

[4]Brown, H. B., and Dolan, J. M., "Novel Gravity Compensation System for Space Robots," *ASCE Specialty Conference on Robotics for Challenging Environments*, edited by L. A. Demsetz and P. R. Klarer, American Society of Civil Engineers, New York, Feb.–March 1994.

[5]Creamer, G., and Hollander, S., "The Spacecraft Robotics Engineering and Control Laboratory," Naval Research Lab., Washington, DC (version current 2004), URL: http://www.nrl.navy.mil/content.php?P=02REVIEW207.

[6]Cobb, H. S., "GPS Pseudolites: Theory, Design, and Applications," Ph.D. Dissertation, Dept. of Aeronautics and Astronautics, Stanford Univ., Palo Alto, CA, 1997.

[7]Kowalski, K. G., "Applications of a Three-Dimensional Position and Attitude Sensing System for Neutral Buoyancy Space Simulation," MS. Thesis, Dept. of Aeronautics and Astronautics, MIT, Cambridge, MA, Oct. 1989.

[8]Churchill, P. J., "Position and Attitude Station-Keeping of a Free-Flying Telerobotic Vehicle," MS. Thesis, Aerospace Engineering Dept., Univ. of Maryland, College Park, Dec. 1993.

[9]Williams, T., and Tanygin, S., "On-Orbit Engineering Test of the AERCam Sprint Robotic Camera Vehicle," *Advances in the Astronautical Sciences*, Vol. 99, No. 2, 1998, pp. 1001–1020.

[10]Chatterji, G., Menon, P., and Sridhar, B., "Vision-Based Position and Attitude Determination for Aircraft Night Landing," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 1, 1998, pp. 84–92.

[11]Gurfil, P., and Rotstein, H., "Partial Aircraft State Estimation from Visual Motion Using the Subspace Constraints Approach," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 5, 2001, pp. 1016–1028.

[12]Campa, G., Seanor, B., Perhinschi, M., Fravolini, M., Ficola, A., and Napolitano, M., "Autonomous Aerial Refueling for UAVs Using a Combined GPS-Machine Vision Guidance," AIAA Paper 2004-5350, Aug. 2004.

[13]Pollini, L., Mati, R., and Innocenti, M., "Experimental Evaluation of Vision Algorithms for Formation Flight and Aerial Refueling," AIAA Paper 2004-4918, Aug. 2004.

[14]Atkins, E. M., Lennon, J. A., and Peasco, R. S., "Vision-Based Following for Cooperative Astronaut-Robot Operations," *Proceedings of the IEEE Aerospace Conference*, Vol. 1, Institute of Electrical and Electronics Engineers, New York, 2002, pp. 215–224.

[15]Sim, R., and Dudek, G., "Learning Environmental Features for Pose Estimation," *Image and Vision Computing*, Vol. 19, No. 11, 2001, pp. 733–739.

[16]Campbell, A., Sukthankar, R., and Nourbakhsh, I., "Techniques for Evaluating Optical Flow for Visual Odometry in Extreme Terrain," *Proceedings of the International Conference on Intelligent Robots and Systems*, IEEE Robotics and Automation Society, New York, Oct. 2004, pp. 3704–3711.

[17]Lowe, E., "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, Vol. 60, No. 2, 2004, pp. 91–110.

[18]Bartoli, A., "A Unified Framework for Quasi-Linear Bundle Adjustment," *Proceedings of the Sixteenth IAPR International Conference on Pattern Recognition*, IEEE Computer Society, Washington, DC, 2002, pp. 560–563.

[19]Hossaini, L. S., "The Design and Analysis of a Second Generation Free Flying Underwater Camera Platform," M.S. Thesis, Aerospace Engineering Dept., Univ. of Maryland, College Park, May 2000.

[20]Tsai, R. Y., "A Versatile Camera Calibration Technique for High-Accuracy 3-D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses," *IEEE Journal of Robotics and Automation*, Vol. RA-3, No. 4, 1987, pp. 323–344.

[21]Clarke, T. A., and Fryer, J. F., "The Development of Camera Calibration Methods and Models," *Photogrammetric Record*, Vol. 16, No. 91, 1998, pp. 51–66.

[22]Heilkkilä, J., and Silvén, O., "A Four-Step Camera Calibration Procedure with Implicit Image Correction," *Proceedings of the IEEE Computer Vision and Pattern Recognition Conference*, Vol. 1, IEEE Computer Society, Washington, DC, 1997, pp. 1106–1112.

[23]Zhang, Z., "Flexible Camera Calibration by Viewing a Plane from Unknown Orientations," *Proceedings of the Seventh International IEEE Conference on Computer Vision*, Vol. 1, IEEE Computer Society, Washington, DC, 1999, pp. 666–673.

[24]Franklin, G., and Powell, J., *Digital Control of Dynamic Systems*, Addison Wesley Longman, Reading MA, 1980, pp. 135, 136.

[25]Gelb, A., *Applied Optimal Estimation*, MIT Press, Cambridge, MA, 1974, pp. 188,189.