PERFORMING VISUAL SURVEYS OF NON-PLANAR BENTHIC TERRAIN

Kiran Murthy[1]
kmurthy@stanford.edu

Stephen Rock[1,2]
rock@stanford.edu

[1]Dept. of Aeronautics & Astronautics
Stanford University
496 Lomita Mall
Stanford, CA 94305

[2]Monterey Bay Aquarium Research Institute
7700 Sandholdt Road
Moss Landing, CA 95039

Abstract

A key issue facing any ROV or AUV-based benthic visual survey is how to point the camera normal to the terrain during the survey. Though planar terrain surveys can be executed using a constant camera orientation, non-planar terrain surveys require that the commanded camera orientation change during vehicle movement..

This paper proposes a method that generates a smooth camera orientation command by first spatially filtering local terrain orientation measurements through the construction of a spline-based model of the terrain. The method then calculates an orientation command from the spline model using the spline point projection algorithm. A real-time version of this algorithm has been fielded on a Monterey Bay Aquarium Research Institute ROV, allowing the ROV to maintain camera perpendicularity autonomously over non-planar benthic sites and enabling small-scale visual surveys of these sites.

1 Introduction

Visual surveys of the benthic environment aim to attain video or still images of a site of interest using a camera mounted on an ROV or AUV. Though such surveys have been conducted successfully over planar terrain [1, 2, 3, 4], several issues arise when performing non-planar terrain surveys. This paper deals with a specific issue: generating a camera orientation command that will point the camera toward the terrain.

During the survey, it is desirable for a visual survey to maintain camera-terrain perpendicularity in order to minimize the perspective distortion experienced when taking images of the terrain. While camera perpendicularity over planar terrain can be maintained by holding a constant camera orientation, non-planar terrain surveys require that the camera change orientation while traversing the terrain. Local terrain orientation can be measured by ranging sensors such as the Doppler Velocity Logger (DVL), and it is possible to control camera orientation based on these measurements. However, controlling camera orientation based solely on local terrain orientation is undesirable due to the high spatial frequency content created by local relief (e.g. small rocks) in the otherwise smooth terrain (see Figure 1).

Even when temporally-filtered local terrain measurements are used to command an orientation control system, spatial frequency content persists and produces undesirable results (see Figure 2). Thus, spatial filtering is required to attenuate the high spatial frequency content.

In this paper, the required spatial filtering is performed by fitting a spline surface to DVL range measurements. A smooth camera orientation command that maintains camera-terrain perpendicularity is then extracted directly from the spline surface.

A spline-based representation of the terrain has three advantages over other spatial smoothing methods:

1. A spline surface can be fit to incoming DVL measurements in real-time using a linear Kalman filter.

2. Using the spline-relative vehicle position es-

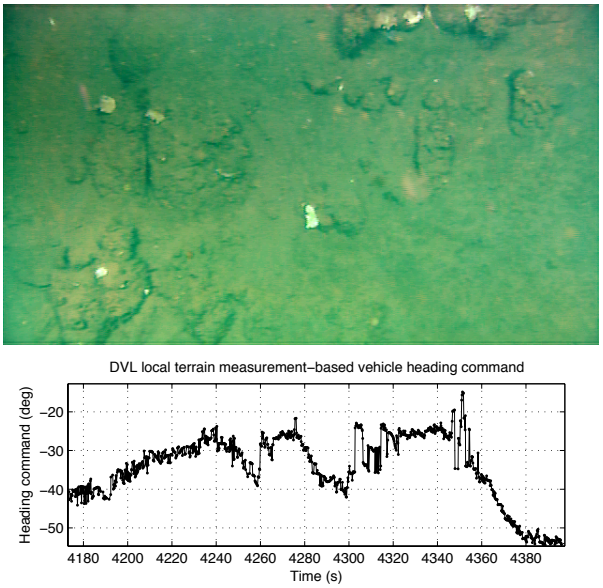

Figure 1: Top: benthic terrain image, showing local relief (small rocks) in smooth benthic terrain. Bottom: resulting orientation (heading) command containing high spatial frequency components due to local relief

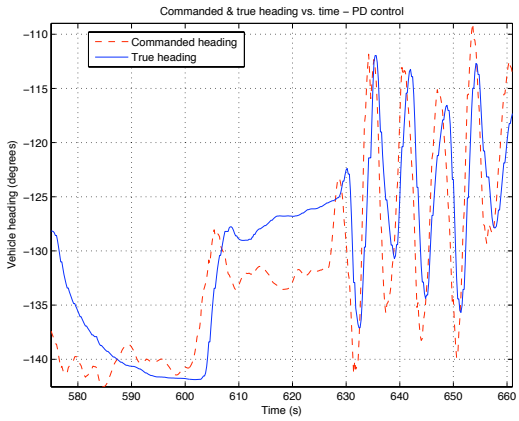

Figure 2: Large oscillations when PD control system is tied to a temporally-filtered local terrain orientation measurement

timate, an orientation command can be generated using the spline point projection algorithm. Deriving the orientation command from a spline representation of the terrain guarantees that the orientation command will be spatially-smooth.

3. The properties of spline surfaces offer an approximate choice of spatial smoothing bandwidth.

Results from ROV field trials at non-planar sites in Monterey Bay, CA show that the spline-based spatial filtering method produces a smooth camera orientation command that points the camera perpendicular to the terrain successfully. In addition, a control loop tracking this orientation command has enabled the execution of small-scale surveys over non-planar terrain.

2 Background

During planar terrain visual surveys, camera-terrain perpendicularity can be maintained in a straightforward manner by holding a constant camera orientation [2]. In addition, maintaining camera perpendicularity during visual surveys of smooth non-planar surfaces such as ship hulls is also relatively straightforward. Since the surface is smooth, spatial filtering is not required and local surface measurements can be used to select a camera orientation [5].

On the other hand, maintaining camera perpendicularity during visual surveys over non-planar terrain can be problematic due to the high spatial frequency content in the terrain. Solutions can be derived from previous work in either 3-D terrain reconstruction or Simultaneous Localization and Mapping (SLAM), but those solutions are typically computationally-complex and do not lend themselves to the task of performing simple visual surveys. These methods usually store the reconstructed terrain as features, voxels, or a grid of elevations [6, 7, 8], and such terrain representations rely on heuristics to generate a camera orientation command [8]. However, there exist many non-planar sites that are not complex enough to necessitate an accurate 3-D terrain reconstruction to attain camera-terrain perpendicularity. At these sites, the terrain's low spatial frequency information can be used to maintain perpendicularity.

Spline surfaces are a well-studied mathematical construct that can capture this low spatial frequency information. There has been an abundance of work on spline reconstruction within the computer graphics community in order to create smooth representations of objects using range scan data. The most relevant work has focused on real-time spline surface reconstruction of an object by modifying a spline surface with Kalman filtered range scan measurements [9, 10, 11]. This work lays the foundation for using DVL measurements to create

a real-time spline-based terrain reconstruction that enables the generation of a smooth camera orientation command.

3 Algorithm

The algorithm creates a smooth representation of the terrain from which a camera orientation can be calculated that will point the camera perpendicular to the terrain. Figures 3 and 4 illustrate the proposed method, which first updates a smooth spline to fit DVL-based terrain measurements, then uses the updated spline to generate a spatially-smooth camera orientation command that maintains camera perpendicularity.

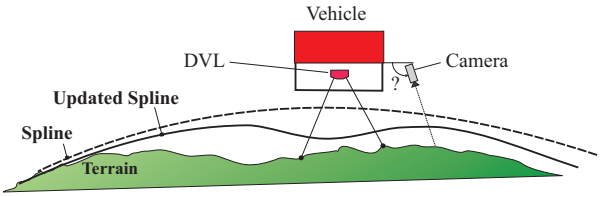

Figure 3: Spline being updated with DVL range measurements.

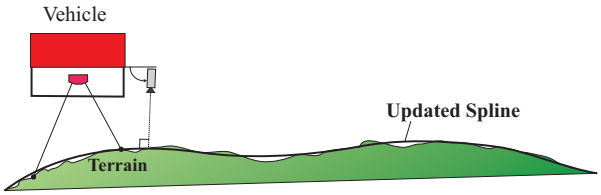

Figure 4: Spline updated with many range measurements being used to choose camera orientation

The algorithm consists of two steps:

1. **Spline Kalman Update:** The spline-based model of the terrain is updated with the most recent DVL range measurements using a linear Kalman filter measurement update step. The Kalman filter's state vector tracks both the shape of the spline as well as the spline-relative vehicle position error.

2. **Vehicle Position Projection:** The point on the spline surface closest to the camera's position is found using the point projection algorithm. The surface normal vector's direction at the projected point is interpreted as the camera orientation command that places the camera perpendicular to the surface.

3.1 Spline Representation of a Surface

Splines are functions that map a parameter domain into an n-dimensional manifold by way of a series of polynomial basis functions forming linear combinations of n-dimensional points. The spline functions used in this paper take the form of "tensor product cubic B-spline surfaces," which map a plane (u, v) to 3-D points using cubic basis functions. The basic B-spline surface equation is

$$\mathbf{S}(u,v) = \sum_{j=0}^{n}\sum_{k=0}^{m} N_{j,p}(u)N_{k,p}(v)\mathbf{P}_{j,k} \qquad (1)$$

where $\mathbf{S}(u, v)$ is the 3-D point on the spline surface at a particular parameter tuple (u, v), $N_{j,p}$ and $N_{k,p}$ are the jth and kth B-spline basis functions of polynomial degree p, and $\mathbf{P}_{j,k}$ are a set of 3-D spline surface control points. In this paper, the polynomial degree p is always 3.

The B-spline basis functions are structured such that at any given parameter value, only $p + 1$ basis functions are active (nonzero). The resulting spline surface becomes C^p continuous with respect to changes in (u, v), but is allowed to break one degree of continuity at specific parameter values called knots [12]. At the knots, one active basis function becomes inactive and one inactive basis function becomes active [12]. In addition, the number of knots in a parametric direction directly determines the number of control points in that direction.

Since B-spline surfaces represent combinations of smooth piecewise polynomial functions, they are able to represent the low spatial frequency variations in surfaces. In particular, since the knots denote breaks in the spline's C^p continuity, the number of knots in a spline surface is representative of the smoothness of a spline [12]. Hence, the spatial frequency represented by the spline surface can be changed by modifying the number of knots in the spline, allowing for control over the spatial filtering bandwidth. Additionally, two characteristics of tensor product B-spline surfaces allow for a well-behaved update to the spline to reflect measurements of the surface:

- **Strong convex hull property:** The surface $\mathbf{S}(u, v)$ lies within the convex hull of its control points [12]. Due to this property, the control points can be interpreted as a coarse representation of the spline [13]. Thus, making a small change to the location of a control point will make a small change to the spline in a well-behaved manner.

- **Local support:** The notion of $p+1$ active basis functions results in $(p+1) \times (p+1)$ control points determining the location of a specific point $\mathbf{S}(u, v)$ [12]. Therefore, if a measured point on the surface does not match the associated point on the spline, a limited number of control points need to be changed for the spline to better reflect the measurement. In addition, the changed control points only affect a local area of the spline surface: distant areas of the surface are NOT changed by the measurement update [12].

Finally, another property of B-spline surfaces enables the robust determination of spline surface normal:

- **Differentiability:** The first partial derivatives of the cubic B-spline surface is well defined at all locations on the spline [12]. This leads to the reliable determination of spline surface normal at any given (u, v) tuple.

3.2 Spline Kalman Update

The spline Kalman update incorporates the most recent DVL measurements into an estimate of the shape of the spline-based terrain model (control point locations $\mathbf{P}_{j,k}$), and the error in the spline-relative vehicle position, named $\mathbf{B}$. The Kalman update recasts $\mathbf{P}_{j,k}$ and $\mathbf{B}$ as a state vector $\vec{P}$, then uses the linear relation between the control points and the spline surface in order to update the estimates.

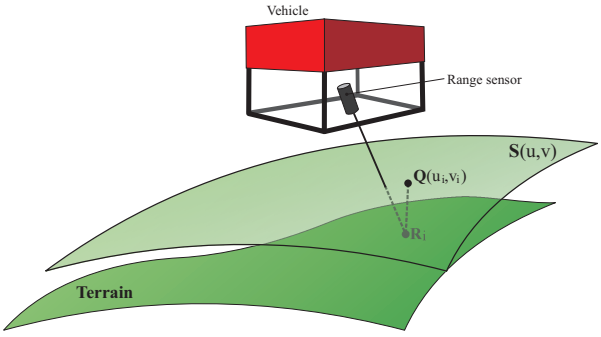

Figure 5: Definitions for spline surface Kalman update

Figure 5 depicts the spline Kalman update definitions. Assume $\mathbf{x}$ is the best spline-relative estimate of vehicle position. Let r_i be the ith range measurement (scalar) along a ray with known orientation with respect to a vehicle-fixed frame. Let $\mathbf{R}_i$ be the 3-D point obtained by projecting the range measurement r_i from $\mathbf{x}$ along the range measurement's ray. Let $\mathbf{Q}_i$ be the 3-D point on the spline surface closest to $\mathbf{R}_i$, with (u_i, v_i) being the spline parameters at the point. These parameters are computed using the point projection algorithm presented in Section 3.3.

From Equation 1, a relation between $\mathbf{Q}_i$ and the spline surface control points can be written:

$$\mathbf{Q}_i = \sum_{j=0}^{n} \sum_{k=0}^{m} N_{j,p}(u_i) N_{k,p}(v_i) \mathbf{P}_{j,k} \tag{2}$$

In order for the spline surface to match the range measurements, $\mathbf{R}_i$ should equal $\mathbf{Q_i}$. Thus, the measurement model equation becomes:

$$\mathbf{R}_i = \sum_{j=0}^{n} \sum_{k=0}^{m} N_{j,p}(u_i) N_{k,p}(v_i) \mathbf{P}_{j,k} \tag{3}$$

where the goal is to estimate $\mathbf{P}_{j,k}$ using measurements $\mathbf{R}_i$. Thus, the Kalman filter aims to modify the spline control point locations $\mathbf{P}_{j,k}$ in order to match the point on the spline $\mathbf{S}(u_i, v_i)$ to the range measurement point $\mathbf{R}_i$.

For simplicity, Equation 3 is written in matrix-vector form:

$$\begin{aligned}(\mathbf{R}_i)_d &= \left[\, N_{0,p}(u_i) \;\cdots\; N_{n,p}(u_i) \,\right] \left[\, (\mathbf{P}_{j,k})_d \,\right] \begin{bmatrix} N_{0,p}(v_i) \\ \vdots \\ N_{m,p}(v_i) \end{bmatrix} \\ &= N_{u_i}^T P_d N_{v_i}\end{aligned} \tag{4}$$

for each of the three dimensions $d = \{x, y, z\}$. However, since the range measurement $\mathbf{R}_i$ is projected from a potentially erroneous estimate of spline-relative vehicle position, the range measurement may be biased by the vehicle position error. This bias is especially relevant if the initial spline is derived from *a priori* map data, as the vehicle's position with respect to the map may be biased. In order to alleviate the bias, it is explicitly estimated; the new measurement model equation is:

$$(R_i)_d = N_{u_i}^T P_d N_{v_i} + B_d + \epsilon \tag{5}$$

where B is the bias to be estimated, and $\epsilon \sim \mathcal{N}(0, \Sigma_\epsilon)$ is random Gaussian noise that is incurred in the range measurement. Note that due to the Gaussian noise assumption and the linear measurement model with respect to the parameters (P, B), a Kalman filter can be used to optimally update the parameters using the measurements.

To write Kalman filter update equations, the parameters need to be formed into a column vector. "Vectorization" is defined as the operation of converting a matrix into a column vector by vertically concatonating the matrix's columns. Define $\vec{P}_d$ be the vectorized version of the matrix $(\mathbf{P}_{j,k})_d$ and C_i be the vectorized version of the matrix $N_{u_i} N_{v_i}^T$. The measurement model equation can now be written:

$$\mathbf{R}_i = \begin{bmatrix} C_i^T & 0 & 0 & 1 & 0 & 0 \\ 0 & C_i^T & 0 & 0 & 1 & 0 \\ 0 & 0 & C_i^T & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \vec{P}_x \\ \vec{P}_y \\ \vec{P}_z \\ \mathbf{B} \end{bmatrix} + \epsilon = C\vec{P} + \epsilon \tag{6}$$

The vectorized measurement model equation now leads to the Kalman filter update equation. Denoting Σ_P the covariance of the estimation state $\vec{P}$, we can compute Σ_R, the covariance of $\mathbf{R}_i$:

$$\Sigma_R = C\Sigma_P C^T + \Sigma_\epsilon \tag{7}$$

Finally, the Kalman measurement update step becomes:

$$\vec{P} := \vec{P} + \Sigma_P C^T \Sigma_R^{-1} \left(\mathbf{R}_i - C\vec{P} \right) \tag{8}$$

with the following covariance update:

$$\Sigma_P := \Sigma_P - \Sigma_P C^T \Sigma_R^{-1} C \Sigma_P \tag{9}$$

The updated spline is then constructed by reconstructing each control point $\mathbf{P}_{j,k}$ from $\vec{P}$, then adding the spline-relative vehicle position error $\mathbf{B}$ to each control point. The error can simply be added to the control points since any affine transform can be applied to a spline by applying the same transform to the spline's control points [12]. After the update, the spline's shape better models DVL terrain measurements and the spline's position cancels the estimated vehicle position error.

3.3 Point Projection Algorithm

Assuming a spline surface consistent with DVL measurements has been computed, the spline point projection algorithm determines the camera perpendicularity orientation. A well-known technique in the geometric modeling community, the point projection algorithm aims to find the point on the spline surface closest to a given point. If the vehicle position is projected onto the spline surface, the spline normal vector $\hat{\mathbf{n}}$ at the projected point is the direction that the camera must point in order to achieve perpendicularity.

The following is an overview of the spline point projection algorithm [12]. Let $\mathbf{S}(u, v)$ be the spline surface, as described in Equation 1, and let $\mathbf{P}$ be the point to be projected. Also, define

$$\mathbf{r}(u, v) = \mathbf{S}(u, v) - \mathbf{P} \tag{10}$$

as the vector from $\mathbf{P}$ to the spline surface at particular (u, v) values.

The following set of equations must be solved in order to find the (u, v) values that minimize the distance between $\mathbf{P}$ and the spline:

$$\begin{aligned} f(u, v) &= \mathbf{r}(u, v) \cdot \mathbf{S}_u(u, v) = 0 \\ g(u, v) &= \mathbf{r}(u, v) \cdot \mathbf{S}_v(u, v) = 0 \end{aligned} \tag{11}$$

where $\mathbf{S}_u(u, v)$ and $\mathbf{S}_v(u, v)$ are the partial derivative vectors of the spline surface with respect to u and v. The solution (u^*, v^*) to this set of equations yields a vector $\mathbf{r}(u^*, v^*)$ that is either zero magnitude ($\mathbf{P}$ lies on the spline surface), or is normal to the spline surface ($\mathbf{S}(u^*, v^*)$ is the point on the spline surface closed to $\mathbf{P}$).

A Newton iteration can efficiently solve the set of equations. Computing the Jacobian of the equations:

$$J(u, v) = \begin{bmatrix} ||\mathbf{S}_u||^2 + \mathbf{r} \cdot \mathbf{S}_{uu} & \mathbf{S}_u \cdot \mathbf{S}_v + \mathbf{r} \cdot \mathbf{S}_{uv} \\ \mathbf{S}_u \cdot \mathbf{S}_v + \mathbf{r} \cdot \mathbf{S}_{vu} & ||\mathbf{S}_v||^2 + \mathbf{r} \cdot \mathbf{S}_{vv} \end{bmatrix} \tag{12}$$

Using the Jacobian, the multi-dimensional Newton iteration step can be written:

$$\begin{bmatrix} u \\ v \end{bmatrix} := \begin{bmatrix} u \\ v \end{bmatrix} - J^{-1}(u, v) \begin{bmatrix} f(u, v) \\ g(u, v) \end{bmatrix} \tag{13}$$

After Newton iteration convergence, the surface normal unit vector at the vehicle position's projected location can be computed by:

$$\hat{\mathbf{n}} = \frac{\mathbf{S}_u(u^*, v^*) \times \mathbf{S}_v(u^*, v^*)}{||\mathbf{S}_u(u^*, v^*) \times \mathbf{S}_v(u^*, v^*)||} \tag{14}$$

A good initial guess of (u, v) is required for proper convergence of the Newton iteration. Generally, an initial guess is formed by computing a set of candidate points on the spline and starting the Newton iteration at the candidate point closest to $\mathbf{P}$. In a real-time system that computes the desired camera orientation at every time step, the previous time step's spline projection result can be used as the initial condition for the current time step.

Note that the current vehicle position estimate can be used for the given point $\mathbf{P}$. Since the spline surface has been updated to accurately represent the

range measurements, the camera orientation that achieves perpendicularity at the current vehicle position is along the surface normal vector in Equation 14. Also, since $\hat{\mathbf{n}}$ has been derived directly from the smooth spline surface, the vector's orientation is spatially smooth.

4 Results

4.1 Data Acquisition Framework

Field data collected using the ROV Ventana (Figure 6) have been used to verify the algorithm's viability. Ventana is operated by the Monterey Bay Aquarium Research Institute (MBARI). It's sensor suite includes:

1. A high definition camera that can be manually panned and tilted by the ROV pilot.
2. A fiber optic gyroscope for accurate an unbiased orientation measurements.
3. A DVL containing four sonar beams that record range and velocity measurements at 5 Hz. The velocity measurements can be integrated to produce a drift-laden ROV position estimate relative to an unknown inertially-fixed point.
4. An Ultra-Short BaseLine (USBL) sensor that produces coarse (5 m accuracy), but unbiased measurements of the ROV's inertial position at a rate of 1 Hz. These measurements are necessary to localize the ROV if *a priori* georeferenced bathymetry data is used to initialize the algorithm (see Section 4.2).

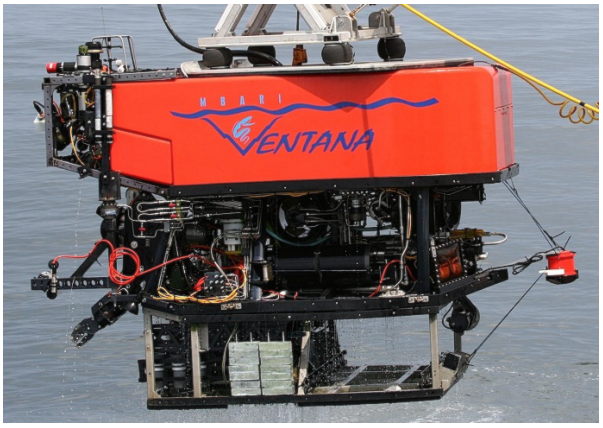

Figure 6: The ROV Ventana operated by the Monterey Bay Aquarium Research Institute (MBARI)

In addition, the ROV has actuator authority in four degrees of freedom: axial, lateral, vertical, and yaw. Roll and pitch are held near zero by way of the ROV's natural buoyancy. The camera's manual tilt actuation gives the camera an additional, yet uncontrollable degree of freedom.

The ROV acquired the data over two benthic waypoints in Monterey Bay, known as "Clam Field South" and "Soquel #4." Prior information regarding the shape of the terrain was available in the form of a discrete elevation map (DEM) with a terrain depth measurement at each vertex of a regularly-spaced grid registered in Universal Transverse Mercator (UTM) coordinates. Clam Field South's DEM had a 5 meter resolution, while Soquel #4's DEM had a 1 meter resolution.

During acquisition of data at both sites for algorithm post-processing, the ROV pilot drove a trajectory over the non-planar sites at a rate of approximately 0.2 m/s. The pilot was tasked with keeping the camera perpendicular to the terrain at all times by holding the camera tilt at the terrain's elevation angle, and then pointing the ROV uphill. During the maneuver, the ROV's UTM position was calculated by fusing the unbiased low-frequency USBL position measurement and the biased high-frequency DVL position measurement via a complimentary filter [14]. The calculated position, ROV orientation, and DVL range measurements were stored at a 5 Hz rate.

Real-time algorithm execution results have been obtained at the Soquel #4 site. During real-time algorithm execution, three automatic control loops were in place to control three degrees of freedom, as described in Table 1. The fourth degree of freedom (lateral) was controlled by the pilot slowly slewing the ROV to port or starboard.

Table 1: Active automatic control loops during real-time algorithm execution

DOF	Sensor	Commanded Value
Vertical	Pressure depth	Constant
Altitude	Camera-mounted altimeter	Constant
Heading	Fiber optic gyro	Formed by algorithm

Figure 7 shows the ROV's trajectory over the Clam Field South and Soquel #4 DEMs. Note that although the *a priori* bathymetry data around the Soquel #4 waypoint indicates nearly planar terrain, the actual terrain contains several corners that necessitate large changes in vehicle heading to keep the camera pointed perpendicular to the terrain. Updating the spline terrain model with DVL range measurements has reconstructed these corners and

the resulting orientation command appropriately changes orientation around the corners.

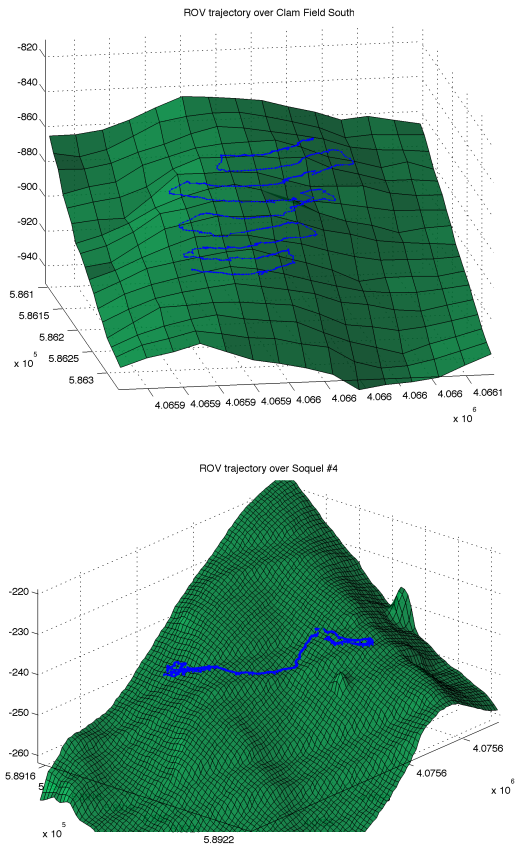

Figure 7: Top: ROV's pilot-driven trajectory over the Clam Field South waypoint. Bottom: ROV's trajectory over Soquel #4 waypoint with three degrees of freedom under automatic control.

To collect benthic images during real-time algorithm execution, a real-time image texture correlation algorithm described in [15] has been employed to track the 2-D image offset from the previously acquired reference image. After the 2-D image offset reaches a certain Euclidean threshold, the camera's image is stored as the new reference image. The collected reference images along with the associated reference image metadata is then used to plot a sequence of overlapping images showing site coverage.

4.2 Algorithm Initialization

The spline surface Kalman filter update must be fed initial control points and covariances. If *a priori* terrain data exists, the spline control points can be initialized to a least-sqares spline fit of the terrain. Otherwise, the spline surface initialization can be based on measurements taken at the site. The online measurement-based spline is a plane created using measurements from the first swath of the trajectory. The surface normal estimates from the DVL are averaged over the swath to produce the plane's normal vector. The plane is anchored at the initial ROV position. Figure 8 shows the *a priori* and online measurement-based start conditions using the Clam Field South data.

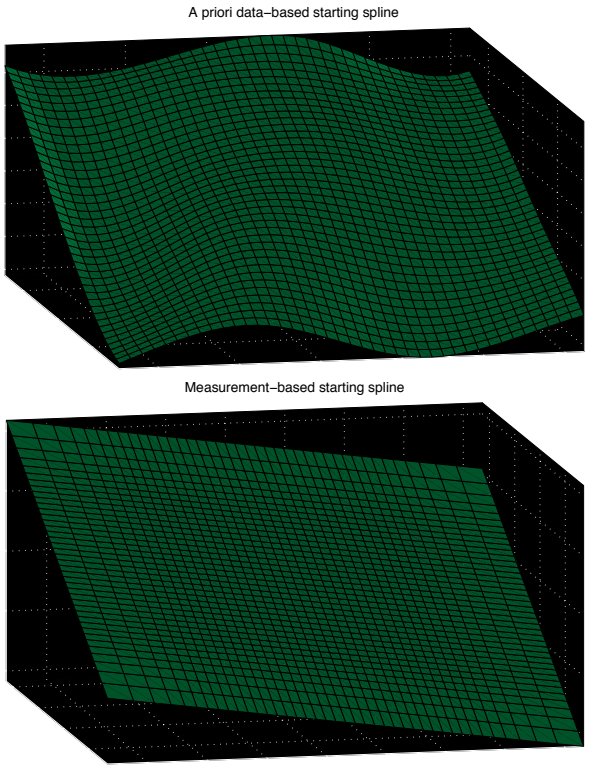

Figure 8: Initial spline terrain representation of Clam Field South fit from *a priori* data (top) and online terrain measurements (bottom).

Both splines are computed by first choosing a parameter that captures the spatial frequency of the spline. This parameter is defined as the number of control points used to describe 1 km of space, since the number of control points is defined by the number of knots, and the number of knots generally captures the smoothness of the spline [12]. Using 30 control points per km produces a smooth yet accurate representation of the terrain at the Clam Field South waypoint, while the sharper corners at the Soquel #4 waypoint are better represented by using 50 control points per km.

4.3 Post-Processing Results

Kalman updating the spline surfaces using DVL range measurements produces the spline surfaces shown in Figures 9 and 10.

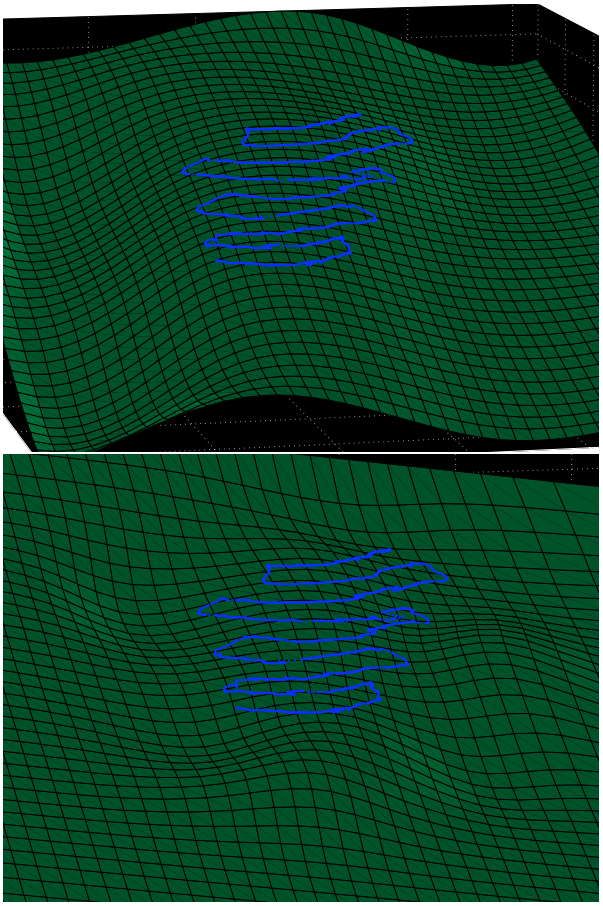

Figure 9: Kalman-updated spline surface of Clam Field South site using DVL range measurements from *a priori* data (top) and from online terrain measurements (bottom). The blue line denotes the ROV trajectory

After each Kalman update of the spline surface, the current vehicle position estimate is projected onto the spline surface in order to derive the causal orientation command, as depicted in Figure 11. The resulting orientation command is compared against the ROV's orientation during the maneuver and the DVL-based orientation command calculated from local terrain measurements.

Figures 12 and 13 compare the causal spline-based heading command with the pilot-chosen and DVL-based heading commands. The figures show that the spline map yields a smooth orientation command, as the command does not change by more than a few degrees every 5 seconds. On the other hand, the DVL-based heading command has regions of high frequency heading variation, where changes of 30 degrees in a 5 second interval are common. Thus, the spline-based orientation command generation produces a smoothed version of the DVL heading command.

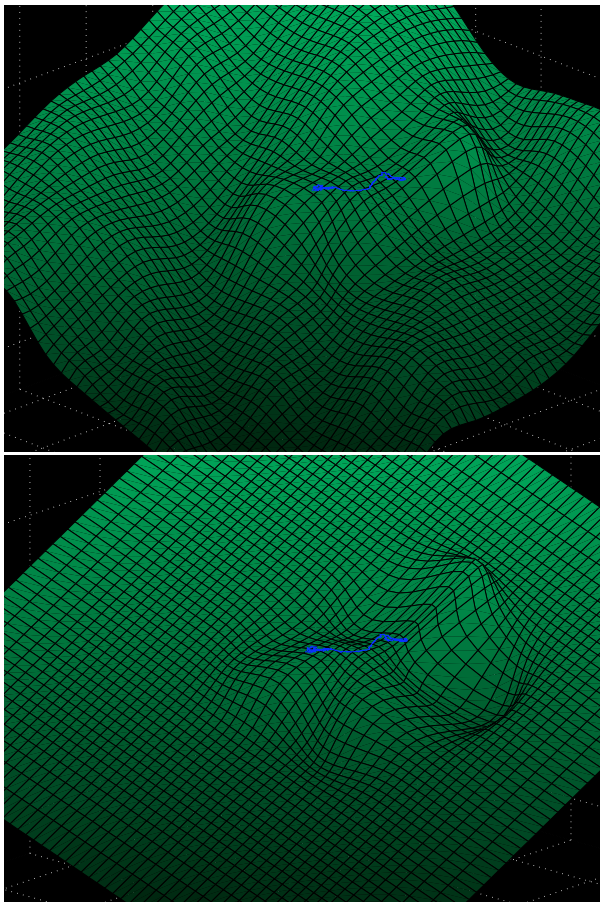

Figure 10: Kalman-updated spline surface of Soquel #4 site using DVL range measurements from *a priori* data (top) and from online terrain measurements (bottom). The blue line denotes the ROV trajectory

4.4 Real-Time Results

Figure 14 shows the real-time algorithm-generated orientation command being tracked by an automatic heading control loop. The comparison of real-time spline-generated and DVL-generated heading commands yields the same conclusion as the post-processing results comparison: the spline heading command is a much smoother version of the DVL heading command, enabling smooth ROV heading changes during terrain traversal.

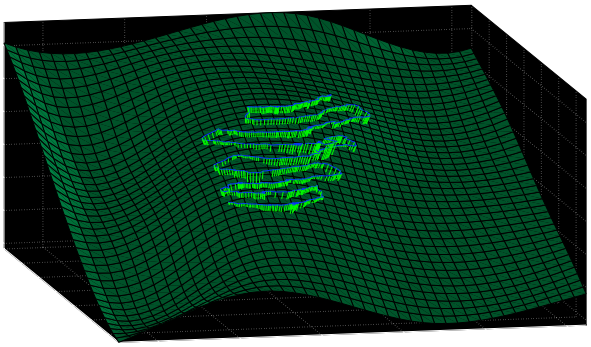

Figure 11: Kalman-updated spline surface with vehicle position estimates (blue) projected onto spline surface (green).

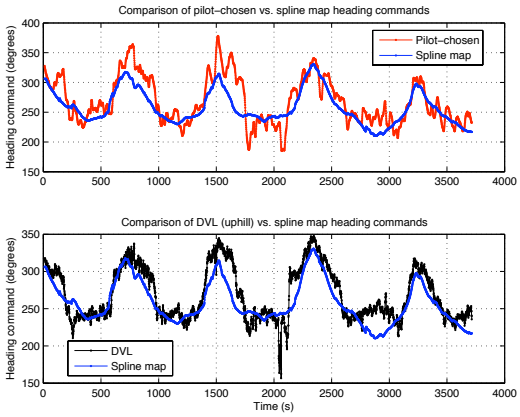

Figure 12: Clam Field South causal heading commands computed in post-processing from *a priori* data-based spline surfaces vs. DVL-chosen and pilot-chosen heading commands.

Using the images and metadata collected during real-time algorithm execution, a strip of images from the non-planar Soquel #4 site has been plotted in Figure 15, showing that one contour of the terrain has been surveyed under automatic heading control.

5 Summary

This paper has shown that a spline surface fit of DVL measurements successfully generates a smooth camera orientation command that keeps a visual survey camera perpendicular to the terrain. Acting like a spatial filter of terrain measurements, the spline fit is performed in real-time by a linear Kalman filter that updates the spline's shape

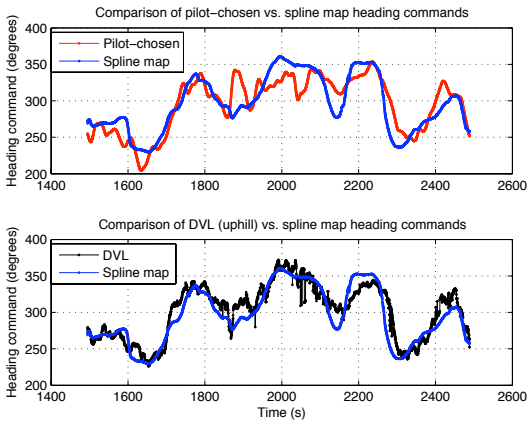

Figure 13: Soquel #4 causal heading commands computed in post-processing from online measurement-based spline surfaces vs. DVL-chosen and pilot-chosen heading commands.

and spline-relative vehicle position error estimate. Subsequently, the spline point projection algorithm derives the orientation command from the normal vector at the vehicle position's spline-projected point. The resulting camera orientation command has been tracked by an orientation control loop in order to keep the camera perpendicular to the terrain in a small-scale visual survey.

6 Acknowledgements

The authors thank the Monterey Bay Aquarium Research Institute for supporting this research. Also, thanks to the pilots of the ROV Ventana (Craig Dawe, Knute Brekke, DJ Osborne, and Mike Burczynski) and the crew of the *R/V Point Lobos* for helping to collect the data that made this research possible.

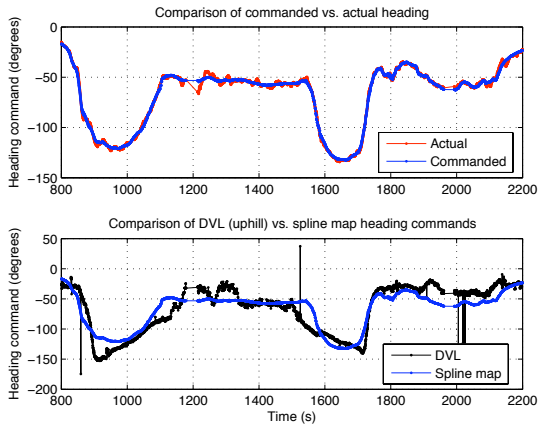

Figure 14: Smooth heading command generated in real-time and used for closed-loop heading control. The top plot shows that the heading control loop accurately tracks the spline-based commanded heading, and the bottom plot compares DVL and spline-generated heading commands.

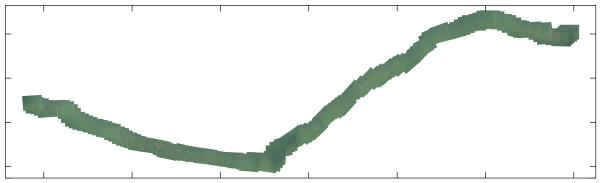

Figure 15: Sequence of overlapping images taken at Soquel #4 site using automatic control and heading command generated by algorithm.

References

[1] H. Singh, J. Howland, and O. Pizarro, "Advances in large-area photomosaicking underwater," *Oceanic Engineering, IEEE Journal of*, vol. 29, no. 3, pp. 872–886, July 2004.

[2] K. Richmond and S. Rock, "An Operational Real-Time Large-Scale Visual Mosaicking and Navigation System," in *Proceedings of the MTS/IEEE OCEANS Conference*. Boston, MA: IEEE, September 2006.

[3] S. Negahdaripour and X. Xu, "Mosaic-based positioning and improved motion-estimation methods for automatic navigation of submersible vehicles," *Oceanic Engineering, IEEE Journal of*, vol. 27, no. 1, pp. 79–99, Jan 2002.

[4] N. Gracias and J. Santos-Victor, "Underwater video mosaics as visual navigation maps," *Computer Vision and Image Understanding*, vol. 79, no. 1, pp. 66–91, 2000.

[5] R. M. Eustice, "Toward real-time visually augmented navigation for autonomous search and inspection of ship hulls and port facilities," in *Intl. Symposium on Technology and the Mine Problem*. Monterey, CA: Mine Warfare Association (MINWARA), 2008.

[6] M. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (slam) problem," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 3, pp. 229–241, Jun 2001.

[7] N. Fairfield, G. A. Kantor, and D. Wettergreen, "Real-time slam with octree evidence grids for exploration in underwater tunnels," *Journal of Field Robotics*, 2007.

[8] S. Thompson and S. Kagami, "Stereo vision terrain modeling for non-planar mobile robot mapping and navigation," in *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, vol. 6, Oct. 2004, pp. 5392–5397 vol.6.

[9] Y.-F. Wang, "New method for sensor data fusion in machine vision," B. C. Vemuri, Ed., vol. 1570, no. 1. SPIE, 1991, pp. 31–42.

[10] Y.-F. Wang, Z. Yang, and J.-F. Lee, "On 3d model construction by fusing heterogeneous sensor data," *lEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, pp. 1071–1078, Jul 1992.

[11] Y. Huang and X. Qian, "Dynamic b-spline surface reconstruction: Closing the sensing-and-modeling loop in 3d digitization," *Computer-Aided Design*, vol. 39, no. 11, pp. 987 – 1002, 2007.

[12] L. Piegl and W. Tiller, *The NURBS book (2nd ed.)*. New York, NY, USA: Springer-Verlag New York, Inc., 1997.

[13] C. De Boor, *A Practical Guide to Splines*. Springer, 2001.

[14] S. Augenstein and S. Rock, "Estimating Inertial Position and Current in the Midwater," in *Proceedings of the OCEANS 2008 MTS/IEEE QUEBEC Conference*, Quebec City, Canada, September 2008.

[15] S. D. Fleischer, "Bounded-Error Vision-Based Navigation of Autonomous Underwater Vehicles," Ph.D. dissertation, Stanford University, Stanford, CA 94305, May 2000.