Virtual Environment Modeling by Integrated Optical and Acoustic Sensing *

A. Fusiello, R. Giannitrapani, V. Isaia
Dipartimento di Matematica e Informatica
University of Udine, Italy
{fusiello,gtrapani,isaia}@dimi.uniud.it

V. Murino
Dipartimento Scientifico e Tecnologico
University of Verona, Italy
murino@sci.univr.it

Abstract

In this paper, the problem of underwater scene understanding from multisensory data is addressed. Acoustic and optical devices onboard an underwater vehicle are used to sense the environment in order to produce an output which is readily understandable even by an inexperienced operator. The main idea is to integrate multiple sensory data by geometrically registering data to a model. In this way, vehicle pose is derived, and the model objects can be superimposed on actual images, generating an augmented reality representation. Results on a real underwater scene are provided, showing the effectiveness of the proposed approach.

1. Introduction

Bad structured environments, like the underwater world, are difficult to perceive and to understand. In these cases, the integration of different sensory data is critical. This paper is devoted to the recognition and the synthetic reconstruction of an underwater environment, in order to support a human operator of a Remotely Operated Vehicle (ROV). Objects of interest are recognized and their three-dimensional (3-D) synthetic models are displayed on the real image to generate an augmented reality representation, which improves the perception and understanding of the surrounding environment.

Two sensing channels, optical and acoustic, are mostly used underwater. Typically, optical images are easier to interpret by a human operator, but underwater visibility range is very limited due to low illumination and clutter presence. On the other hand, acoustic 3-D data are not affected by illumination problems but are more difficult to understand for a human operator. From these considerations, it appears sensible to try to integrate the two channels in order to exploit the best of both, so as to compensate their lacks.

To the best of our knowledge, our approach to sensor integration and data fusion is novel and no similar works are present in the literature. Nevertheless, fusion and integration of different kinds of data is actually a matter of active research [16, 4].

Concerning the usefulness to integrate different sensor modalities and algorithms, a few works addressed such challenging issue [12, 19, 1]. More specifically on the joint use of optical and 3-D information, some interesting papers can be considered concerning the fusion of intensity and range data, the latter mainly derived by a laser range finder [12, 10, 25, 24].

In our work, acoustic and optical underwater data are processed separately in order to recognize the objects present in the scene and estimate their position. In this way the relative position of acoustic and optical cameras is estimated on-line, and data integration is achieved. We propose a system composed by a set of processing modules of acoustic and optical data which already contain *per se* some novel aspects and solutions. However, the original issue consists in the development of a system able to integrate different sensor modalities and fuse different kinds of data in *numerical* form. Our goal is to locate model objects present in a cluttered scene and facilitate human interpretation by displaying such objects on the real images in the correct position and orientation.

The rest of the paper is organized as follows. After an overview of the global system in Sec. 2, the acoustic sensing and related data processing is described in Sec. 3. In Sec. 4, a related description of the optical sensory channel is reported. The integration phase is described in Sec. 5. In Sec. 6, results on real data are reported showing the goodness of the method.

*This work is supported by the European Commission under the BRITE-EURAM III project no. BE-2013 VENICE (Virtual Environment Interface by Sensor Integration for Inspection and Manipulation Control in Multifunctional Underwater Vehicles)

2. System Overview

The application scenario consists in an ROV approaching an oil rig whose geometrical model is given in a descriptive language (e.g., Virtual Reality Modeling Language, VRML). The ROV is equipped with an optical and an acoustical camera located at a fixed but unknown relative position. The optical camera provides 2-D intensity images, whereas the acoustical one provides an image consisting of a set of 3-D points [18]. These images are not registered and they are only partially overlapping, as the points of view and the view frusta are different.
The model of the oil rig is composed by connected pipes. The goal of the system is to identify and locate joints with respect to the ROV, thereby obtaining the position of the ROV in the 3-D model reference system.
The system is subdivided in two data processing schemes related to the different sensory channels and in an integration phase. Each single scheme is composed by several modules devoted to object recognition and pose estimation. The pose of both sensors is computed with respect to the fixed observed object. In this way, the relative sensor pose can be computed and images can be registered (integration). Even if our algorithm has been tested on real scenes of tubular objects variously connected, this does not prevent the generality of the approach and its utility in other contexts. In fact, the specific methods adopted in the various phases for either acoustic or optical data processing (e.g., 3-D skeleton extraction, 2-D edge detection) are able to deal with different kinds of objects.

3. Acoustic sensing

In this section, we describe the processing of three-dimensional data obtained from the acoustic camera, in order to register the sensed data with the model.

3.1 Acoustic Camera

Three-dimensional data are obtained by a high resolution acoustic camera, the *Echoscope* [11]. The acoustic camera is formed by a two-dimensional array of transducers sensible to signals backscattered from the scene previously insonified by a high-frequency acoustic pulse. The whole set of raw signals is then processed in order to estimate signals coming from fixed steering directions (called beamsignals) and to attenuate those coming from other directions. Assuming that the beamsignals represent the responses of a scene from a 2D set of (steering) directions, a 3-D point set can be extracted detecting the time instant (t^*) at which the maximum peak occurs in each beamsignal. Besides, the intensity of the maximum peak can be used to generate another image, registered with the former, representing the reliability of the associated 3-D measures. In other words, the higher the intensity, the safer the 3-D measure associated. Images are formed by 64×64 3-D points ordered according to an angular relation, as adjacent points correspond to adjacent beamsignals. Their coordinates are expressed in a 3-D reference frame attached to the sensor.

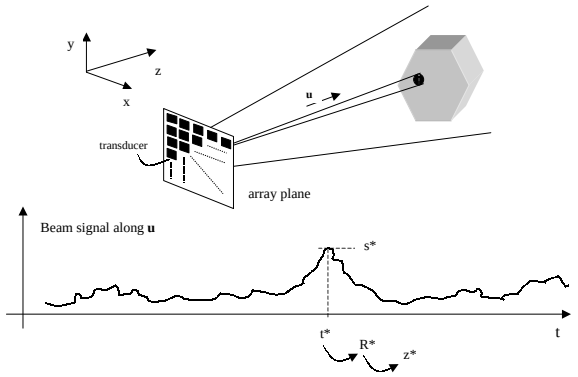

Figure 1. Acoustic camera.

3.2 Filtering and Segmentation

The raw images obtained by an acoustic camera are typically quite noisy mainly because of the environment conditions and of the intrinsic characteristics of the camera. Although the used acoustic camera directly performs a preliminary low level processing, it has been proved useful to filter 3-D data with a suitable algorithm. In particular, in the first step, connected components in the image are computed, where two points are considered connected if they are adjacent in the 64×64 angular relations matrix and if their Euclidean distance is below a fixed threshold, depending on the spatial resolution of the camera. In such a way, it is possible to subdivide the image in a certain number of connected components, while discarding those components formed by a small number of points, likely not representing interesting physical objects. In the second step, "reliable" connected components are formed by the points whose associated intensity is above a certain threshold, still depending on the camera properties.
After this preprocessing phase it is necessary to segment the image, i.e. to subdivide the set of 3-D points in distinct regions that are pipes' candidates. To this purpose, the skeleton [21, 2] is first extracted, and then it is used to subdivide the image in different convex components.
To extract the skeleton, we apply to the image the following procedure [17]: for every point $\mathbf{x}$ we consider all the points that are in a sphere of radius r centered on it. Then, we shift $\mathbf{x}$ from its actual position to the centroid of such distribution of points. We apply this procedure in a parallel way on all the points of the image. In other words, for every point

$\mathbf{x} = (x, y, z)$, we define a new point $\tilde{\mathbf{x}}$ in the following way:

$$\tilde{\mathbf{x}} = \sum_{\mathbf{y}_j \in U} \frac{\mathbf{y}_j}{dim(U)} \tag{1}$$

where

$$U = \{\mathbf{y}_j : (\mathbf{y}_j - \mathbf{x})^2 \leq r\} \tag{2}$$

and $\dim(U)$ is the number of points in U.
The overall effect of this transformation is to shift points on the border towards the center, while leaving points well inside an object unaltered. The iterative application of such a procedure tends to shift all the points of the distribution towards the skeleton.
Then, skeleton points are labeled as branch or joint points, still by exploiting the properties of inertial tensors. Since we are interested in the segmentation of data in tubular structures, it is natural to choose the branches of the obtained skeleton, or more exactly the original 3-D points collapsed on each branch, as pipes' candidates.
At the moment the operator is required to choose the parameter r and the number of iterative steps, however an adaptive algorithm to automatize this procedure is under investigation.

3.3 Classification and Geometric reconstruction

The segmentation phase provides a certain number of clusters of 3-D points that have to be classified as pipe-like or non pipe-like ones; for this task we used a technique, related to the so called Principal Component Analysis, based on the Inertial Tensor. Given a discrete distribution of N points $\{\mathbf{x}_i\}_{i=1\ldots N}$, the inertial tensor is the 3×3 matrix defined as

$$\mathbf{I} = \sum_i (\mathbf{x}_i - \mathbf{o}) \sqcap (\mathbf{x}_i - \mathbf{o}) \tag{3}$$

where $\mathbf{o}$ is the centroid and the symbol $\sqcap$ represents the following operator:

$$\mathbf{a} \sqcap \mathbf{b} = \begin{bmatrix} (a_y b_y + a_z b_z) & -a_x b_y & -a_x b_z \\ -a_y b_x & (a_x b_x + a_z b_z) & -a_y b_z \\ -a_z b_x & -a_z b_y & (a_x b_x + a_y b_y) \end{bmatrix}. \tag{4}$$

The eigenvalues and eigenvectors of $\mathbf{I}$ are then employed to extract useful information on the shape of the discrete distribution. Let $\alpha_1 \leq \alpha_2 \leq \alpha_3$ be the eigenvectors of $\mathbf{I}$. If

$$\alpha_1 \ll \alpha_2 \text{ and } \alpha_1 \ll \alpha_3 \text{ and } \alpha_2 \simeq \alpha_3 \tag{5}$$

then the region is classified as a pipe, otherwise it is discarded. To check these relations a threshold on the ratios α_2/α_1 and α_3/α_1 is introduced, and, if it is too small, it is probable to classify as a pipe something that is only elongated, whereas, if it is too high, it is probable to loose some pipes from the scene.
From the value of the minimum eigenvalue it is possible to roughly estimate the radius of the tubular region. In fact, in the case of a complete cylindrical distribution the following relation holds:

$$\alpha_1 = \frac{1}{2} N r^2 \tag{6}$$

where N is the total number of points in the distribution and r is the radius. Unfortunately, in acoustic images, points are not distributed on the surface of a cylinder, but only on a little portion of it. Moreover, they are so noisy that they carry little information on the curvature. Hence, relation (6) is only an approximation, but it is sufficient to give an order of magnitude for the radius, as we will see in the section of the experimental results. Finally, it is possible to determine the approximate position of the pipe axis, whose direction is given by the eigenvector relative to α_1, by translating the centroid of the distribution in the direction of the eigenvector corresponding to α_3, that is the direction of minimum spread of the 3-D cloud.
Even if pipes form a joint in the scene, their estimated axes may not intersect exactly in one point. To extract an approximate intersection for the pipes we use the following simple algorithm: for every axis pair (i, j), we define their intersection as the midpoint $\mathbf{m}_{ij}$ of the unique segment that connect the two lines defined by the axis and that is perpendicular to both of them. If the number of axes is n, the number of possible pairs is $n(n-1)/2$. We define the joint of the pipes as the centroid of these midpoints, i.e.

$$\frac{\sum_{j=1}^{n-1} \sum_{i=j+1}^{n} \mathbf{m}_{ij}}{n(n-1)/2} \tag{7}$$

This method is straightforward if there is only one joint in the scene; if this is not the case, it is necessary to preliminary subdivide the set of extracted pipes into subsets containing pipes that belong to the same joint. To do this, it is sufficient to group pipes whose distance, defined as the distance between the lines passing through the axis, is below a threshold that depends on the radius of the pipes. This can be done, by building the *Incidence Graph* IG of the pipes, i.e., a graph whose nodes are the pipes and in which two nodes are connected if the distance between the corresponding pipes is below the given threshold. The sought joints correspond to the maximal complete subgraphs of IG, i.e, subgraphs that are complete and that are not contained in any larger complete subgraph. Two maximal subgraphs can have no more than one node in common (corresponding to the pipe that connects two joints).

To summarize, the skeleton segmentation and the subsequent analysis with the inertial tensor is able to recognize most of the pipes present in the observed scene and to reconstruct in a rough way their geometrical properties. Although some pipes may be lost in this phase, a partial reconstruction is sufficient for the subsequent matching and alignment.

3.4 Model-view registration

Acoustic data points which lie on the surface of cylinders are expressed in the acoustic reference frame, whereas the underlying object surface model is placed in the model reference frame. The unknown rigid transformation that links the two reference frames is obtained by registering model and data.
In their paper, Besl and McKay [3] describe a general purpose method for the registration of rigid 3-D shapes which they refer to as the Iterative Closest Point (ICP) algorithm. This approach eliminates the need to perform any feature extraction, or to specify feature correspondence.
The ICP algorithm is only guaranteed to converge to a local minimum, which may not correspond to the global minimum. In our case, pre-alignment is obtained by matching the segmented data with the model, which produces a fairly good initial alignment, sufficient to achieve global convergence.

3.4.1 Pre-alignment

In the previous sections we showed how to extract the relevant geometrical properties for the joints in the observed scene. Such properties are used to match these joints with the ones stored in the VRML model. In particular, we use the angles between pipes as the recognition feature: two joints match if such angles are equal within a certain error. Since the joints analyzed are composed by a low number of pipes, the correspondence is actually performed with an exhaustive method, although the use of more sophisticated algorithms, based on Interpretation Trees [9], are under investigation.
After a matching with the database has been found, it is possible to obtain the raw pose (i.e., position and orientation) of the acoustical camera with respect to the model coordinate system by computing the rigid transformation aligning the estimated pipe axes and the model pipe axes.

3.4.2 ICP algorithm

Let us suppose that we have two sets of 3-D points which correspond to a single shape but are expressed in different reference frames. We will call one of these sets the model set X, and the other the data set Y. Assume that for each point in the data set, the corresponding point in the model set is known. The problem is to find a 3-D transformation which, when applied to the data set Y, minimizes a distance measure between the two point sets. The goal of this problem can be stated more formally as follows:

$$\min_{\mathbf{R},\mathbf{t}} \sum_{i=1}^{N} \|\mathbf{x}_i - (\mathbf{R}\mathbf{y}_i + \mathbf{t})\|^2, \tag{8}$$

where $\mathbf{R}$ is a 3×3 rotation matrix, $\mathbf{t}$ is a 3×1 translation vector, and the subscript i refers to corresponding elements of the sets X and Y. Efficient, non-iterative solutions to this problem were compared in [14], and the one based on Singular Value Decomposition (SVD) was found to be the best. It can be easily seen that (8) is equivalent to

$$\min_{\mathbf{R}} \sum_{i=1}^{N} \|\bar{\mathbf{x}}_i - \mathbf{R}\bar{\mathbf{y}}_i\|^2, \tag{9}$$

where $\bar{\mathbf{x}}_i$ and $\bar{\mathbf{y}}_i$ are the centralized point obtained by subtracting the respective centroids. Equation (9) is minimized when $\mathrm{trace}(\mathbf{R}\mathbf{K})$ is maximized [13], where

$$\mathbf{K} = \sum_{i=1}^{N} \bar{\mathbf{x}}_i \bar{\mathbf{y}}_i^{\top}.$$

If the SVD of $\mathbf{K}$ is given by $\mathbf{K} = \mathbf{V}\mathbf{D}\mathbf{U}^{\top}$, then the optimal rotation matrix is $\mathbf{R} = \mathbf{V}\mathbf{U}^{\top}$. The optimal translation is then computed as the difference between the centroid of X and the centroid of the rotated Y set.
The general 3-D registration problem that ICP addresses, differs from the corresponding point set registration problem in two important regards. First, the point correspondence is unknown. Second, 3-D shapes to be registered are not necessarily represented as point sets.
For each point $\mathbf{y}_i$ from the set Y, there exists at least one point on the surface of X which is closer to $\mathbf{y}_i$ than all other points in X. This is the *closest point*, $\mathbf{x}_i$. The basic idea behind the ICP algorithm is that, under certain conditions, the point correspondence provided by sets of closest points is a reasonable approximation to the true point correspondence. Besl and McKay proved that if the process of finding closest point sets and then solving equation (8) is repeated, the solution is guaranteed to converge to a local minimum. The ICP algorithm can now be stated:

1. For each point in Y, compute the closest point in X;
2. With the correspondence from step 1, compute the incremental transformation $(\mathbf{R}, \mathbf{t})$ with SVD;
3. Apply the incremental transformation from step 2 to the data Y;
4. Compute the change in total mean square error. If the change in error is less than a threshold, terminate. Else goto step 1.

In this way we obtain the rigid transformation that brings this reference frame onto the model reference frame, given by a 4×4 homogeneous matrix $\mathbf{G}_a$.

4. Optical sensing

In this section, we describe the processing of the optical data in order to perform *registration*, that is solving for the camera pose that best fit a model to some matching image features.
Since the model is a tubular rig, the relevant image features are the segments forming the bounding contours of the pipes[1].

4.1 Camera model

The optical device is modeled by the *pinhole camera*, which is given by its *optical center* C and its *retinal plane* (or *image plane*) $\mathcal{R}$. A 3-D point W is projected into an image point M given by the intersection of $\mathcal{R}$ with the line containing C and W (Figure 2). The line containing C and orthogonal to $\mathcal{R}$ is called the *optical axis* (the Z axis in Figure 2) and its intersection with $\mathcal{R}$ is the *principal point*. The distance between C and $\mathcal{R}$ is the *focal distance* (note that, since in this model C is behind $\mathcal{R}$, real cameras will have negative focal distance).

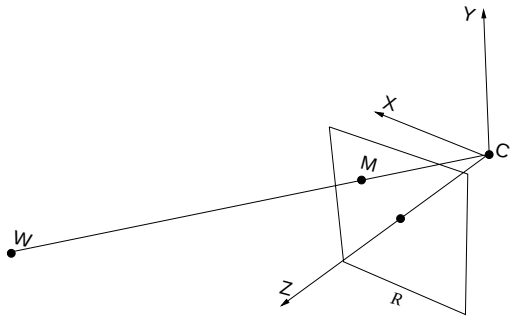

Figure 2. The pinhole camera model, with the *camera reference frame* **(**X,Y,Z**) depicted.**

Let $\mathbf{w} = [x\ y\ z]^\top$ be the coordinates of W in the *model reference frame* and $\mathbf{m} = [u\ v]^\top$ the coordinates of M in the image plane (pixels). The mapping from 3-D coordinates to 2-D coordinates is the *perspective projection*, which is represented by a linear transformation in *homogeneous coordinates*. Let $\tilde{\mathbf{m}} = [u\ v\ 1]^\top$ and $\tilde{\mathbf{w}} = [x\ y\ z\ 1]^\top$ be the homogeneous coordinates of M and W respectively; then, the perspective transformation is given by the 3×4 matrix $\tilde{\mathbf{P}}$:

$$\lambda\tilde{\mathbf{m}} = \tilde{\mathbf{P}}\tilde{\mathbf{w}}, \tag{10}$$

where λ is an arbitrary scale factor. The camera is therefore modeled by its *perspective projection matrix* (henceforth PPM) $\tilde{\mathbf{P}}$, which can be decomposed, using the QR factorization, into the product

$$\tilde{\mathbf{P}} = \mathbf{A}[\mathbf{I}|\mathbf{0}]\mathbf{G}_o. \tag{11}$$

The 3×3 matrix $\mathbf{A}$ depends on the *intrinsic parameters* only: focal length in pixels, aspect ratio, principal point and skew factor. The camera position and orientation (pose) are encoded by the 4×4 matrix $\mathbf{G}_o$ representing the rigid transformation that brings the camera reference frame onto the model reference frame. $\mathbf{R}$ is the 3×3 rotation matrix and $\mathbf{t}$ is the 3×1 translation vector.
We seek the matrix $\mathbf{G}_o$, assuming that the *constant* intrinsic parameters have been computed by off-line by a calibration procedure [22].

4.2 Lines grouping

Underwater images have a very low signal to noise ratio, because of the low illumination and bad environmental conditions. In order to filter the noise without affecting the signal, we use the Perona-Malik [20] anisotropic smoothing filter, which preserves the information about the object contours. Basically, it is a Gaussian smoothing with a standard deviation depending on the grey levels gradient.
Straight lines are extracted by combining Canny [6] edge detector and Burn's *Plane Fit Algorithm* [5]. First, edge points are extracted with the Canny edge detector, that allows to find very sharp edges (often one pixel large) thanks to the non-maxima suppression. Then, pixels are clustered in support regions if they are spatially adjacent and if their gradient orientation is roughly the same. The line parameters are computed with plane intersections of the weighted fit to the intensity values and the horizontal average pixel intensity plane, within a support region. The weight favours intensity values of pixels with high gradient magnitude. Taking primarily the gradient orientation as evidence for a line and using the plane fit method, the algorithm actually extracts long, straight lines as well as shorter lines and is effective in finding low-contrast lines.
Each extracted segment is then labeled, and its attributes are computed. In order to find pipes in the image, pairs of segments are grouped together, which are likely the projection of the boundaries of a pipe (not every segments pair is the projection of a pipe). Grouping is based on proximity and *covering* criteria: two segments are paired if their projections onto their median axis overlap by more than 60% and the distance between their midpoints is less than a threshold (that is related to the expected distance of pipes boundaries in the image).

[1] Given a viewpoint, the *rim* of an object is the set of all the points on the object surface where the line joining the viewpoint (optical ray) is tangent (assuming perspective projection). The projection of the rim is the *bounding contour* of the object in the image.

4.3 Model-view registration

Optical alignment is performed using an algorithm due to Lowe [15] that finds the camera pose yielding the best matching between each image segment and the projection of its corresponding cylinder rim. The algorithm assumes that image-model correspondences are given. In our case the initial pose for the optical camera is assumed to be the same of the acoustic one ($\mathbf{G}_a$), already computed. This allows to project the model accordingly, and model segments are matched against the image segments using an algorithm introduced by Scott and Longuet-Higgins [23] for associating features of two arbitrary patterns.
Were the approximate camera pose unknown, a more complex recognition algorithm should be used [7].

4.3.1 The Scott and Longuet-Higgins' algorithm

Scott and Longuet-Higgins [23] proposed an algorithm based on the singular value decomposition (SVD) for associating features of two images. The algorithm incorporates both the principle of proximity and the principle of exclusion.
Let I and J be two images, containing m features I_i and n features J_j, respectively, which we want to put in one-to-one correspondence. The algorithm consists of three stages. The first stage is to build a *proximity matrix* $\mathbf{G}$ of the two sets of features with elements

$$G_{ij} = e^{-r_{ij}^2/2\sigma^2} \tag{12}$$

where r_{ij} is a well defined distance between feature I_i and J_j and σ is an appropriate unit of distance, that controls the scale of interaction. The next stage is to perform the SVD of $\mathbf{G}$

$$\mathbf{G} = \mathbf{U}\mathbf{S}\mathbf{V}^\top \tag{13}$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal and $\mathbf{S}$ is a non-negative $m \times n$ diagonal matrix.
Finally, $\mathbf{S}$ is converted into a new $m \times n$ matrix $\mathbf{D}$ by replacing every diagonal element S_{ii} with 1 and obtain another matrix

$$\mathbf{P} = \mathbf{U}\mathbf{D}\mathbf{V}^\top \tag{14}$$

of the same shape of the original proximity matrix and whose rows are mutually orthogonal. The element P_{ij} indicates the extent of pairing between feature I_i and feature J_j. If P_{ij} is both the greatest element in its row and the greatest element in its column, then we regard those two different features I_i and J_j as being in correspondence with one another.
This matrix incorporates the principle of proximity by construction of $\mathbf{G}$, and the principle of exclusion by virtue of its orthogonality.

In our application, elements to be matched are lines, expressed in the normal form:

$$u\cos\alpha_i + v\sin\alpha_i - d_i = 0. \tag{15}$$

As a distance between model lines and image lines we used the following

$$r_{ij} = \left\| \left[\cos\alpha_i, \sin\alpha_i, \frac{2d_i}{\max_l d_l}\right] - \left[\cos\alpha_j, \sin\alpha_j, \frac{2d_j}{\max_l d_l}\right] \right\| \tag{16}$$

The first two components are bounded in the interval $[-1, 1]$, whereas the third belongs to $[0, 2]$. Since the initial pose is quite close to the true one, this simple matching is sufficient.

4.3.2 Lowe's algorithm

Let us suppose that *point correspondences* are available and that the intrinsic camera parameters are known. Let $\mathbf{w}_1 \ldots \mathbf{w}_N$ be N points of an object model expressed in the model reference frame and $\mathbf{m}_1 \ldots \mathbf{m}_N$ be the image points, projections of the $\mathbf{w}_i$. The relation between an object point and an image point is given by the perspective projection:

$$\kappa\mathbf{A}^{-1}\tilde{\mathbf{m}}_i = [\mathbf{R}|\mathbf{t}]\tilde{\mathbf{w}}_i. \tag{17}$$

derived from (11) by putting $\mathbf{G}_o = \begin{bmatrix}\mathbf{R} & \mathbf{t}\\ \mathbf{0} & 1\end{bmatrix}$. Let $\tilde{\mathbf{p}}_i = [u_i, v_i, 1]^\top = \mathbf{A}^{-1}\tilde{\mathbf{m}}_i$ be the *normalized image coordinates*. Expanding, we see that each point correspondence generates two equations,

$$\begin{cases} u_i = \dfrac{\mathbf{r}_1^\top \mathbf{w}_i + t_1}{\mathbf{r}_3^\top \mathbf{w}_i + t_3} \\ v_i = \dfrac{\mathbf{r}_2^\top \mathbf{w}_i + t_2}{\mathbf{r}_3^\top \mathbf{w}_i + t_3}. \end{cases} \tag{18}$$

where $\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3]^\top$ and $\mathbf{t} = [t_1, t_2, t_3]^\top$. The 12 unknown components of $\mathbf{R}$ and $\mathbf{t}$ can be determined from a sufficient number of points correspondences solving a linear system. The resulting $\mathbf{R}$, however, is not guaranteed to be orthogonal. To explicitly enforce orthogonality, $\mathbf{R}$ must be parameterized with the three Eulerian angles ϕ, ψ, θ, ending up with a nonlinear system of six unknown. Let's think of (18) as a mapping $\mathbf{F}_i : R^6 \to R^2$ from the six parameters space to the image coordinates u_i, v_i. Then (18) is equivalent to

$$\mathbf{p}_i = \mathbf{F}_i(\mathbf{e}). \tag{19}$$

where $\mathbf{e} = [\mathbf{t}, \phi, \psi, \theta]^\top$. This is solved by the following Newton iteration: $\mathbf{e} \to \mathbf{e} - \Delta\mathbf{e}$ where $\Delta\mathbf{e}$ is the solution of the following linear system of equations:

$$\mathbf{p}_i - \mathbf{F}_i(\mathbf{e}) = \mathbf{J}_i\ \Delta\mathbf{e} \tag{20}$$

where $\mathbf{J}_i = \partial \mathbf{r}_i / \partial \mathbf{e}$ is the Jacobian of the residual $\mathbf{r}_i = \mathbf{p}_i - \mathbf{F}_i(\mathbf{e})$. The six unknowns $\Delta\mathbf{t}, \Delta\phi, \Delta\psi, \Delta\theta$ can be determined if at least three points correspondences are known. However, to counteract the effect of inaccurate measurements or correspondences, as many correspondences as possible are typically use.

Newton's method starts off with an initial guess for $\mathbf{t}, \phi, \psi, \theta$, and, for each point $\mathbf{w}_i \quad i = 1 \ldots N$, computes the location of $\mathbf{p}_i$ through (18). The method proceeds by computing new estimates for the rotation matrix and translation vector, and iterating the procedure until the norm of the residuals $\mathbf{r}_i$ becomes small enough. In this way, a least-squares minimization is performed.

$\mathbf{F}_i$ is linear with respect to translation and scaling over the image plane, and approximately linear over a wide range of values of the rotational parameters. Hence, the method is likely to converge to the desired solution for a rather wide range of possible starting positions. Given the small displacement between the two and the negligible rotation, this is usually sufficient to ensure convergence.

The method can be easily extended to cope with *line correspondences*. Let us write the equation of a line in the image in the normal form

$$u \cos\alpha + v \sin\alpha - d = 0. \tag{21}$$

Now, given a set of pairs of corresponding image and model lines, we choose two points on each model line and compute the distance between each projected point. Let $[u(\mathbf{e}), v(\mathbf{e})]^\top$ be one of the projected points, then the residual is the signed distance of the point from the matching line:

$$r = u(\mathbf{e}) \cos\alpha + v(\mathbf{e}) \sin\alpha - d \tag{22}$$

The derivatives of r, needed in the Newton iteration, are simply

$$\frac{\partial r}{\partial \mathbf{e}} = \frac{\partial u}{\partial \mathbf{e}} \cos\alpha + \frac{\partial v}{\partial \mathbf{e}} \sin\alpha, \tag{23}$$

that is a linear combination with weights $\sin\alpha$ and $\cos\alpha$ of the partial derivatives that compose the Jacobian found for point correspondences. Since each point gives one equation for the correction parameters, and two points are sufficient to uniquely identify the model line, a line-to-line correspondence yields the same information (two equations) of a point-to-point correspondence, and the structure of algorithm remains unchanged.

The case of a smooth boundaries objects, as cylinders, is different. A rim generated by a sharp edge is stable on the object as long as the edge is visible, whereas a rim generated by a smooth surface changes continuously with the viewpoint. In our case, the rim is a line in space whose position is function of the parameters $\mathbf{t}, \phi, \psi, \theta$. Hence, the expression of the Jacobian of the residuals becomes more complicated. However, as noted by Lowe [15], ignoring this dependence in the computation of the Jacobian, thereby treating the rim as a fixed line in space, does not prevent the algorithm to converge, and does not affect the precision of the final alignment.

5. Integration and Virtual modeling

Given a rig composed by an optical and an acoustic camera, and given an acoustic image, composed by a set of target points, each with a certain 3-D position, we want to project it onto the optical image plane, obtaining a depth map with reference to the image plane.

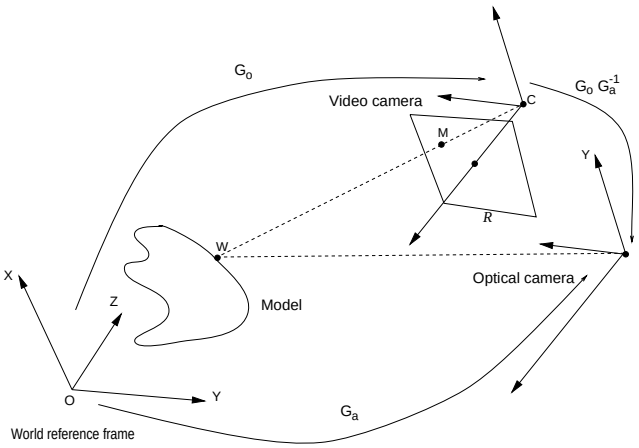

Figure 3. Optical/acoustic calibration

To this purpose, the relative pose of optical and acoustic cameras is needed. In principle, one should calibrate the cameras. A suitable object should be manufactured which gives raise to distinct features both in the acoustic image and optical image. This is very difficult to achieve, mainly because of the low resolution of acoustic device. In our approach, we use the scene itself as a calibration object. Knowing the CAD model of the observed objects, we register both acoustic and optical data to the model, thereby obtaining the relative pose of optical and acoustic cameras. Since this process is done on-line, better estimates can be obtained by integrating the measurements over time, using a Kalman filter [8].

Let $\mathbf{G}_o$ be the matrix representing the pose of the optical camera, obtained after optical alignment, as described in Sec. 4:

$$\tilde{\mathbf{w}}_{\mathrm{std}} = \mathbf{G}_o \tilde{\mathbf{w}}_{\mathrm{model}}, \tag{24}$$

and let $\mathbf{G}_a$ be the rigid transformation that brings the acoustic camera reference frame onto the model reference frame,

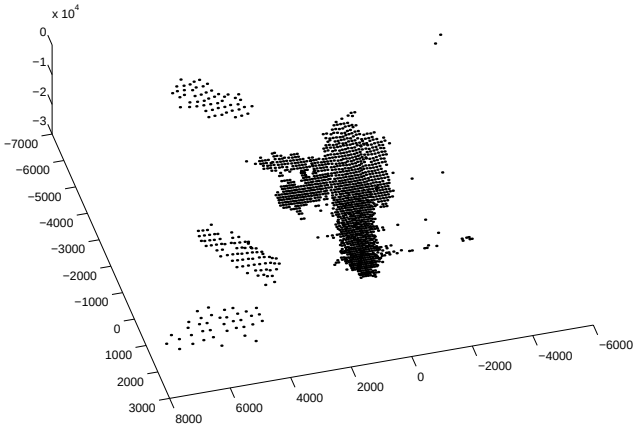

Figure 4. Raw acoustic data, as recorded by the Echoscope. The joint is clearly visible, but there are also spurious points.

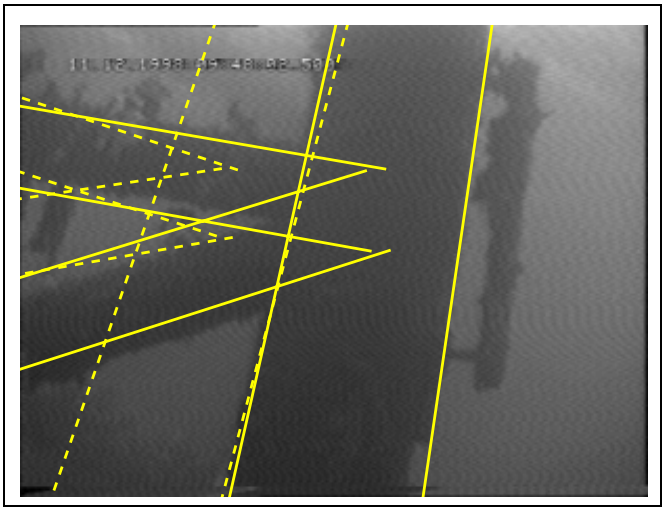

Figure 5. Optical image with superimposed the projected model according to the initial pose estimate (dashed lines) and to the final pose estimate (solid lines).

computed by 3-D alignment, as described Sec. 3:

$$\tilde{\mathbf{w}}_{\mathrm{sonar}} = \mathbf{G}_a \tilde{\mathbf{w}}_{\mathrm{model}}. \tag{25}$$

By composing the two transformations we get: $\tilde{\mathbf{w}}_{\mathrm{std}} = \mathbf{G}_o \mathbf{G}_a^{-1} \tilde{\mathbf{w}}_{\mathrm{sonar}}$. Hence, the PPM that projects the 3-D points expressed in the acoustic camera reference frame to the image plane of the optical camera is given by

$$\tilde{\mathbf{P}}_{\mathrm{oa}} = \mathbf{A}[\mathbf{I}|\mathbf{0}]\mathbf{G}_o \mathbf{G}_a^{-1}. \tag{26}$$

The intrinsic parameters matrix $\mathbf{A}$ is the same of the optical camera, and is obtained from a calibration procedure.
By projecting the 3-D points onto the image plane while keeping the third coordinate, which represents the distance of the point to the focal plane of the camera, we obtain a depth field defined at sparse locations. To obtain a proper depth map, first a surface mesh is generated by Delaunay triangulation in the image plane. The mesh may have several unwanted features upon creation, such as small, insignificant noise patches and jagged boundaries. Long edges and small unconnected surface patches are then removed. Moreover, since the acoustical data has been registered to the model, the points falling outside the pipes boundaries – because of the low spatial resolution of the acoustic device – are discarded. Finally, a uniformly sampled surface at a higher resolution than the original mesh data is obtained by interpolation and resampling over the image pixels grid. In this way, we obtain a depth map referred to the optical image.

Moreover, the accurate estimate of the position of the system relative to the environment is used in combination with the database information to provide a high quality, 3D graphics, virtual display of the environment. This scene can be viewed from any position and direction, including from the ROV itself, and as this virtual view is unaffected by turbidity, etc., it provides a clear and easily understood view of the complete working environment.

6. Results

In this section, we show one of the results obtained on a typical real case. A ROV equipped with a video camera and the Echoscope was used to take images of an underwater rig off Bergen (Norway). The video camera was calibrated underwater using a suitable calibration jig [22]. The lateral displacement between the two cameras was approximately 300 mm, and the view axes were approximately parallel. We didn't rely on this measurements though, for the relative pose of the cameras was obtained as explained in the previous sections.
Our procedure starts from the raw acoustic data (Fig. 4) and the image (visible in Fig. 5) of a scene consisting of three pipes of radii 500mm and 250mm, viewed from a distance of approximately 7.7m. The registration of 3-D data converged to a solution with a residual (RMS points-model distance) of 70 mm. The result of optical registration is shown in Fig. 5. The result of integration can be appreciated in

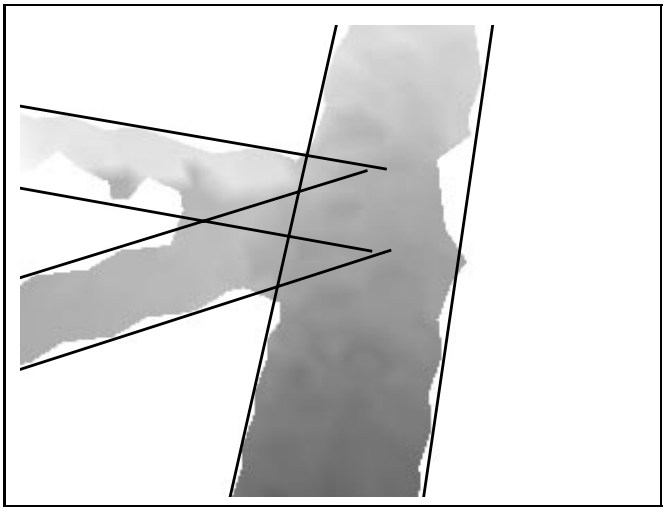

Figure 6. Acoustic depth map registered to the optical image. The gray level of a given pixel represents its depth (the darker the closer). Solid lines represents the model projected according to the camera pose estimate.

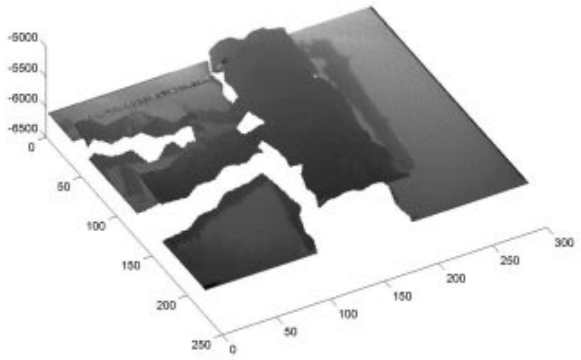

Figure 7. Surface interpolating the (processed) 3-D acoustic points, with the real image texture mapped onto it. An arbitrary background plane is also shown.

Fig. 6, which shows a depth image registered to the optical image, where the depth for each pixel is computed from the acoustic measures.
In Fig. 7, the same depth map is shown as a surface, with the original image texture-mapped onto it. Finally, in Fig. 8, a view of the synthetic VRML model with the 3-D points superimposed is shown. The point of view has been slightly changed to make visible two pipes that were occluded in the original image.

7. Conclusions

Guidance and inspection/maintenance/repair (IMR) tasks performed by ROVs are very difficult for several reasons. They require specialist crew, expensive training and many hours of practice. Output from the video camera is difficult to understand due to the 2-D nature of the images or bad environmental conditions leading to disorientation. This situation is not significantly improved if traditional acoustic sensors are used, as their output is not available in a form which is readily understandable even by a trained operator. The aim of the VENICE[2] project, within which this work has been carried out, is to overcome this difficulties.
This paper presents a system aimed at assisting an ROV pilot by presenting him an augmented reality image by integrating multisensory data coming from an optical and a novel acoustic sensor (Echoscope). This virtual display of the working environment provides the basis for undertaking many typical underwater tasks in comparative ease compared with existing methods using video cameras only.
Available data are matched separately against a model in order to compute the pose of each single sensor with respect to the model reference frame. In addition, the calibration of the two sensors leads to the registration of 3-D acoustic data with 2-D optical image.
The system includes some interesting methods for both acoustic and optical data processing. Among these ones, the main significant issues addressed can be identified in the synergic use of two different sensory devices, the calibration of the relative pose of the two sensors using the observed objects, and the integration of 3-D and 2-D data at numerical level.

Acknowledgements

The authors would like to thank Dr. R. Hansen of Omnitech A/S[3] for kindly providing the images acquired by the Echoscope acoustic camera.

[2] http://www.disi.unige.it/project/venice/

[3] http://www.omnitech.no

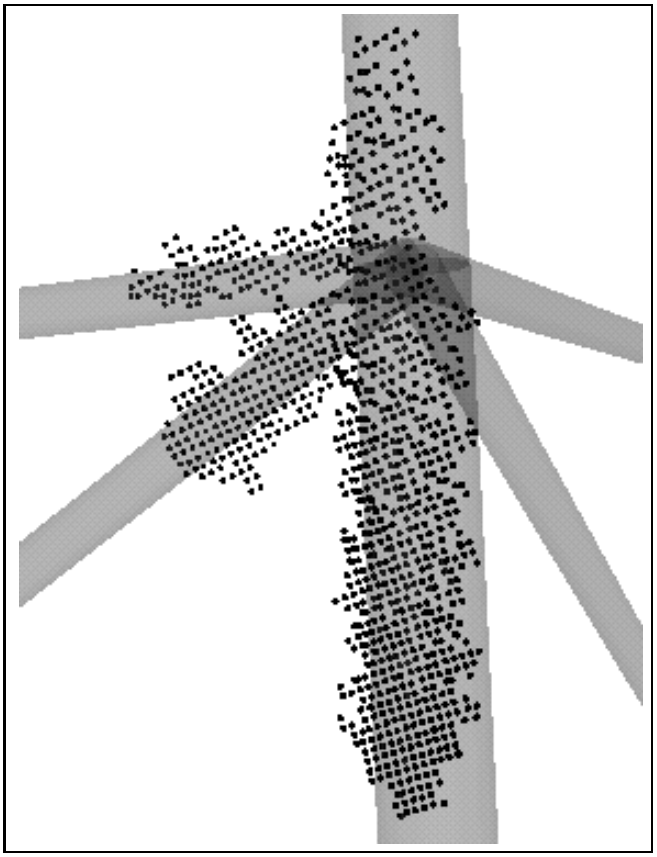

Figure 8. Virtual modeling of the scene with the 3-D acoustic points superimposed.

References

[1] J. Aloimonos and D. Shulman. Integration of visual modules, an extension to the marr paradigm. *Academic press*, 1989.

[2] D. Attali and A. Montanvert. Computing and simplifying 2D and 3D continuous skeletons. *Computer Vision and Image Understanding*, 67(3):261–273, 1997.

[3] P. Besl and N. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, February 1992.

[4] R. Brooks and S. Iyengar. *Multi-Sensor Fusion*. Prentice Hall, Upper Saddle River, USA, 1998.

[5] J. Burns, A. Hanson, and E. Riseman. Extracting straight lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(4):425–456, 1986.

[6] J. Canny. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, November 1986.

[7] L. D. Floriani, V. Murino, G. Pieroni, and E. Puppo. Virtual environment generation by CAD-based methodology for underwater vehicle navigation. In *Signal Processing IX, Theories and Applications (EUSIPCO '98)*, volume II, pages 1105–1108, 1998.

[8] A. Gelb, editor. *Applied Optimal Estimation*. The M.I.T. Press, 1974.

[9] W. Grimson, T. Lozano-Perez, and D. Huttenlocher. *Object Recognition by Computer: The Role of Geometric Constraints*. MIT Press, 1990.

[10] B. Gunsel, A. Jain, and E. Panayirci. Reconstruction and boundary detection of range and intensity images using multiscale MRF representations. *CVGIP: Image Understanding*, 63(2):353–366, March 1996.

[11] R. K. Hansen and P. A. Andersen. A 3-D underwater acoustic camera - properties and applications. In P.Tortoli and L.Masotti, editors, *Acoustical Imaging*, pages 607–611. Plenum Press, 1996.

[12] A. Jain and S. Nadabar. Edge detection and labeling by fusion of intensity and range images. *SPIE*, 1708:108–119, April 1992.

[13] K. Kanatani. *Geometric Computation for Machine Vision*. Oxford University Press, 1993.

[14] A. Lorusso, D. W. Eggert, and R. B. Fisher. A comparison of four algorithms for estimating 3-D rigid transformations. *Machine Vision and Applications*, 9:272–290, 1997.

[15] D. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):441–450, May 1991.

[16] R. C. Luo and M. G. Kay. Multisensor integration and fusion in intelligent systems. *IEEE Transactions on Systems, Man and Cybernetics*, 19(5):901–931, September-October 1989.

[17] V. Murino and R. Giannitrapani. Three-dimensional skeleton extraction by point set contraction. In *IEEE International Conference on Image Processing*, Kobe, Japan, October 1999. (in press).

[18] V. Murino, A. Trucco, and C. Regazzoni. A probabilistic approach to the coupled reconstruction and restoration of underwater acoustic images. *PAMI*, 20(1):9–22, January 1998.

[19] S. Pankanti and A. K. Jain. Integrating vision modules: Stereo, shading. grouping, and line labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):831–842, September 1995.

[20] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990.

[21] I. Pitas and A.N.Venetsanopoulos. *Nonlinear Digital Filters: Principles and Applications*. Kluwer Academic Press, 1990.

[22] L. Robert. Camera calibration without feature extraction. *Computer Vision, Graphics, and Image Processing*, 63(2):314–325, March 1996.

[23] G. Scott and H. Longuet-Higgins. An algorithm for associating the features of two images. In *Proceedings of the Royal Society of London B*, volume 244, pages 21–26, 1991.

[24] C. Y. W. Lie and Y. Chen. Model-based recognition and positioning of polyhedra using intensity-guided range sensing and interpretation in 3D space. *Pattern Recognition*, 23:983–997 1990.

[25] G. Zhang and A. Wallace. Physical modeling and combination of range and intensity edge data. *CVGIP*, 58(2):191–220, September 1993.