

# Autonomous river navigation

Franklin D. Snyder<sup>a</sup>, Daniel D. Morris<sup>a</sup>, Paul H. Haley<sup>a</sup>, Robert T. Collins<sup>b</sup>, Andrea M. Okerholm<sup>a</sup>

<sup>a</sup> Northrop Grumman, 1501 Ardmore Blvd, Pittsburgh, PA 15221

<sup>b</sup> Robotics Institute, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh PA 15213

## ABSTRACT

Existing maritime navigation and reconnaissance systems require man-in-the-loop situation awareness for obstacle avoidance, area survey analysis, threat assessment, and mission re-planning. We have developed a boat with fully autonomous navigation, surveillance, and reactive behaviors. Autonomous water navigation is achieved with no prior maps or other data – the water surface, riverbanks obstacles, movers and salient objects are discovered and mapped in real-time using a circular array of cameras along with a self-directed pan-tilt camera. The autonomous boat has been tested on harbor and river domains. Results of the detection, tracking, mapping and navigation will be presented.

**Keywords:** 3D Computer Vision, Autonomous Navigation, Detection, Tracking, Multi-camera Stereo

## 1. INTRODUCTION

There has been significant progress in autonomous navigation for air and land vehicles. However, current technologies do not take advantage of the opportunities nor account for the constraints of water navigation. The surface water domain is flat to the horizon or shoreline and contains few obstacles. Absolute localization is generally available via GPS but prior map information on water hazards and obstacles is not dependable and does not have the accuracy needed for precision navigation and sensor directed reconnaissance. Transitory obstacles such as other boats or floating debris are best detected and analyzed at navigation time with visual means. Sophisticated visual automation techniques have not previously been developed in the surface water domain.

We have performed research and development on a wide range of perception and behavior components to demonstrate autonomous river navigation and sensor directed reconnaissance. Rivers provide a challenging environment for autonomous operation as mobility is constrained, and the close proximity of riverbanks provides a cluttered environment with strong surface water reflections. The critical needs addressed with our video processing algorithms include riverbank detection and safe-area delineation, long-range tracking and mapping of stationary objects, and tracking and detection of moving objects. In addition, a variety of cues were developed to enable focus of attention for a pan-tilt-zoom camera, and automatic acquisition and recording of salient objects seen along the path of the boat. All of these technologies were implemented in a real-time system onboard a boat which we called Roboat. Details of the algorithms, instruments, and demonstrations are described in this paper.

## 2. WATER SURFACE DETECTION AND OPTICAL FLOW

A first requirement for water surface navigation is to robustly determine where the shoreline or riverbank is relative to the watercraft and so to first approximation where the craft can venture. But modeling water in order to determine its boundary is challenging. Color and texture may provide strongly distinctive cues in some situations, but a change in environment or weather can completely obviate these cues. For example water can sometimes be differentiated by color modeling, but this frequently fails in real environments due to strong reflections. Water surfaces also undergo non-rigid motion and vary in reflectivity resulting in a challenging surface to model and delineate.

Our work depends on a key observation: water surfaces observed by a camera provide sparse, short-lived and non-rigidly moving flow. While these factors are unfortunate for many applications, here they are leveraged to help distinguish the water surface from the riverbanks, and so determine the shoreline. In this section the water-surface properties that affect flow are explained.

Water has no inherent surface texture. Its appearance is a result of a combination of transmitted light, diffuse body emission, and reflected light. In general the transmitted light from the river bottom is negligible, and that body emission is roughly uniform. The remaining sources of texture are twofold. Texture is generated as the ratio of emitted light to

reflected light varies. Also the texture of the shoreline or nearby objects is reflected. These factors and their impact on optical flow are considered in detail below.

Typically the body light emitted from the water is much darker than the light reflected from the sky. Now as the water surface normal varies relative to the observer, the ratio of reflected light to body light emitted from each point on the water will vary. The larger the incident angle from the normal the greater the reflected component. Hence waves, which result in variations of water normal relative to an observer, can generate texture even with diffuse uniform incident light. This texture, however, typically has weaker gradients and provides fewer features than natural or manmade scenes. In addition, since waves move, this texture will continuously change in appearance and provide poor matches between successive images. Therefore, a video corner detector can be used to delineate water from other surfaces.

The second source of texture is reflected light. In open water areas much of this is sky with little or no texture. But in confined areas riverbanks and structures may have textures that are reflected or may generate textures on the water waves due to the varying surface normal. These reflected textures are strongest in calm water and become negligible in rough water. In addition direct sunlight can create texture on the water surface. In all of these cases the texture will vary in a non-rigid fashion over time. Thus features that are detected will often not find good matches over many frames, and when they are tracked they can be discarded by their non-rigid motion.

Figure 1 illustrates corner detection and tracking of features on water and on the shoreline over a short time interval. Most of the water features are eliminated after one second, and the ones that remain have significant random motion even though the camera was stationary in this case.



Figure 1 (a) The 1000 top features detected using a typical corner detector. Some but not many are detected on the water surface. (b) After tracking for one second at 10 frames per second most of the water features are lost whereas features on the land and a foreground object maintain track. While the camera has not moved, the water features have shifted considerably. In this example the detector and tracker used is that by [1] and it shows similar ability to exclude water surface features to our optical flow algorithm.

For real-time implementation our optical-flow algorithm was based on the work of [11]. The algorithm computes sparse, two-frame optical flow by applying Harris corner detection in the first frame, then searching for matches in the second frame using normalized correlation. Detected features are forced to be roughly uniformly distributed across the whole image by breaking the image into 16x16 pixel blocks and choosing the top N highest contrast corner features in each block. The blocks are also used as a spatial indexing mechanism for quickly locating candidate features within a predetermined radius for match score computation. Code bottlenecks such as normalized correlation computation are hand-coded in Intel Pentium MMX for maximum speed.

### 3. RIVERBANK ESTIMATION

Two complementary approaches were developed for riverbank estimation. One was based on elevation angle of objects in the images, and the other used a structure-from-motion approach. Both techniques depend on a careful calibration of the images and are described in this section.

### 3.1 Calibration

Six cameras provide a 360-degree field of view, with each camera covering roughly 65 degrees with 1024 pixels. The cameras were individually calibrated for intrinsic parameters (including radial and tangential distortions) using Bouguet's calibration code [2, 4]. Since object distance is large compared to the camera spacing, all cameras were approximated as having the same center of projection. Then using overlapping regions all six cameras were calibrated for relative extrinsic parameters as well as an additional correction on focal length. By enforcing a 360 total field of view, the individual fields of view, which are primarily determined by the focal lengths, can be made consistent and in so doing improve the focal length estimates.

Once calibration parameters were determined, lookup tables for desired measures were created. The undistorted, optically centered and focal-length-normalized pixel positions  $\widehat{u}_i, \widehat{v}_i$  are obtained:

$$(\widehat{u}_i, \widehat{v}_i) = L_p(u, v, i). \quad (1)$$

where  $u, v$  are the pixel coordinates of camera  $i$ . These undistorted values are used by the multi-camera stereo. Similarly a lookup table was created mapping pixels to aspect and elevation angles, denoted as:

$$(\theta^{(b)}, \phi^{(b)}) = L_a(u, v, i) \quad (2)$$

where  $\theta^{(b)}, \phi^{(b)}$  are the aspect and elevation angles in boat coordinates respectively.

### 3.2 Optical Sonar

Since the water surface is horizontal, the simplest measure for distance to an object is the elevation angle to a point on the water surface. Using this, the distance to water-surface objects and the shoreline can be obtained from single images. The following technique was developed to estimate nearby objects.

The transformation in equation (2) puts all camera measurements into a single boat coordinate system. These boat parameters can be transformed into world coordinate system parameters by forming a unit vector  $\mathbf{p}^{(b)} = (\cos(\theta^{(b)})\cos(\phi^{(b)}), \sin(\theta^{(b)})\cos(\phi^{(b)}), \sin(\phi^{(b)}))$ , then rotating this by the boat's rotation,  $\mathbf{R}$ , as obtained from an onboard Inertial Measurement Unit (IMU) to get:

$$\mathbf{p} = \mathbf{R}\mathbf{p}^{(b)}. \quad (3)$$

The aspect and elevation angles in world coordinates, denoted  $\theta, \phi$ , are obtained directly from  $\mathbf{p}$ .

Before using this flow, it is important to eliminate any points tracked on the water surface as these will confuse the shoreline estimate. First an epipolar-like constraint can be used. The direction of the boat's motion is denoted by vector  $\mathbf{d}$  in world coordinates. The normal to the flow in the viewing direction  $\mathbf{p}$  is given by  $\mathbf{n} = \mathbf{p} \times \mathbf{d}$ . The instantaneous flow can be expanded in angular space as:

$$\Delta\mathbf{p} = \frac{\partial\mathbf{p}}{\partial\theta}\Delta\theta + \frac{\partial\mathbf{p}}{\partial\phi}\Delta\phi. \quad (4)$$

The rigid-motion constraint requires that the flow be orthogonal to  $\mathbf{n}$ , that is:

$$\mathbf{n} \cdot \Delta\mathbf{p} = 0. \quad (5)$$

An example of valid flow directions is given in Figure 2. Applying this constraint eliminates the non-rigid motion of flow on the water surface. Additionally, unlike water features, shoreline points are generally clustered in high-density regions, and so by estimating the density of features and eliminating features in low-density regions, most of the remaining water surface features can be eliminated.

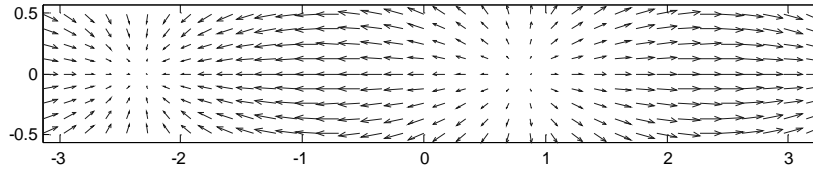


Figure 2 Plot of flow directions consistent with boat motion of  $\pi/4$  in aspect angle. Aspect and elevation angles are shown in radians and displayed horizontally and vertically respectively.

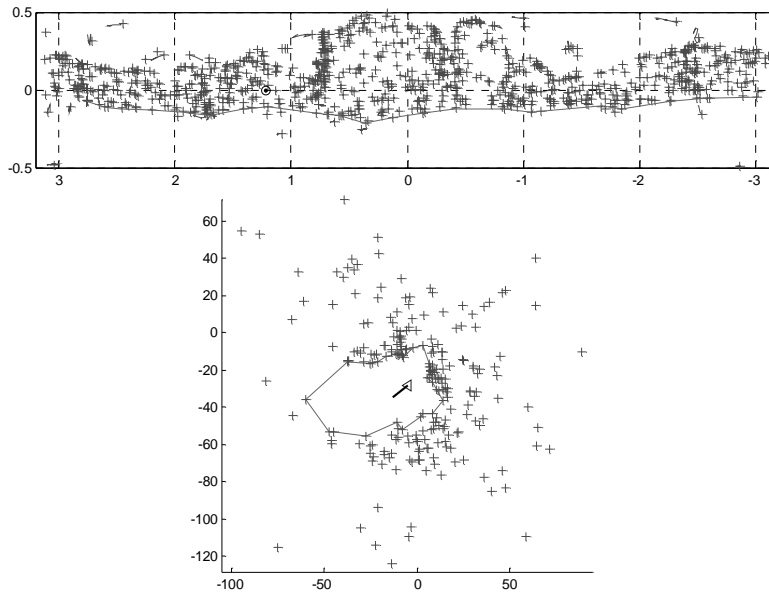


Figure 3 (Top) The flow over 10 frames in world coordinate aspect-elevation space  $(\theta, \phi)$ . Flow for points at infinity should be zero since boat rotation is removed. (Bottom) The projection of points below the horizon onto the plane is shown with units in meters. The inverse-convex hull technique is used to find a safe region and is illustrated by the boundary. The projection of this boundary is plotted on the top chart, and points below this are the ones that were eliminated.

It is now possible to leverage the shore-finding properties of optical flow. The flow calculated over 10 frames at 7.5 frames/sec from all cameras is transformed into aspect-elevation space as illustrated in Figure 3. Some of these points are eliminated using the procedure described above. Flow points below zero degrees elevation are potentially on the shoreline, and their position in the world space can be obtained by projecting them along the line from the camera center-of-projection on top of the mast to the water plane. These are shown in the lower portion of Figure 3 with non-rigid and isolated points removed. This will over-estimate the distance to points that are actually above the water plane, but that is okay since the points of interest are on the shoreline close to the water and the goal is to achieve the boundary formed by these. This boundary is found using an inverse convex hull technique described in Section 3.4, and typical results are shown in Figure 3. The cameras act similar to sonar giving a distance-to-object at orientations around the boat, and hence we call this *optical sonar*. An important limitation of this technique is the accuracy to which the vertical is known as this will significantly affect the accuracy of point positions close to the horizon. The IMU used here was accurate to 2 degrees, and thus a set of virtual points 2 degrees below the horizon were added to limit the extent of the boundary. With the mast at 2.25m, this limits the range to 65m.

An important advantage of optical sonar is that its scale is known, unlike structure-from-motion approaches where scale must be set using known boat distance traveled. It is useful for estimating nearby shorelines and objects and works equally well in all directions. For longer-range objects and shorelines a motion-based estimation is used as described in Section 3.3.

### 3.3 Multi-camera Stereo

Multi-frame analysis using optical flow avoids many of the single-image shoreline-finding problems by also capitalizing on the fact that water reflections will not support a persistent track of features over many frames. Additionally if the tracks are required to be consistent with stationary points as in the case of a riverbank feature even more on-water detections are eliminated. Described next is the use of six synchronized and calibrated cameras to provide data for implementing 3D structure from motion by optical flow. River bank estimation is then obtained by processing the 3D points to detect the boundary of non-water locations.

Multi-camera stereo processing of simultaneous frames not only provides full 360 degree coverage but also yields an optimal numerically conditioned estimation problem. For a pair of opposite facing cameras, camera rotation gives similar flow in both camera's data while camera translation gives flow of opposite signs in both camera's data. Singular value decomposition of the Jacobian matrix describing this relation shows minimal spread in the singular values while for the single camera analysis the opposite is true. Moreover the use of six equally spaced in azimuth cameras virtually eliminates degenerate cases, except for the case where the boat is completely stationary.

To use all cameras simultaneously in a single stereo calculation, all the camera images are rotated into a common boat image plane, in our case facing upwards:

$$\hat{\mathbf{u}}^{(b)} \equiv \begin{pmatrix} \hat{\mathbf{u}}^{(b)} \\ \hat{\mathbf{v}}^{(b)} \\ \hat{\mathbf{w}}^{(b)} \end{pmatrix} = \mathbf{R}_i \begin{pmatrix} \hat{\mathbf{u}}_i \\ \hat{\mathbf{v}}_i \\ 1 \end{pmatrix} \quad (6)$$

where  $\mathbf{R}_i$  is the appropriate rotation for camera  $i$ . Then features from all cameras can be treated uniformly as if from one camera, and the epipolar geometry between two boat positions,  $k, h$ , written as:

$$\hat{\mathbf{u}}_k^{(b)T} \mathbf{E} \hat{\mathbf{u}}_h^{(b)} = 0, \quad (7)$$

with  $\mathbf{E}$  being the Essential matrix. The essential matrix can be robustly solved for using Random Sampling Consensus (RANSAC) to eliminate outliers, followed by a full non-linear optimization of feature points and boat rotations to obtain 3D structure and boat motion. An example is shown in Figure 4, that shows significantly greater range capability than optical sonar.

### 3.4 Inverse Convex Hull

Once points on the plane are estimated, both the optical sonar and the multi-camera stereo techniques need to obtain a safe area for boat motion. Ideally this is the largest region around the boat with no shoreline points in it, however the shoreline is only sparsely sampled. Thus an "inverse convex hull" was created. For multi-camera stereo, 3D points are then projected to a plane parallel to the river surface yielding a scatter plot of points centered about the camera system (boat position). As the inside, rather than the outside, of this point distribution is needed, points are transformed to have the same angle but the inverse distance from the center. Then the convex hull of this is taken to find which are the appropriate boundary points. Details for this method are as follows. Denote  $(x_i, y_i)$  as the location of point  $i$  with,

$$\beta_i = \tan^{-1} \left( \frac{y_i}{x_i} \right) \quad \text{and} \quad r_i = \sqrt{x_i^2 + y_i^2}. \quad (8)$$

The inverse rooted distance

$$d_i(a) = r_i^{-a} \quad (9)$$

is computed along with the points

$$x_i^{inv} = d_i(a) \cos(\beta_i), \quad \text{and} \quad y_i^{inv} = d_i(a) \sin(\beta_i) \quad (10)$$

Boundary points are the parents of the points defining the convex hull of  $\{x_i^{inv}, y_i^{inv}\}$ . Depending on the density of points, the inverse parameter,  $a$ , can be varied to determine how many of the nearby points to include in the boundary. Experimentation has shown  $a = 0.05$  is a suitable value for a typical density of 100 tracks per camera.

River bank and river direction are gotten by further processing the identified boundary points. Second order moments of the 2D distribution of the boundary points are computed along with the major and minor axes. If the major axis is strongly indicated by the ratio of the major to minor moment the river direction can be associated with the direction of the major axis. The major axis also can be used to bifurcate the boundary points into two sets identifying each bank of the river.

The convex hull is used to define boundaries for safe navigation. Figure 4 shows the safe navigation boundary derived with the optical multi-camera stereo convex hull algorithm.

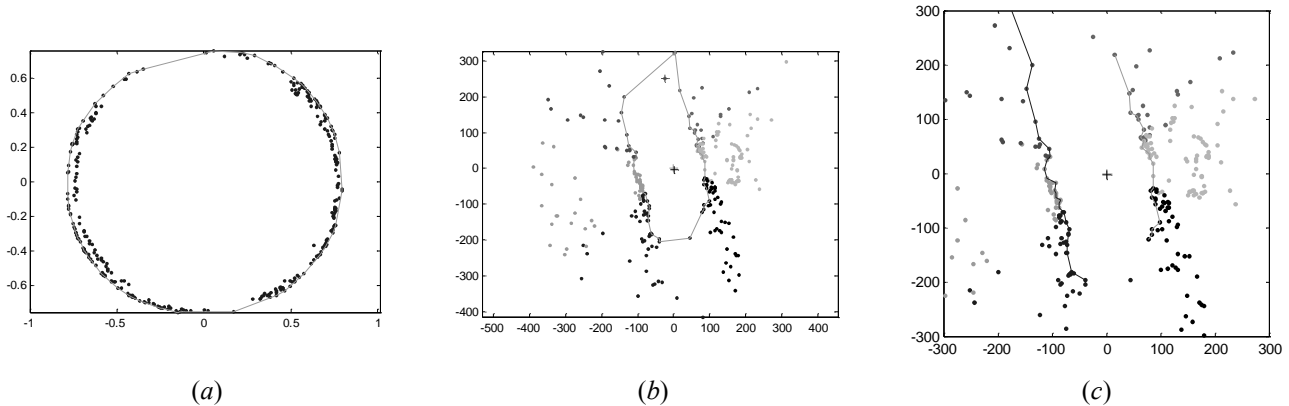


Figure 4 (a) The inverse hull for shoreline points determined by multi-camera stereo. (b) The resulting safe area produced by connecting the points from the inverse convex hull. (c) The riverbanks where the boat is the + in the center. Units are in meters.

#### 4. CLUSTER TRACKING, MAPPING AND MOVING OBJECT DETECTION

Most 3D estimation techniques operate on a collection of single features. This works for short-term tracking in single cameras, but it fails when an object moves between cameras. Also since individual features are not reliable they cannot be used on their own to detect moving objects but require a later global grouping as in [Costeira95]. In our system we wished to track objects for long periods of time and maintain track of them even as they passed from one camera to the next, and simultaneously to map them if they are stationary, or determine if they are moving. Hence rather than working on the flow features in image space, we proposed tracking clusters of features in aspect-elevation space. Clusters give more reliable tracks whose motion can be individually determined to be stationary or moving, and by working in aspect-elevation space from overlapping cameras, clusters are naturally tracked between cameras. These clusters are generated by initially laying a grid of rectangular regions over the aspect-elevation space and using mean shift to converge each of these to a local maxima of feature density. Initial feature clusters are shown in Figure 5. Clusters are tracked using a rigid 2D model with scaling. When new features are found, they are added to the closest cluster if their motion is compatible with the 2D motion of that cluster. Features in that cluster that fail to fit the 2D model with scaling are excluded using RANSAC. A fixation point is maintained for each cluster even though features are continuously added and removed. This is similar to the affine tracking done of Hayman et al. [8], although here tracking is done in a world-based aspect-elevation space enabling us to use a more constrained 2D plus scale model. With this model a minimum of two points, rather than 3 for affine, are needed to maintain a cluster, and we found that clusters could be reliably tracked over hundreds of frames. Periodically new clusters are initialized.

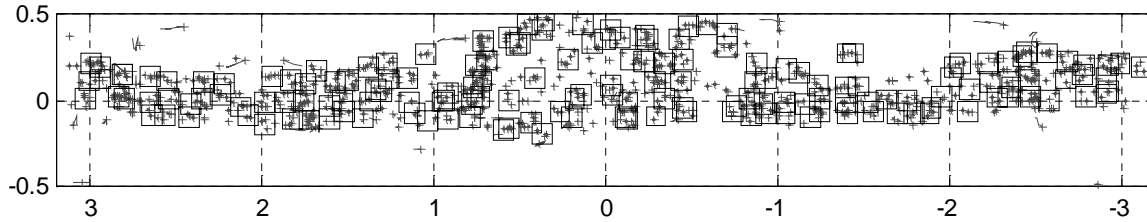


Figure 5 Clusters (squares) of features (+ marks) are initialized in the aspect-elevation space, and tracked in this space.

Clusters-objects are mapped in the 3D world in an online fashion. For each cluster object an initial viewing direction and boat position are stored, and an inverse distance to it and covariance are updated as the cluster is tracked and its position triangulated, as illustrated in Figure 6. An object is determined to be moving when its newly triangulated value is incompatible with its previously updated location and covariance. Figure 6 shows detections of a moving object in front of the boat path.

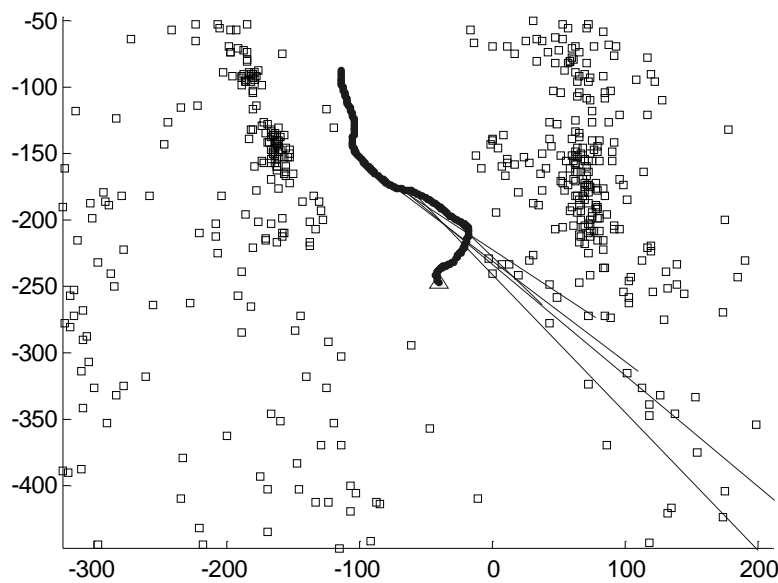


Figure 6 Two minutes of the boat progressing up the river (the final point is the triangle) with cluster-objects mapped on each riverbank. Units are in meters. Another moving boat was detected in front of the boat trajectory, and the bearing to it is indicated by the solid lines.

## 5. FOCUS OF ATTENTION PROCESSING

There has been relatively little work in computer vision and robotics on the high-level Visual Attention task of finding salient objects in the scene [9; 3]. This perceptual task is largely pre-attentive in humans. We have developed a focus of attention mechanism to control a pan/tilt camera on the boat to point at and record “interesting” areas of the scene. This is achieved by finding blobs of pixels that are distinctive in either color or motion from their surroundings.

### 5.1 Color Blobs

Much current work on detecting color landmarks for robot navigation is very simple – the environment is augmented with artificial landmarks having bright, vibrant hues, so that detection reduces to color thresholding on a user-defined portion of color-space [5; 12]. Although this approach is fast, it is extremely limited to demo-engineered environments with controlled illumination. In contrast, we search for distinctive color blobs in natural outdoor scenes, by replacing absolute definitions of color with a measure of relative distinctiveness, where distinctive is defined as having color characteristics that are statistically atypical with respect to the local scene around the blob. By using a definition of

color distinctiveness that is computed relative to a current histogram of scene colors, the approach is self-adapting to the wide variation in lighting that occurs in typical outdoor scenes



Figure 7 Some sample color blobs extracted during a test run on the river. These correspond to: the dock support poles, boat covers, buildings, lifejackets, distinctive trees and stumps, bridges.

The following computationally inexpensive algorithm is used for finding color blobs that are “unusual” with respect to the overall palette of colors observed in the image. The first step is to form a histogram of the U,V color components of the current frame (note, the raw images produced by our firewire cameras are in YUV format, so we can access the U,V color data directly from the image buffer). This color histogram indicates which colors appear most frequently in the image. We smooth the histogram with a Gaussian kernel and normalize, which is equivalent to forming an empirical color probability distribution using Parzen estimation. To find unusual colors, we take the negative log likelihood of the values in this color distribution, scale them between 0 and 255, and project these values back into the pixels of the image. Pixels with “typical” colors in terms of frequency of occurrence appear dark in the new image, whereas pixels with unusual colors appear bright. To find spatial regions of unusual colors, we threshold the scaled negative log likelihood image and look for connected components representing blobs of neighboring pixels having unusual colors. Each such connected component forms an “unusual color” blob, and blobs are ranked according to their average negative log likelihood score. In the current implementation, we save the top five color blobs for later review. We also send the top-ranked color blob to the pan/tilt camera for camera servo control. Figure 7 shows a small sample of the color blobs that were extracted during a test run on the river.

## 5.2 Motion Blobs

Independent object motion is a strong saliency cue. We wish to detect independently moving objects in the vicinity of our boat. The problem is difficult, since the boat (and thus camera) are in continuous motion, and therefore simple motion detection techniques based on background subtraction or frame differencing are not applicable. We use a more general approach, based on clustering optic flow vectors. The input to the algorithm is the real-time, sparse, two-frame optic flow described in Section 2.

After finding sparse flow, the algorithm decomposes the flow field into overall apparent background flow, induced by motion of the camera, and independent flow produced by moving objects in the scene. We have implemented an efficient flow clustering approach based on the RANSAC algorithm. We approximate the global background motion by an affine transformation, fit by RANSAC through repeatedly sampling sets of three flow vectors. The affine transformation that accounts for the most flow vectors is chosen as the model of background motion. After fitting the background motion model, a second RANSAC procedure is performed on the outlier flow vectors to find the largest consistent set of spatially contiguous outlier flow vectors that exceeds a threshold size. The location and shape of a set of vectors that passes this test comprises our motion blob, and is passed as a candidate “interesting” region to the pan/tilt head servoing mechanism. Note that motion regions extracted in this way can correspond to nearby objects exhibiting parallax motion, as well as independently moving objects. Although not strictly “moving” objects, nearby objects with lots of parallax motion are indeed good candidates for being interesting regions to look at, since an object that close to



the boat could be a hazardous obstacle that should be avoided. Some examples of motion blobs detected during a test run on the river are shown in Figure 8.



Figure 8 Some sample motion blobs extracted during a test run on the river. These correspond to people walking in front of the camera while still docked, and passing boats seen while traveling on the river.

### 5.3 Focus-of-Attention for Pan/Tilt Control

To demonstrate the above mechanisms for selective attention, we designed a real-time demo to direct Roboat's pan/tilt camera to point at "interesting" scene features that the boat encounters on its journey. We use a simple measure of salience to choose between multiple interesting regions detected within the same frame. Motion regions are given highest priority, since by definition they need to be attended to first before they leave the camera field of view. By totally ignoring color regions while a motion region is being attended to, it is possible to servo the pan/tilt camera fast enough to track moving objects. When no moving objects are in the field of view of the pan/tilt camera, detected color blobs are ranked by the average negative log likelihood of the pixels within them. In this way, blobs with the most unusual (least likely) color are given higher priority for attention. If a single color blob remains most distinctive over a period of time, the camera appears to automatically servo to keep that blob in the field of view. Thus, a very simple perceptual control loop is able to exhibit behavior that can be interpreted as purposeful by a human viewer – the camera appears to search for salient scene regions, then servos to fixate on the most salient region as the boat passes by.

## 6. NAVIGATION

### 6.1 Obstacle Avoidance

The obstacle avoidance module takes in the angle of the nearest shoreline and the visual sonar array, observes the location of obstacles, and adjusts the heading to avoid them. The new heading is then passed on to the autopilot to be executed. The purpose of the module is to do local obstacle avoidance, not planning, and it assumes that the desired heading is otherwise optimal, and therefore chooses a new heading as close to the input heading as possible while maintaining a distance of 5m from all obstacles detected by the optical sonar.

Beginning with a completely clear (traversable) angle map of 720 half-degrees, all obstacles (visual sonar depths) within 30 meters of the current location were examined. For each obstacle within 30 meters, the angles subtended by a circular configuration space with a 5-meter radius were marked as non-traversable. Once this angle map was completed, the desired heading was checked. If the desired heading was still marked as traversable, then the obstacle avoidance routine exited. However if the desired heading had been marked as non-traversable due to an obstacle, a linear search was conducted in a  $(5\pi/6, -5\pi/6)$  bracket centered on the desired heading to find the nearest traversable heading, which was then returned as the new plan heading as shown in Figure 9. If this also failed to find a traversable heading, then there exists no safe heading in the desired river direction (upriver or downriver) and we must turn around. In that case, the above algorithm was run again with an input heading of  $\pi-h$ , where  $h$  is the previously desired but non-traversable heading. If both searches fail, an unrecoverable error is posted.

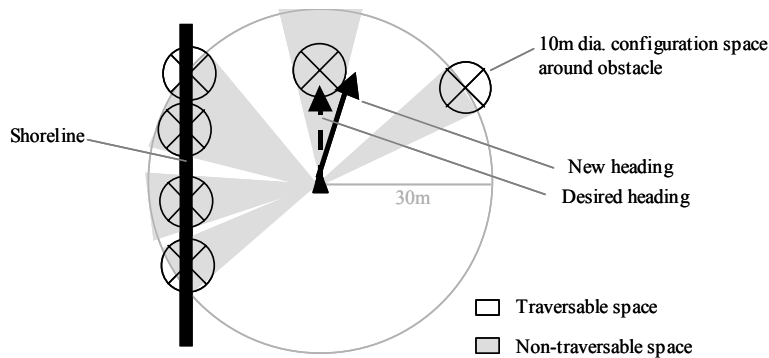


Figure 9: Diagram of the obstacle avoidance algorithm. Heading angles that would result in passing 5m or less from one of the obstacles are shaded, marked as non-traversable. The desired heading (dashed arrow) is blocked by an obstacle, so a new heading (solid arrow) is chosen that is at least 5m away from all obstacles.

In simulation, this algorithm successfully chose a traversable heading each time. In on-water testing, the algorithm always chose a traversable heading, but sometimes the boat did not have sufficient thrust to achieve the new heading. In future work, the algorithm will be updated to take into account the dynamics of the boat on the river.

## 7. SYSTEM HARDWARE AND ARCHITECTURE

### 7.1 Hardware and Sensors

Maritime image/video processing algorithms were tested on a 13 ft electric deck boat that operates at 5 kn using rechargeable batteries. The test craft is configured with maritime sensors, instruments, and actuators (Figure 10) including: GPS, electric compass, six cameras for 360 deg visual coverage, radio data link, constant heading autopilot, and on-board real-time image processing and control computers. All of the sensors are interfaced to the on-board real-time multi-computer system for sensor processing enabling autonomous navigation and reconnaissance.

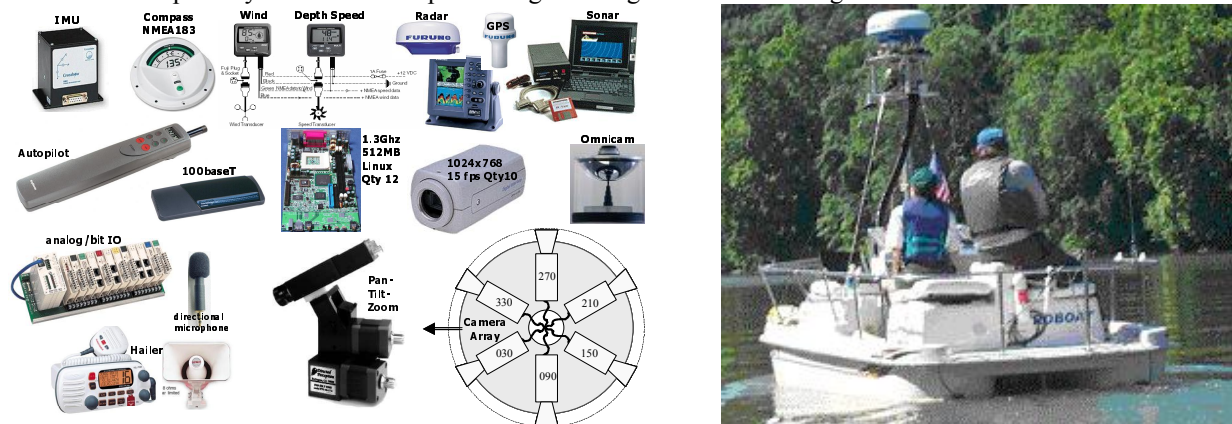


Figure 10 Sensors, computers, and actuators used for autonomous navigation and surveillance.

In order to produce accurate results it was necessary to ensure that all six directional cameras snapped pictures at exactly the same time, however the cameras that we have do not support triggering and automatic exposure simultaneously. We developed a two-part solution to this problem. In hardware, we clocked the cameras with the same line and reset them together so that their base frame clock rate of 15 fps was perfectly in sync across all six cameras. Since our processing was slower than this, we then further reduced the frame rate to 7.5 fps in software, taking into account dropped frames.

## 7.2 System Architecture

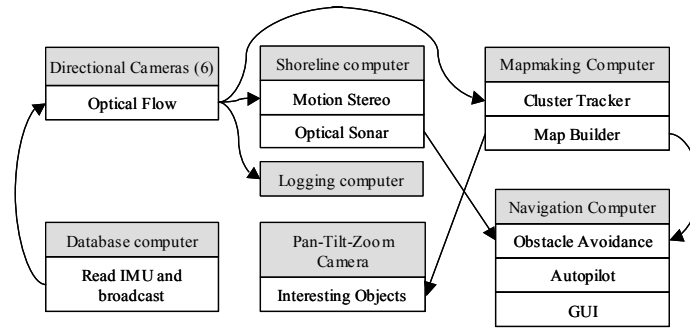


Figure 11 System diagram showing the computers and inter-CPU data transfer.

The processing was achieved with 15 single-board Pentium3 computers running Red Hat Linux 8.0 with a custom compiled 2.4.22 kernel. Math and image processing algorithms used the Intel Integrated Performance Primitives (IPP) 3.0 and Math Kernel Library (MKL) 6.1.

One computer was dedicated to each Firewire camera (six directional cameras, and one pan-tilt-zoom camera), running software to assure phase lock between the cameras, and processed optical flow from each frame to be sent to other computers (Figure 11). The optical flow results from the six directional cameras was sent to three locations: a flow logging computer, a map computer, and a computer to develop the convex hull shoreline and clustering flows into interesting objects. The pan-tilt-zoom camera used its optical flow locally. Results from the convex hull shoreline and from the map generator were transferred to another computer, which then checked the path for obstacles, read the maritime NMEA sensors, controlled the autopilot, and displayed the results in a GUI. Finally, a computer reads and broadcasts IMU data via UDP messages at 60 times a second. The IMU computer also logs all events into a MySQL database. All of the inter-CPU communication was achieved with TCP or UDP socket messages.

## 8. SUMMARY

We have described our developments of technologies for vision enabled river navigation and sensor directed maritime reconnaissance. After exploring contrast, color, and texture approaches, we concluded that the water surface could most robustly be distinguished from land using persistent optical flow. We demonstrated two independent shoreline algorithms using mast-height optical sonar and multi-camera shape from motion. Our improvement on stereo was to transform all images into a single image-plane, and this lead to far fewer degeneracies. River obstacles and other objects of interest were tracked across multiple cameras using feature clusters in aspect-elevation space and mapped onto the 3D world. Distant moving objects were detected and tracked using feature clusters, and nearby movers were detected with motion blobs. A new approach for finding salient objects based on color distinctiveness was developed and demonstrated with a focus of attention algorithm that directed the pan-tilt camera. A reactive path planner was implemented to use the shoreline and obstacle information to periodically re-plan safe navigation up or down the river. All of the techniques developed were tested and demonstrated on an autonomous river boat that navigates on a river while performing sensor directed reconnaissance. Additional refinements are expected to provide more accurate and more robust river navigation of an unmanned boat. Additional work is also underway to develop classification and identification of boats and other objects in harbors and open sea.

## 9. ACKNOWLEDGEMENT

This work was performed under the sponsorship of DARPA-IPTO, PM Doug Gage #NBCHC020090NG; and ONR FNC-AO-IA, PM Marc Steinberg, #N00421-03-C-0027. We thank the sponsors for their support.

## 10. REFERENCES

1. S. Birchfield, "KLT: An Implementation of the Kanade-Lucas-Tomasi Feature Tracker," available online at: <http://www.ces.clemson.edu/~stb/klt/>.
2. J.Y. Bouguet, "Camera Calibration Toolbox for Matlab" available online at: [http://www.vision.caltech.edu/bouguetj/calib\\_doc/](http://www.vision.caltech.edu/bouguetj/calib_doc/).

3. Breazeal, C. and Scassellati, B., "A context-dependent attention system for a social robot". In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI99)*. Stockholm, Sweden, pp.1146-1151, 1999.
4. D.C. Brown, "Decentering Distortion of Lenses," *Photometric Engineering*, pages 444-462, Vol. 32, No. 3, 1966.
5. J.Bruce, T.Balch and M.Veloso, "Fast and inexpensive color image segmentation for interactive robots," *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '00)*, Vol. 3, October, 2000, pp. 2061 - 2066.
6. R.Collins and Y.Liu, "On-Line Selection of Discriminative Tracking Features," *International Conference on Computer Vision*, Nice, France, October 2003, pp.346-352
7. J. Costeira and T. Kanade "A Multi-body Factorization Method for Motion Analysis," in *Proc. 5<sup>th</sup> Int. Conference on Computer Vision*, (ICCV '95), June, 1995, pp. 1071-1076.
8. E. Hayman, T. Thórhallsson and D. Murray, "Tracking while Zooming using Affine Transfer and Multifocal Tensors," *International Journal of Computer Vision*, Volume 51(1) Jan 2003.
9. Itti L, Koch C, Niebur E. A Model of Saliency-Based Visual Attention for Rapid Scene Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254-1259, 1998.
10. Lucas, B. D. and Kanade, T. "An iterative image registration technique with an application to stereo vision," *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, Vancouver, pp. 674—679, 1981.
11. D.Nister, O.Naroditsky and J.Bergen, "Visual Odometry", *IEEE Conference on Computer Vision and Pattern Recognition*, Washington DC, 2004, pp.: 652-659.
12. A.Rowe, C.Rosenberg, I.Nourbakhsh, "A Low Cost Embedded Color Vision System," *International Conference on Intelligent Robots and Systems (IROS '02)*, Lausanne Switzerland, 2002.