Using Visual Sensing for Control of an Underwater Robotic Vehicle

Richard L. Marks * Michael J. Lee † Stephen M. Rock ‡

Abstract

This paper documents research exploring the issues in using visual sensing to control an underwater robot. The local nature of visual sensing makes it useful for underwater situations in which the exact global location of the robot is not accurately known. In these situations, the *relative* measurement obtained from visual sensing can be used for precise control of the vehicle. To perform visual sensing in an underwater environment, several challenges must be overcome. In addition to standard vision processing issues, extreme lighting variations and marine snow make visual sensing problematic. To handle these difficulties, an approach using Laplacian of Gaussian sign correlation hardware has been chosen which offers robust real-time optical flow and stereo disparity measurements. This approach has been applied to several representative underwater vehicle tasks including: 1) object following, 2) station keeping, 3) mosaicking, and 4) navigation.

Local sensing

One of the largest problems in achieving automatic control of underwater vehicles is adequate sensing capability. Unlike land and space regimes for which GPS is available, there is no global position measurement system in place underwater.[1] Long baseline and other sonic beacon systems can be used in many situations; however, these systems cover limited areas and provide inadequate accuracies and update rates for many applications. A promising alternative to such systems is control from local sensors.

Using local sensors for control offers several advantages. The accuracy and bandwidth achievable with local sensors is much greater. In addition, the accuracies of many local sensors increase with proximity. Local sensors are carried on board and are self-contained. They do not rely upon proper functioning of external systems for operation. Many types of local sensing are unobtrusive and have no environmental interaction.

One of the most important characteristic of local sensors is their ability to measure parameters of interest directly. Rather than inferring relative information from global quantities, local sensors make relative measurements. This often limits the number of error sources and delays in a system and simplifies the control methodology. No global world model is necessary for reacting to local effects.

One particularly powerful type of local sensing for underwater robots is visual sensing. This paper describes several characteristics of visual sensing and presents an approach for achieving useful measurements in the ocean environment.

Visual sensing

Strengths

Visual sensing for robotic applications has been a topic of research for many years. It is unobtrusive, accurate, and high-bandwidth, and therefore ideal for sensing the environment. The information content is geometric and robot-relative, perfect for tracking and positioning applications [3, 4]. The following sections expand upon several of the strengths of visual sensing.

Geometric, robot-relative measurements

The sensing requirements for many robot applications are strictly geometric. Positions, orientations, and their rate-of-change comprise a large portion of the measurements used to accomplish many tasks. Images are a geometric projection of light from a 3-dimensional scene onto a 2-dimensional surface. By inverting this projection, the original 3-dimensional geometry can be reconstructed.

The geometric measurements of a visual sensor are made with respect to the sensor. Relative measurements can also be made between objects (e.g., the distance between an end-effector and an object to be manipulated). This eliminates errors which can occur due to

*Doctoral Candidate, Department of Aeronautics and Astronautics, rlmarks@sun-valley.Stanford.Edu

†Senior Research Engineer, Monterey Bay Aquarium Research Institute, lemi@hp850.Mbari.Org

‡Associate Professor, Department of Aeronautics and Astronautics, rock@sun-valley.Stanford.Edu

[1]Ideas for such a system are currently being examined; however, the accuracy would be on the order of 100 meters.

poorly-conditioned coordinate transformations and repeated discretization. The ability to measure the quantity of interest directly simplifies the sensing process.

Unobtrusive, accurate, high-bandwidth measurements

Optical sensing possesses several other characteristics which make it appealing, particularly compared to sonar. Optical sensing is unobtrusive in situations exhibiting adequate ambient lighting. When artificial lighting must be introduced, the effects on the environment are relatively benign. This is often not the case for sonar sensors, which add energy to the environment in the form of sound waves.

Resolution is typical measurement of sensing capability. Typical CCD cameras can be used to perform pixel measurements with milliradian spacing, and subpixel accuracies can often be achieved [22]. The resolution of positions computed from pixel measurements increases as the distance from the camera decreases; thus, resolution can be improved by moving the camera closer. Because the wavelengths used for optical sensing are much shorter than those used for sonar, the achievable resolution is much greater.

Bandwidth and latency are common measures of sensing capability. Because optical information travels at the speed of light, the bandwidth and latency of visual sensing is introduced solely by the camera and subsequent processing. This is not the case for sonar since the speed of sound in water is much slower ($\approx$ 1500 m/s). Reasonable CCD cameras generate more than 200,000 8-bit measurements at 60 Hz, with latencies of much less than one sample period. The amount of total data per unit time of a typical vision sensor is large compared to most other sensors.

Underwater issues

Visual sensing for underwater applications has received little attention compared to other sensing strategies despite the general acknowledgment of its many strengths. This is primarily due to problems in underwater optical imaging including limited range, nonuniform lighting, and marine snow. The following sections describe these issues and the effect that each has on visual sensing.

Limited range

Several physical effects limit the achievable range of underwater visual sensing. A combination of the attenuation of light in water with light absorption and scattering by suspended matter cause the amount of reflected light to exponentially decay as a function of distance. It should be noted that increasing the amount of illumination will not significantly increase the range for typical camera systems. This is because these systems tend to be contrast limited due to backscattering [7].

Nonuniform lighting

Nonuniform lighting causes the assumptions made by many computer vision techniques to be invalid. The lighting in typical underwater scenes is often nonuniform for several reasons. The attenuation of light as a function of distance causes objects farther away to appear darker. Because here is little ambient light at significant depths, artificial light sources must be provided. Efficient light sources are highly directional and do not illuminate the entire scene uniformly. Figure 1 shows an example of the degree of lighting nonuniformity commonly found in underwater images.

Marine snow

Marine snow is a term which describes the small, observable suspended particles which are often present in underwater images (Figure 1). These particles move with the water column but a small disturbance can cause them to swirl in a near-random fashion. The visibility of a given particle changes rapidly as it moves. The presence of these particles can adversely affect visual sensing measurements by causing the sensing system to base its measurements upon the particle locations. Scenes with marine snow have many more degrees of freedom because each particle moves independently. Processing strategies which make assumptions about image rigidity will break down, and the appearance and disappearance of particles can confuse feature tracking strategies.

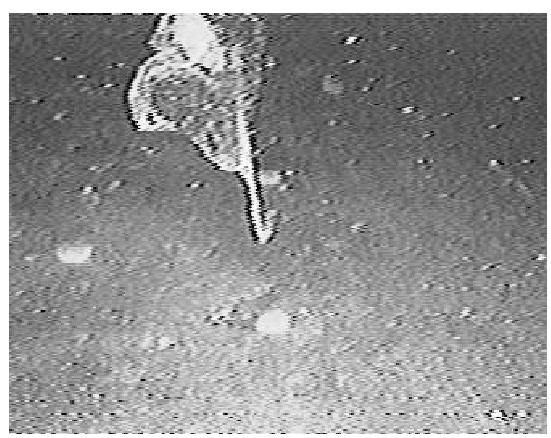


Figure 1: **Underwater image. Note the nonuniform lighting and presence of marine snow.**

Visual sensing approach

We have chosen a visual sensing approach which addresses the issues listed above. The visual sensing approach described below involves a combination of two basic ideas in computer vision–image filtering and image correlation. Image filtering is used to ensure that images will have certain characteristics. For example, a Gaussian filter can be used to smooth an image by removing spatial frequency information above a certain level. Image correlation is used to measure the likeness between sections of images, usually in order to establish correspondence. Maximum (perfect) correlation occurs when image sections are identical. The following sections provide a detailed description of the image filtering and correlation approaches we have chosen to utilize, followed by a explanation of why each was chosen.

Signum of Laplacian of Gaussian filter

The information present in typical images varies greatly. This makes it difficult for a single processing approach to perform successfully on all images. To address this, the information content of an image can first be filtered to extract only information of particular interest. Several common filters which have been used are: Gaussian filters, to lowpass the spatial frequencies in images; Gabor filters, to bandpass the spatial frequencies; and edge detectors, to extract the locations of sharp intensity transitions.

A filter with several interesting properties is the signum of Laplacian of Gaussian filter:

$$\mathbf{I}'(x,y) = \mathbf{sgn}[\bigtriangledown^2 * \frac{e^{\frac{-x^2-y^2}{2\sigma^2}}}{2\pi\sigma^2} * \mathbf{I}(x,y)] \qquad (1)$$

$\mathbf{I}(x,y)$ corresponds to the original image intensities, and $\mathbf{I}'(x,y)$ corresponds to the output intensities. The Gaussian convolution smoothes the image and acts as a low-pass filter, limiting the range of spatial frequencies. The frequency roll-off is adjusted by varying σ, the width of the Gaussian. Increasing the width reduces the range of frequencies that is passed by the filter.

The Laplacian portion of the filter is a double differentiation of the Gaussian-smoothed image (Figure 2). The Laplacian is a high-pass filter which completely removes both constant (DC) and linear-varying intensity information. The Laplacian evaluates to zero at local maxima of the magnitude of the intensity gradient. Zero-crossings of the Laplacian of Gaussian are often used to detect edges in images [13]. The zero-crossings are locally more stable in the presence of noise than most other simple image properties, thus making them attractive features for tracking and pattern matching.

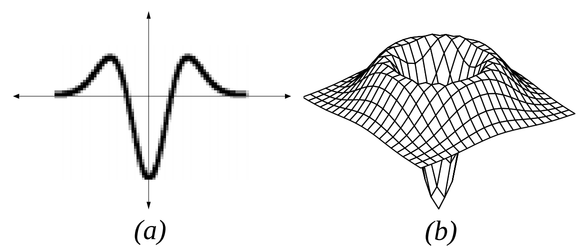


Figure 2: **Laplacian of Gaussian filter.** *(a)* **One-dimensional case.** *(b)* **Two-dimensional case, used for images.**

The final part of the filter is the **signum** function. This effectively binarizes the image, mapping all positive values of the Laplacian of Gaussian filtered image to white and all nonpositive values to black. The zero-crossings of the Laplacian of Gaussian occur at the borders between black and white. Thus, the image edge information is still present but further processing can be done on *areas* rather than lines. In the presence of noise, these areas can be tracked with greater accuracy than the actual edges [20].

Figure 3 demonstrates the three filtering steps described above.

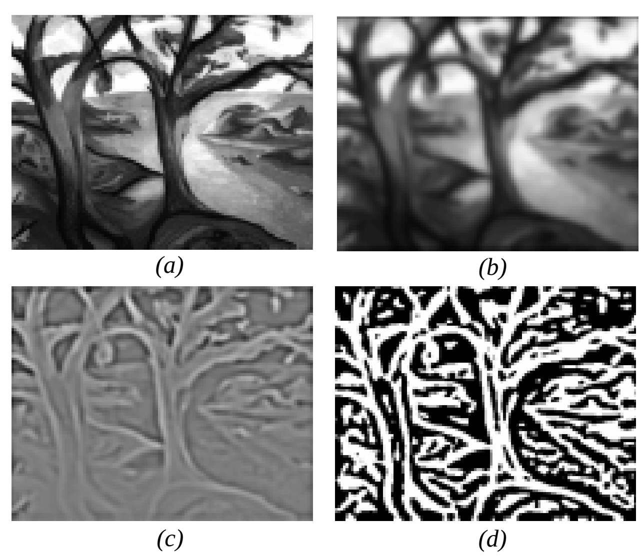


Figure 3: **Image filtering.** *(a)* **Original image.** *(b)* **Gaussian filtered image.** *(c)* **Laplacian of Gaussian filtered image.** *(d)* **Signum of Laplacian of Gaussian filtered image.**

Correlation

The *correspondence* problem occurs in many computer vision applications [15, p.220]. One of the most common techniques used to address this problem is correlation. Correlation can be thought of as computing the likeness of image sections. One image section is correlated with many other image sections to generate a *correlation surface*. The correlation surface can be examined to determine the section of maximum correlation. Figure 4 shows a generic example of correlation.

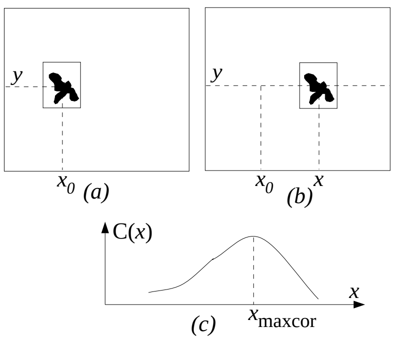


Figure 4: **Principle of the correlation technique.** *(a)* **First image section.** *(b)* **Set of second image sections.** *(c)* **One-dimensional correlation surface.**

For the binary images obtained from the filter in Equation (1), the correlation operation between two sections is simply the sum of XORs of the pixels. Correlation can be used to determine many useful pieces of information from the images. The following sections describe three types of information we have obtained from image correlation.

Stereo disparity (range)

One of the most common uses of correlation is stereo ranging [5, p.189]. For example, two parallel cameras perceive objects to be at different horizontal locations in the images (Figure 5). The difference between the locations is called the *disparity* and is inversely related to the distance of the object from the cameras (range). Correlation for ranging is a one-dimensional problem–matching sections must lie somewhere along a line which is a function of the camera geometry (known as the epipolar line). For the parallel setup shown, the epipolar lines for all sections are horizontal. This geometrical constraint greatly reduces the number of correlations required for stereo ranging.

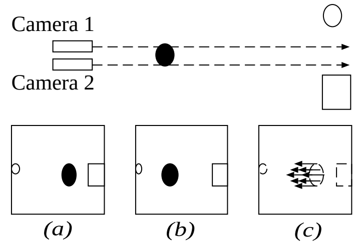


Figure 5: **Stereo disparity.** *(a)* **Image obtained from first camera.** *(b)* **Image obtained from second camera.** *(c)* **Calculated disparity.**

Optical flow

Correlation is also commonly used to compute the optical flow in images. Optical flow is best described as an approximation of the 2-D motion field (which is a projection of 3-D velocities onto the imaging surface) from spatiotemporal patterns of image intensity [6, 23, 1]. To compute optical flow, temporally separated images are correlated. Unlike stereo correlation, there is no geometrical constraint that limits the necessary search area to one dimension. However, an assumption can generally be made concerning the maximum velocity of image sections. For images with a sufficiently small temporal difference or sufficiently slow motion, the size of the search area remains tractable. Figure 6 shows an example of optical flow.

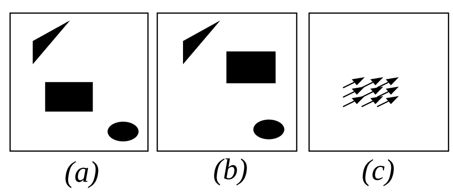


Figure 6: **Optical flow.** *(a)* **Image at time** t**.** *(b)* **Image at time** $t + \delta t$**.** *(c)* **Resulting optical flow.**

Feature tracking

Correlation can also be used to track features as they move about the image. Feature tracking is similar to optical flow calculation, except the search area is moved as the feature is moved. Optical flow is analogous to Lagrangian fluid dynamics analysis and feature tracking is analogous to Euclidean analysis.

The particular feature tracking approach we chose involves tracking features with respect to a starting image. All subsequent images are correlated with the first image. Thus, the feature which subsequent images are correlating against does not drift and true positional offset information can be obtained (see Figure 7). Note that this approach implicitly assumes that the feature will not change significantly in the subsequent images. A great deal of research has gone into algorithms for selecting "good" features. We have chosen to allow all image sections to be considered as potential tracking features.

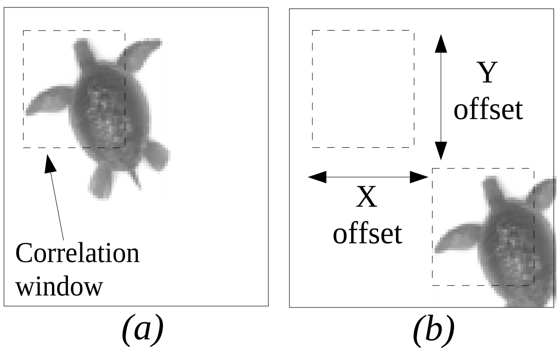


Figure 7: **Feature tracking with respect to an initial image.** *(a)* **Image at time $t_{initial}$.** *(b)* **Image at time $t_{subsequent}$. Positional offset from the initial image is measured directly.**

Characteristics of approach for underwater sensing

The combination of the specific filtering and correlation techniques described above is uniquely suited for underwater visual sensing. The Laplacian of Gaussian filter reduces the negative effects of many of the described issues. It acts as a band-pass filter, rejecting both high and low spatial intensity frequencies. By properly adjusting the filter width of the Gaussian, the high-frequency information introduced by the presence of marine snow can be removed. In addition, lighting biases and gradual nonuniformities are effectively reduced by the Laplacian. The lighting effects are further minimized by the binary quantization of the signum operation. The Laplacian and signum operations address the problem of low ambient lighting and limited contrast as well. An impressive demonstration of the power of the signum of Laplacian of Gaussian filter is shown in Figure 8 for two images with differing lighting, contrast, and marine snow. The two filtered images correlate better than 90 percent, and the inside quarters of the images correlate better than 95 percent.[1]

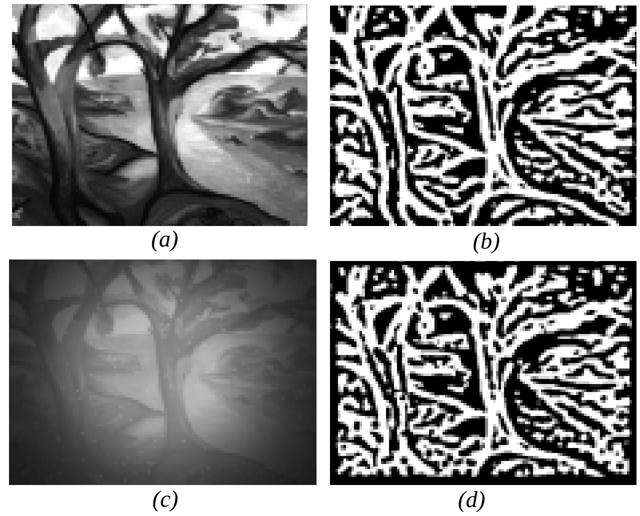


Figure 8: **The signum of Laplacian of Gaussian filter.** *(a)* **Original image.** *(b)* **Filtered image.** *(c)* **Original image with simulated contrast reduction, nonuniform lighting, and marine snow.** *(d)* **Filtered image. Note the similarity to** *(b)*.

The binary correlation technique is a fast, highly parallel operation. This allows many measurements to be obtained between each camera frame and makes correlation of large image sections feasible. Correlation can be used to obtain a variety of sensory data, including velocity, position, and range measurements. Thus, the same implementation (namely hardware) can be used to gather many pieces of information. We are using special-purpose hardware developed by Teleos Research[2] to perform the signum of Laplacian of Gaussian correlation [21, 19].

Applications of visual sensing

There are many underwater applications which could benefit from the use of visual sensing [14]. Because humans use vision as their primary sensor, many tasks are conceptually tied to visual stimuli. The most obvious of these are observational applications, for which the goal is to keep a particular object or scene centered in the image. An example of an observational application is fish (moving object) following.

[1]Contrast was reduced by dividing image intensity by 4 and centering to middle range (gray). Nonuniform lighting was simulated by an intensity reduction factor which is a linear function of $(r - r_0)^2$ for $r > r_0$, radius measured from image center. Marine snow was simulated by 250 uniformly random-spaced solid white 3×3 pixel squares.

[2]Teleos Research. Palo Alto, CA. *keith@teleos.com*.

Other types of applications use visual sensing to achieve different goals, such as vehicle positioning. Terrain-relative station keeping, ocean floor mosaicking, and mosaic navigation fall into this category. The following sections describe these four sample tasks in detail and present the requirements/specifications of each individually. The specific approaches we have developed are also presented briefly. Experimental verification of the approaches has been done using OTTER, an unmanned underwater robot (shown in Figure 9) being developed jointly by the Stanford Aerospace Robotics Laboratory and the Monterey Bay Aquarium Research Institute [25, 12].

Object following

The ability to automatically follow objects is necessary for certain underwater robot missions. Automatic object tracking is a component needed for a variety of military and scientific applications [2]. The particular application of object tracking we have investigated is the remote observation of fish by marine scientists. The goal of the fish-following application is to keep a single mobile undersea lifeform in the camera's field of view so that a scientist might be able to study it for an extended period (Figure 9).

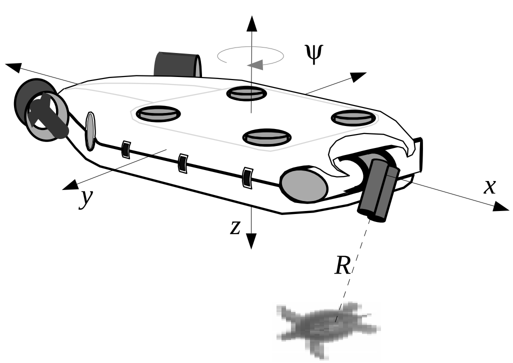


Figure 9: **Fish following. The robot keeps the turtle centered in the camera's field of view by changing location and moving the pan/tilt.**

The task of following a fish is difficult for several reasons. The speeds at which many fish travel is large compared to typical vehicle sensor and actuator bandwidths. Fish are asymmetric objects without any well-behaved geometrical properties. They can quickly change orientation, and some are even deformable. Natural camouflage can make it difficult to distinguish fish from terrain, and they often remain still so that their motion will not alert predators.

The sensing difficulties are all facets the same basic problem–quickly discerning the fish from the rest of the scene. This is a specific example of the *segmentation* problem, a widely studied topic in computer vision [15, p.113]. We have investigated two segmentation strategies, each to be used in different tracking scenarios. These strategies use fundamentally different visual measurements to discern the object from the background–range (disparity) and motion (optical flow). A more detailed account of these strategies can be found in [10].

Range segmentation

Range segmentation is useful for midwater tracking when no background terrain is within view. The fish can be separated from the rest of the scene by making the following assumption: range measurements closer than a certain threshold must correspond to a part of the fish, and larger measurements correspond to something which is not the fish. The effectiveness of this simple segmentation scheme is dependent upon the fish being large enough to produce several range measurements.

Spurious measurements also degrade the performance of such a scheme considerably. To help limit spurious measurements beyond the efforts taken to validate each measurement individually, further reasonable assumptions about the fish can be made. For most cases, it is valid to assume the 2-D projection of the fish is a single closed polyhedron. This assumption can be used to filter out noncontiguous measurements (which have a high probability of being spurious).

Motion segmentation

Motion segmentation can be used in situations involving background terrain which is sufficiently near to impede range segmentation. The key assumption behind motion segmentation is that the object is moving noticeably with respect to the background. Optical flow measurements can be used to differentiate between the background and the fish in many cases. If most of the image consists of background with a relatively constant optical flow, an overall image velocity can be obtained. Optical flow measurements inconsistent with this motion can be assumed to be measurements corresponding to the fish. This assumption is fairly weak; realistic situations can be contrived for which this breaks down. However, this assumption holds for many cases in which the background is relatively smooth and perpendicular to the camera.

The effectiveness of this segmentation scheme is also dependent upon the size of the fish in the image. It must be large enough to affect several optical flow measurements but still be small compared to the background. Spurious measurements can be identified by making the same assumption of contiguity as in range segmentation.

Tracking performance

The actuation bandwidth which can be achieved for an underwater robot is limited by many operational parameters. Typical bandwidths fall well short of the speeds exhibited by many fish. The effective tracking bandwidth can be improved by independently actuating the cameras with respect to the robot [24]. The OTTER robot possesses a high-speed cable-drive pan/tilt to accomplish improved object tracking performance (Figure 10).

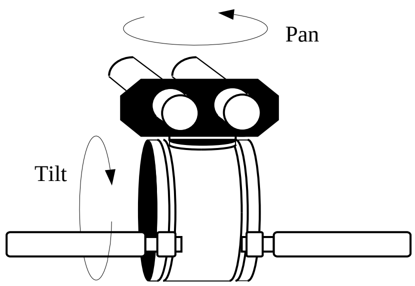


Figure 10: **OTTER's pan and tilt system for a stereo camera pair..**

Because of the directional nature of imaging, some amount of separate camera actuation is necessary for visual sensing to be most effective. The requirement of repositioning the entire robot in order to change the camera view is not acceptable for many applications. Also, in applications for which the desired goal is achieving a certain image for viewing, camera actuation may be sufficient. For many situations, robot positioning is unnecessary and can actually be detrimental due to excess power consumption and visual disturbance (e.g., mud clouds from the thrusters agitating the ocean floor).

Station keeping

Terrain-relative station keeping is an example of another observational task. Many applications require the ability to accurately maintain position for photographing, sampling, or performing other location-dependent functions. In many cases the desired position is relative to terrain such as the ocean floor or a canyon wall (Figure 11). Direct measurement of this relative position can produce a more accurate result than inferring position from other sensing systems.

The problem of visual station keeping of an underwater vehicle has been investigated by several researchers [18, 17, 16]. The approach presented in that work involves determination of the optical flow in a scene. Because the scene is assumed to be static, the optical flow

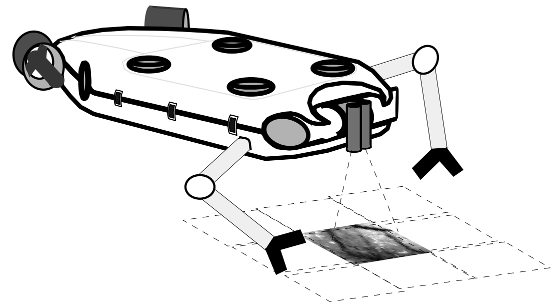


Figure 11: **Terrain-relative station keeping. The robot holds its position relative to an outcrop while collecting small rock samples. The arms are a part of future plans for OTTER.**

determines motion of the camera. Integration of this motion generates a measure of position.

Feature tracking

To avoid the possibility of drift due to motion integration, a feature tracking approach can be used in place of optical flow measurements. A suitably large section of terrain with adequate texture must be chosen as the feature to be tracked. The section is tracked with respect to an initial image which does not change with time. Thus, the initial image feature being tracked does not change and correlation can be used to obtain a true measure of its positional change. For the optical flow approach, some form of segmentation is needed to keep the station keeping reference from drifting.

The feature tracking approach assumes that the projection of the terrain section will not change significantly in subsequent images. Changes can occur for many reasons including camera motion and rotation, lighting differences, and occlusion. Image filtering can be used to reduce the impact these effects have on correlation. However, large angle and zoom changes will warp the terrain section enough that correlation will not provide useful data. To be effective, the feature tracking requires a minimal amount of net camera motion in these dimensions.

Translation, rotation, and zoom

A single feature can be tracked to obtain the vertical and horizontal image offsets. Assuming the scene is static and camera orientation and range to the scene remain constant, these offsets can be used to determine a scaled measurement of the robot's translation from the following equations:

$$x = R \sin(Offset_{vert}) \tag{2}$$
$$y = R \sin(Offset_{horiz}) \tag{3}$$

These measurements use the coordinate system shown in Figure 9 and assume the cameras are fixed to the vehicle and are pointed parallel to the z-axis. Note that $Offset_{horiz}$ and $Offset_{vert}$ are angular image measurements and R is the distance between the camera and the feature being tracked. For small offsets, the approximation $\sin x = x$ can be invoked to linearize the equations.

Multiple features can be tracked to obtain additional measurements of both rotation and scaled zoom (or range) [8]. Figure 12 shows the case of two features, one of which is chosen at the center of the image. Assuming only twist rotation occurs, and the offsets, rotation, and change in zoom are small:

$$x = R\,(\vec{t_1} \cdot \hat{\jmath}) \tag{4}$$
$$y = R\,(\vec{t_1} \cdot \hat{\imath}) \tag{5}$$
$$\psi = \frac{(\vec{t_2} - \vec{t_1}) \times \vec{r}}{|\vec{r}|} \cdot \hat{k} \tag{6}$$
$$z = \frac{(\vec{t_2} - \vec{t_1}) \cdot \vec{r}}{|\vec{r}|} \tag{7}$$

The z equation computes a range measurement normalized to the range of the initial image. The true range R can be obtained using stereo correlation, although in many cases a rough estimate is adequate for station keeping. Results of the accuracy of this approach are presented in [8].

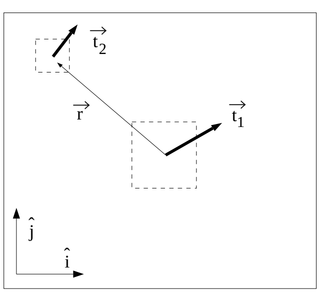


Figure 12: **Two tracked features. $\vec{t_1}$ and $\vec{t_2}$ are 2-vector measures of feature position offsets. $\vec{r}$ is the 2-vector displacement between the two features.**

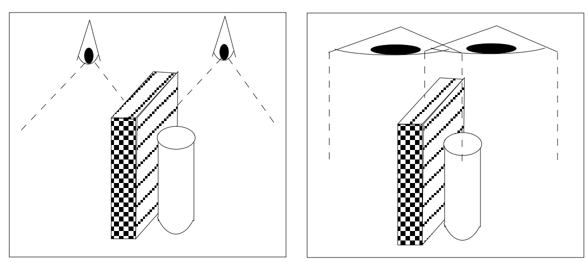
Mosaicking

Video mosaicking of the ocean floor is an application for which visual sensing is uniquely qualified. Mosaics are composite images made up of multiple smaller images which have been properly matched. Due to the limited visual range in the ocean, mosaics are the only means for creating single images of large areas. Background related to automatic mosaic creation can be found in [11].

To build a mosaic, a robot vehicle must move about and collect snapshots. To piece the images together into the final mosaic, some positional measure must be determined for each image. Feature tracking correlation can be used to obtain the relative positional offsets between images. In this respect, mosaicking is closely related to station keeping. Station keeping consists of regulating about a constant desired position with respect to a fixed section of terrain. Mosaicking, however, involves measuring positional offsets and repeatedly selecting new sections of terrain for tracking as the previous section moves out of view. Automatic mosaicking can be achieved by controlling the robot to zero the error between a desired positional offset and the measured offset.

Projection geometry

The mosaic quality which can be achieved is largely a function of the projection geometry of the mosaicking system. Most cameras can be modelled as pinhole cameras which image using *perspective* projection. Mosaicking multiple images obtained using perspective projection is problematic for nonplanar scenes, although no problems arise when *orthographic* projection is used. (Figure 13). The mosaic quality is directly related to how close the image projection geometry is to orthographic.



(a) *(b)*

Figure 13: **Projection geometry and mosaicking.** *(a)* **Perspective projection. The first and second eye observe completely different data in the overlapping parts of the images.** *(b)* **Orthographic projection. The first and second eye observe the same data in the overlapping parts of the images.**

Perspective projection can also cause a problem for feature tracking correlation. For station keeping, the offsets between the initial image and subsequent images are nominally small. However, for mosaicking this is not the case. The offsets are on the order of the field of view of the camera. The perspective effect can greatly degrade correlation performance for terrains of significant relief.

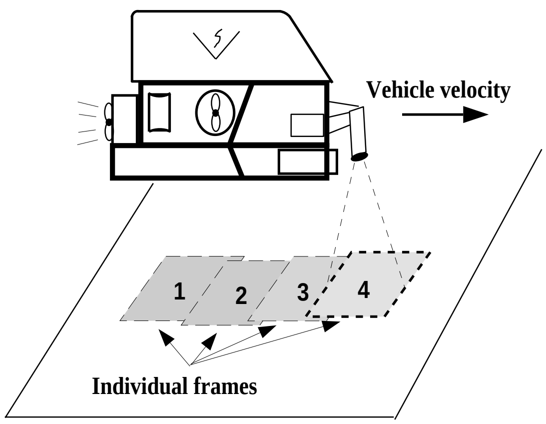


Figure 14: **Creation of a single-column mosaic.**

Single-column mosaics

Figure 14 shows how a robot can be used to generate single-column mosaics. The following algorithm can be used to gather images:

```
procedure make_single_column_mosaic()
{
Start:
  get_image()
  store_image()
  cor_window_pos = Center_Top_Of_Image
  repeat
    x = pos_of_best_cor_in_live_image()
    posError = Center_Bottom_Of_Image - x
    if automatic_mode  move_robot(posError)
  until posError < epsilon
  record_image_offset(x)
  goto Start
}
```

The error between desired offset and current offset is computed by the above algorithm; this can be used for servoing to create automatic mosaics (hence the *move_robot*() command above). Note that the positional offset of every new image is measured with respect to the previous image only.

This algorithm has been tested experimentally for a variety of platforms. Figure 15 shows a mosaic created using *Ventana*, MBARI's remotely operated vehicle.

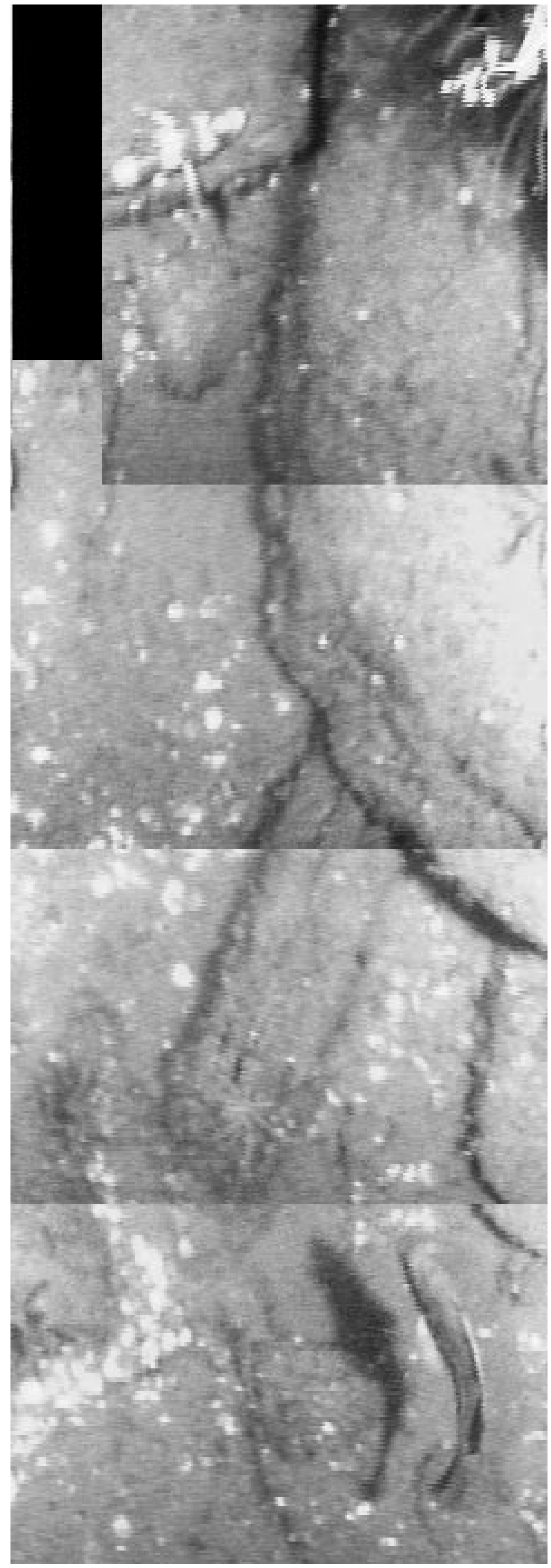


Figure 15: **A four-image single-column mosaic of the ocean floor.**

Multiple-column mosaics

The single-column algorithm can be extended to obtain images for mosaics of multiple columns by replacing *Center_Top_Of_Image* and *Center_Bottom_Of_Image* with values dependent upon the image number. For instance, at the end of each column, *Center_Top_Of_Image* could be replaced with *Right_Center_Of_Image* and *Center_Bottom_Of_Image* with *Left_Center_Of_Image* for a pattern like that in Figure 16. However, this simple extension of the single-column algorithm behaves poorly when slight errors occur in image offsets. This can be shown by examining the error propagation from image to image.

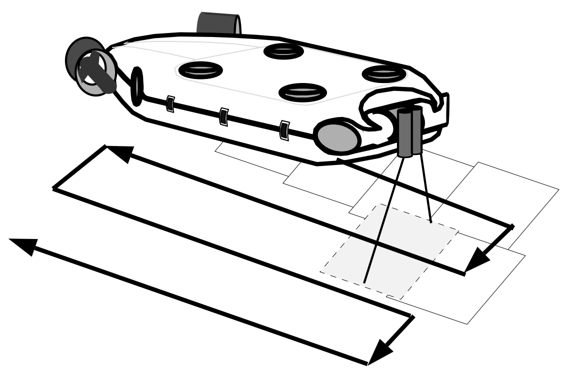


Figure 16: **Multiple-column mosaic creation pattern.**

Every time an image is obtained for the final mosaic, there is a chance for error in the measurement of the offset between the image and its predecessor. Between image 0 and image n there are n chances for errors to be introduced. If the errors are of constant magnitude and are modelled by a random walk, or if the errors have a Gaussian distribution, the expectation and standard of deviation of the error between image 0 and image n are:

$$\mathrm{E}(error) = 0 \qquad \sigma(error) \propto \sqrt{n} \tag{8}$$

If there is also a bias present in every offset measurement, this becomes:

$$\mathrm{E}(error) \propto n \qquad \sigma(error) \propto \sqrt{n} \tag{9}$$

As n grows large, this error can grow to substantial levels. Consider a square $m \times m$ mosaic of images 0 to n created by traversing the pattern shown in Figure 16. If m is even, image 0 and image n will correspond to adjacent corners of the mosaic with $m - 2$ images separating them. The expected error from Equation (9) for this worst-case scenario would be proportional to n, or $m^2 - 1$. The total separation between the images is $O(m)$, while the error is $O(m^2)$.

A mosaicking strategy for which the worst-case error is $O(m)$ is shown in Figure 17. Instead of always measuring offsets between sequentially obtained images, offsets can be measured between an images and its adjacent image from the previous column. The expected worst-case error for this approach is proportional to $m + m$, using the same assumptions that were made about the errors to obtain the results in Equation (9).

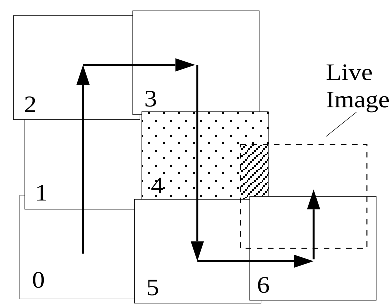


Figure 17: **Multipe-column mosaicking by correlation of images from adjacent columns. The live image, which will eventually be snapped and stored as image 7, is correlated with image 4.**

The improvement in performance achievable using the latter strategy is a strong incentive. However, this strategy also has several difficulties associated with it. Previously obtained images must be retrieved in a random-access fashion. The time required to do this can be considerable due to the amount of data contained in each image. During the retrieval time, it may not be possible to perform feature tracking because of conflicts between hardware resources or limits in computational resources.

Another difficulty arises because the amount of overlap between nonsequential images available for feature tracking is considerably less. For the case of sequential images, the live image and the image it is correlated against completely overlap immediately after an image is obtained (*Start* in the algorithm). As the camera is moved to obtain the next image, the amount of overlap decreases, but one dimension of the overlapping area is always equal to one of the image dimensions. For example, in case of square images of size $s \times s$, one dimension of the overlap region will be s. As shown in Figure 17, this is not the case for the second strategy. After image 6 is obtained, the live image is correlated with either image 4 (the case shown) or image 5. When the live image is halfway between image 6 and the eventual location of image 7, the height of the overlap region will be half the image height, or $s/2$. This assumes that it is possible to switch between correlating the live image between image 4 and image 5 at the halfway point. If this is not the case (due to image retrieval difficulties, for example) the overlap region would be even smaller.

It can also be more difficult to initially locate a matching feature in the live image when correlating to an image from the previous column. The offset between the images must be estimated from the offsets between previously obtained images, each of which may have some amount of error. The magnitude of the possible error of this estimate determines the size of the area which must be searched to ensure the location of correct correlation is examined.

Figure 18 shows an eighteen-image multiple-column mosaic of the bottom of a test tank which OTTER created autonomously. The second, more complex algorithm of adjacent column correlation was used to create the mosaic.

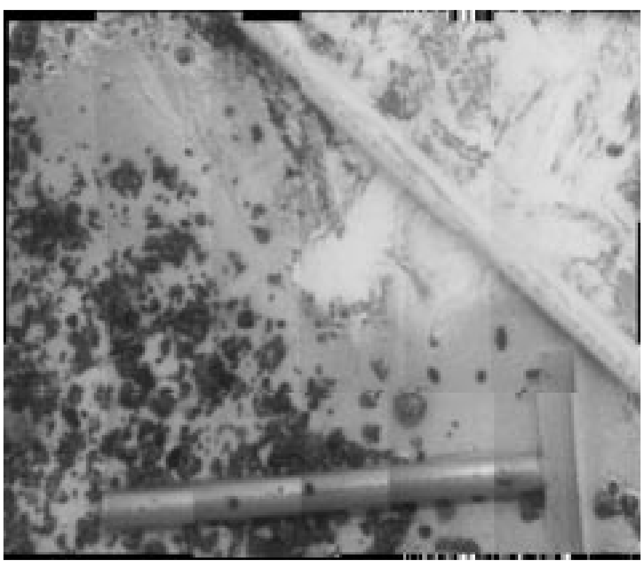


Figure 18: **A 3×6 image mosaic. The length of the aluminum pipe is approximately 2.5 feet.**

There are many other factors which affect mosaicking capability and performance. Rotation and scaling errors, for example, can cause significant problems for both of the described multiple-column strategies. A detailed analysis of multiple-column mosaicking strategies can be found in [9].

Navigation

One of the most powerful potential applications for visual sensing is robot navigation. Robot navigation can be described as a generalized positioning of the robot in a global sense. The automatic mosaic creation described above is a specific example of navigation for which the desired positions are a preprogrammed sequence of pixel offsets. More general navigation can be achieved given the prior existence of a high-resolution visual map (such as a mosaic).

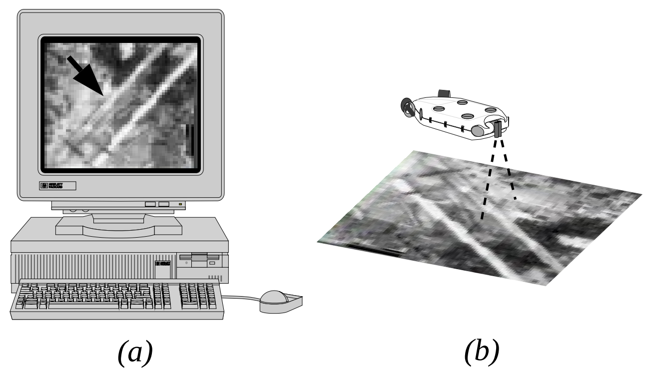


Figure 19: **Visual map navigation.** *(a)* **Operator selects a desired robot location.** *(b)* **Robot automatically moves to desired location.**

Consider the situation shown in Figure 19. Both the robot operator and the on-board automatic controller possess consistent visual mosaics of the ocean floor. The operator selects the desired robot location on the visual map by using a mouse, and the ontrol system automatically repositions the robot by correlating the live camera view with sections of the mosaic. This general capability would be useful for many types of applications. However, there are several issues which must addressed for such a capability to be realized.

Map correctness

The quality of the visual map could affect the navigation performance. In a mosaic, errors at image borders cause inconsistencies between the mosaic and the actual underwater scene. Nonuniform lighting and small offsets can cause sharp intensity discontinuities at image borders. This could cause correlation to fail as the robot passes over terrain corresponding to these parts of the mosaic.

This problem can be avoided if the robot has access to the original images used to create the mosaic. The original images could be used for correlation instead of sections of the mosaic. Although the overlapping image sections contain redundant information, there are no unnatural discontinuities. Correlation can be performed with one image until the robot is sufficiently above the next image. Note that when correlation is switched from one stored image to the next, mosaicking errors will necessitate a larger search area to find the correct correlation in the live image.

Map dimensionality

Another navigational issue arises because mosaics are only limited two dimensional representations of the world. For straight correlation between the mosaic and the live scene to work perfectly, the robot must be at the exact range and orientation as it was when that image in the mosaic was created. Although the correlation approach described previously can be used successfully despite small variations in orientation and range, there are limits to the maximum deviation for which correlation will work. Strict orientation and height limitations are not desirable in a general navigation system.

These limitations be overcome by changing the image used for correlation in one of two ways. By altering camera zoom and physical orientation, the resulting raw camera image can be matched to a section of the stored mosaic. The other method for achieving this result is to digitally warp the image with an affine transformation. Both approaches have difficulties. Mechanically achieving the required camera degrees of freedom may not be possible, and the computation required for digital warping is tremendous. It should also be noted that not every image in the mosaic was necessarily obtained at the same range and orientation. Thus, the process for matching the live image scene to the mosaic must be redone each time the robot moves over a scene corresponding to a different image in the mosaic.

Initial experiments

Although the problems for visual navigation may at first appear prohibitive, the experimental success we have shown for creating mosaics is nevertheless inspiring. The adjacent-column correlation strategy in particular faces many of the same issues. Initial experiments are planned for fixed-location pan/tilt navigation using mosaics generated from pan/tilt motion. This will demonstrate the ability to "look at that" while restricting the number of degrees of freedom which must be controlled.

Summary and Conclusions

This paper examined the use of visual sensing for an underwater robot. Several strengths and issues associated with using visual sensing in a marine environment were listed. The paper presented an approach which combined filtering and correlation to address these items. In light of this approach, four basic applications of visual sensing were characterized and strategies for accomplishing each were described.

The performance of the processing approach presented is exceptional considering the optical challenges imposed by the marine environment. The approach has been tested experimentally by applying it to several challenging representative applications. Results of these experiments have demonstrated capabilities previously unattainable by other means.

It is our conclusion that visual sensing has great potential for use in control of underwater robots. Many promising results have been obtained, although there remains a considerable number of avenues left to explore. Continuing research is being done on visual sensor characterization, navigation, and development of application strategies. Future work will focus on the integration of visual sensing with data obtained from complimentary local sensors including a scanning sonar and an inertial measurement unit.

References

[1] J.L. Barron, D.J. Fleet, and S.S. Beauchemin. Peformance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994.

[2] R. Blidberg. Autonomous underwater vehicles. current activities and research opportunities. *Robotics and Autonomous Systems*, 7(2-3):139–150, August 1991.

[3] F. Chaumette, P. Rives, and B. Espiau. Positioning of a robot with respect to an object, tracking it and estimating its velocity by visual servoing. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 2248–2253, 1991.

[4] B. Espiau, F.Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation*, 8(3):313–326, June 1992.

[5] O. Faugeras. *Three-dimensional computer vision: a geometric viewpoint.* MIT Press, 1993.

[6] B.K.P. Horn. *Robot Vision.* MIT Press, 1986.

[7] J.S. Jaffe. Sensors for underwater robotic vision: status and prospects. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 2759–2766, 1991.

[8] R. Marks, M. Lee, and S. Rock. Automatic visual station keeping of an underwater robot. In *Proceedings of IEEE Oceans '94*, Brest, France, September 1994. IEEE.

[9] R. Marks, S. Rock, and M. Lee. Analysis of strategies for automatic multi-column video mosaicking. Paper in progress.

[10] R. Marks, S. Rock, and M. Lee. Automatic Object Tracking for an Unmanned Underwater Vehicle using Real-Time Image Filtering and Correlation. In *Proceedings of IEEE Systems, Man, and Cybernetics*, France, October 1993. IEEE.

[11] R. Marks, S. Rock, and M. Lee. Real-Time Video Mosaicking of the Ocean Floor. In *Proceedings of IEEE Symposium on Autonomous Underwater Vehicle Technology*, Cambridge, MA, July 1994. IEEE.

[12] R.L. Marks, T.W. McLain, D.W. Miles, S.M. Rock, G.A. Sapilewski, H.H. Wang, R.C. Burton, and M.J. Lee. Monterey Bay Aquarium Research Institute/Stanford Aerospace Robotics Laboratory Joint Research Program Summer Report 1992. Project report, Stanford Aerospace Robotics Laboratory and Monterey Bay Aquarium Research Institute, 1992.

[13] D. Marr and E. Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London*, (207):181–217, 1980.

[14] David W. Miles and Glen A. Sapilewski. Uses of semi-autonomous underwater vehicles: A scientific survey. Technical report, Stanford Aerospace Robotics Laboratory and Monterey Bay Aquarium Research Institute, October 1992.

[15] V.S. Nalwa. *A Guided Tour of Computer Vision.* Addison-Wesley, 1993.

[16] S. Negahdaripour and J. Fox. Undersea optical stationkeeping. Improved methods. *Journal of Robotic Systems*, 8(3):319–338, June 1991.

[17] S. Negahdaripour, A. Shokrollahi, J. Fox, and S. Arora. Improved methods for undersea optical stationkeeping. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 2752–2758, 1991.

[18] S. Negahdaripour and C. Yu. Passive optical sensing for near-bottom stationkeeping. In *Proceedings of IEEE Oceans 1990*, pages 82–87, 1990.

[19] H.K. Nishihara. Prism: a practical realtime imaging stereo matcher. In *SPIE*, pages 134–142, 1983.

[20] H.K. Nishihara. Practical realtime imaging stereo matcher. *Optical Engineering*, 23(5):536–545, October 1984. Also in *Readings in Computer Vision: issues, problems, principles, and paradigms*, edited by M.A. Fischler and O. Firschein, Morgan Kaufmann, Los Altos, 1987.

[21] H.K. Nishihara and N.G. Larson. Towards a realtime implemenation of the Marr-Poggio stereo matcher. In *Proceedings of DARPA Image Understanding Workshop*, pages 114–120, April 1981.

[22] S. Schneider. *Experiments in the Dynamic and Strategic Control of Cooperating Manipulators.* PhD thesis, Stanford University, Stanford, CA 94305, September 1989. Also published as SUDAAR 586.

[23] A. Verri and T. Poggio. Against quantitative optical flow. In *Proceedings of 1st International Conference on Computer Vision*, pages 171–180, 1987.

[24] H. H. Wang, R. L. Marks, S. M. Rock, M. J. Lee, and R. C. Burton. Combined Camera and Vehicle Tracking of Underwater Objects. In *Proceedings of Intervention/ROV '92*, San Diego CA, June 1992.

[25] Howard H. Wang, Richard L. Marks, Stephen M. Rock, and Michael J. Lee. Task-Based Control Architecture for an Untethered, Unmanned Submersible. In *Proceedings of the 8th Annual Symposium of Unmanned Untethered Submersible Technology*, pages 137–147. Marine Systems Engineering Laboratory, Northeastern University, September 1993.