

REFLICS: Real-time flow imaging and classification system

Sadahiro Iwamoto¹, David M. Checkley, Jr.², Mohan M. Trivedi¹

¹ Computer Vision and Robotics Research Laboratory, Electrical and Computer Engineering Department, University of California, San Diego, 9500 Gilman Drive 0407, La Jolla, CA 92093-0407, USA; e-mail: {siwamoto,mtrivedi}@ucsd.edu; Tel.: +1 (858) 822-0002, Fax: +1 (858) 534-1004

² Marine Life Research Group, Scripps Institution of Oceanography, University of California, San Diego, 9500 Gilman Drive, La Jolla, CA 92093-0218, USA; e-mail: dcheckley@ucsd.edu; Tel.: +1 (858) 534-4228, Fax: +1 (858) 822-0562

Received: 12 April 2000 / Accepted: 6 July 2000

Abstract. An accurate analysis of a large dynamic system like our oceans requires spatially fine and temporally matched data collection methods. Current methods to estimate fish stock size from pelagic (marine) fish egg abundance by using ships to take point samples of fish eggs have large margins of error due to spatial and temporal undersampling. The real-time flow imaging and classification system (REFLICS) enhances fish egg sampling by obtaining continuous, accurate information on fish egg abundance as the ship cruises along in the area of interest. REFLICS images the dynamic flow with a progressive-scan area camera (60 frames/s) and a synchronized strobe in backlighting configuration. Digitization and processing occur on a dual-processor Pentium II PC and a pipeline-based image-processing board. REFLICS uses a segmentation algorithm to locate fish-egg-like objects in the image and then a classifier to determine fish egg, species, and development stage (age). We present an integrated system design of REFLICS and performance results. REFLICS can perform in real time (60 Hz), classify fish eggs with low false negative rates on real data collected from a cruise, and work in harsh conditions aboard ships at sea. REFLICS enables cost-effective, real-time assessment of pelagic fish eggs for research and management.

Key words: Real-time machine vision system – Pipeline-based image processing – Plankton – CUFES – Fish egg sampling – Survey

1 Introduction

Scientific research, resource management, and the general understanding of life in the ocean depend on the collection and analysis of samples. Two important factors in such data collection and analysis are spatial resolution and temporal synchronization. High spatial resolution enhances our ability to investigate phenomena at smaller scales. This is particularly important when studying physical-biological interactions in the sea. Temporal synchrony in the collection

of data of various types is also important in a dynamic, biological system like the ocean, where phenomena are time dependent. To investigate the relationships between variables and thus allow strong inference about phenomena, sampling should be synoptic (e.g., a map of sea surface temperature from satellite) or performed in as short a time period as possible. These spatial resolution and temporal synchronization rules are appropriate to the estimation of the size of fish stocks by use of fish egg surveys and the daily egg production method (DEPM) (Lasker 1985). Fishes like anchovy and sardine spawn eggs in patches, of scale of hundreds of meters to tens of kilometers, which last for only 2–3 days (Smith 1973). For reasons of time and cost, egg samples are usually collected by towing nets at discrete stations every 4–40 km on a predetermined grid, with intervals of 20 min to several hours between samples. An entire cruise may take 2–3 weeks. Such spatial data are objectively interpolated between the sampling points to construct distribution maps. Since the data are undersampled (e.g., patches may occur between stations and thus go unsampled) and not synoptic, resource managers and scientists obtain only a rough estimate of fish stock size in this way. Continuous sampling and real-time analysis may enhance such estimates (Checkley et al. 2000b).

2 Research motivation and objectives

Significant progress towards high-resolution fish egg sampling was made when the continuous underway fish egg sampler (CUFES) was developed (Checkley et al. 1997). CUFES (Fig. 1) consists of a submersible pump (3 m depth, 600–800 l/min) and mesh filters (e.g., 500 μ m pore size) to concentrate and collect fish egg-size particles from the sea as the ship carrying it moves at full speed. Although water is continuously sampled by CUFES, samples are collected over discrete intervals, e.g., 5–30 min. The lower limit of the interval duration is determined by the speed of sample processing and analysis by humans. A CUFES collection interval of 5 min at a ship speed of 10 knots (\sim 5 ms) corresponds to a sampling interval of 1.5 km. Typically, these intervals are 20 min and 6 km, respectively. Ideally, one would like to know when and where each egg was sampled by CUFES to

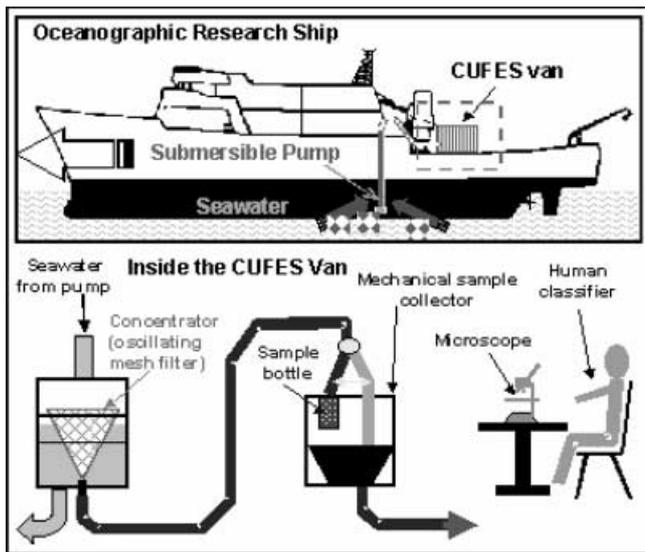


Fig. 1. Diagram of CUFES system onboard a research ship. Configuration shown has a submersible pump outside of the ship's hull 3 m below sea surface. An alternate configuration has a submersible pump inside ship's hull

achieve maximal sampling resolution. The real-time analysis of fish egg distributions also enables adaptive sampling and thus optimization of expensive ship time.

Replacing the sample collection and human analysis with a machine vision system to count and classify fish eggs in the CUFES flow will allow true continuous sampling and real-time analysis of fish eggs. But there are several challenges in realization of such a machine vision system. The system must deal with water flow of 10–20 l/min through a cross section of 2.85 cm² (1.9 cm diameter) and image objects (fish eggs) that are less than 3 mm in diameter. Furthermore, the water flow rate is not constant; it fluctuates due to ship movement (roll and pitch) and vibration of the CUFES concentrator and the ship's engine. Because of these conditions, the machine vision system must acquire and process images at high frame rates and brief exposure while synchronizing image acquisition to the flow. Another difficulty is the appearance of fish eggs in the ambient assemblage of similar-sized objects, including plankters and air bubbles. The machine vision system must accurately distinguish fish eggs from these other objects. Since the appearance of fish eggs is a rare event, the machine vision system must have acceptable a low false negative rate (a fish egg present but not detected). To be used as a sampling tool in management and science, the machine vision system must reliably detect and classify fish eggs equal to and better than current methods, while minimizing end-user cost (cost-effective). Finally, but not least, the whole machine vision system must be rugged and compact to operate in the harsh (e.g., salt corrosion, vibration, condensation) and space-limited environments aboard ships at sea.

Images shown in Figs. 2 and 3 illustrate various imaging and analysis problems a machine vision system faces. Thirty consecutive frames, equal to 0.5 s of video from a 60-Hz progressive scan camera, in Fig. 2 show a typical CUFES flow in a filmstrip fashion. A single egg of the Pacific sardine (*Sardinops sagax*) is imaged in frame 23. Large objects in

frames 14, 20, and 22 are copepods, and the object in frame 10 is a euphausiid, a type of plankton. Small air bubbles can be seen in some frames. Figure 3 shows how large air bubbles can be confused with fish eggs. Bubbles arise primarily from waves breaking in storms. A machine vision system must image, process, and analyze at 60 frames/s to achieve real-time detection and classification of fish eggs in the flow.

Our research in the design, development, and evaluation of REFLICS, a specialized machine vision system, has a specific and clear application domain context. This has resulted in assigning two separate perspectives in which the research and its contributions can be appreciated. The first perspective is that of the oceanographer or fisheries resource manager. We hope that REFLICS will provide them with a novel instrument for accurate and timely measurements that cannot be made using other available systems. Also, we think that several extensions and modifications of the basic REFLICS can be realized for other applications, such as medical and biological analysis of cells in flow and industrial inspection of fluids. The second perspective is that of the computer and machine vision community. Here, we hope that our research study has elements of value related to flow imaging and robust segmentation and classification algorithms as well as overall vision system architecture. The system performance requirements associated with accurate and high-speed imaging, processing, and classification of very small marine organisms (less than 3 mm diameter) in a high-speed water flow, with other small objects and air bubbles, pumped from within the ocean, and operation with high reliability onboard a ship, pose significant challenges to machine vision technology. In this paper, we present detailed design, development and evaluation of such a system. It has been tested and evaluated in the laboratory as well aboard ship during three fish egg survey cruises in the Pacific Ocean off the coast of California.

In the next section, we review machine vision work similar or related to REFLICS. In Sect. 4, we describe REFLICS's design concerning flow imaging, video processing, and algorithms. In Sect. 5, we present imaging, speed, and accuracy results from the current version of REFLICS. Finally, we conclude with future development work for REFLICS.

3 Review of relevant research

REFLICS is a real-time machine vision system that can be grouped into a broader category of automated visual inspection (AVI) systems (Newman and Jain 1995). REFLICS shares features common to other AVI systems in that it images, processes, and presents an analysis of objects being inspected. AVI systems differ by their application domain and REFLICS gains its uniqueness from its job of performing real-time assessment of fish eggs in CUFES. REFLICS must image small objects in high-speed flow, perform accurate segmentation and classification of fish eggs from other objects in real-time, and work accurately and robustly in a harsh environments (Iwamoto 1998). We describe three existing, AVI systems that image and analyze particles in flow for oceanographic purposes.

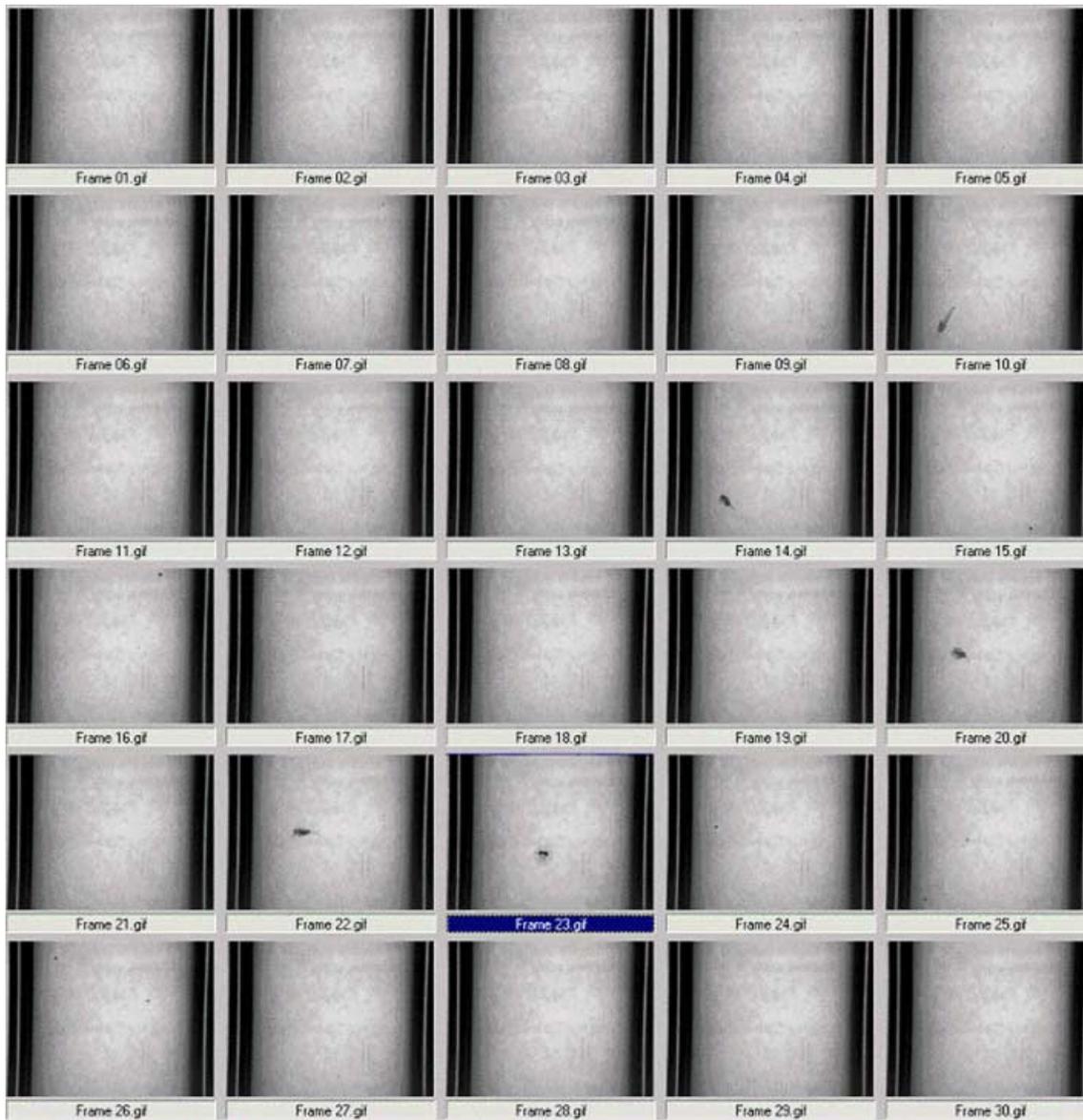


Fig. 2. Thirty consecutive frames of the CUFES flow from a 60-Hz progressive scan camera. A single sardine egg can be seen in *frame 23*. Other objects are copepods (*frames 10, 14, 20, and 22*) or air bubbles (*frames 16 and 26*)

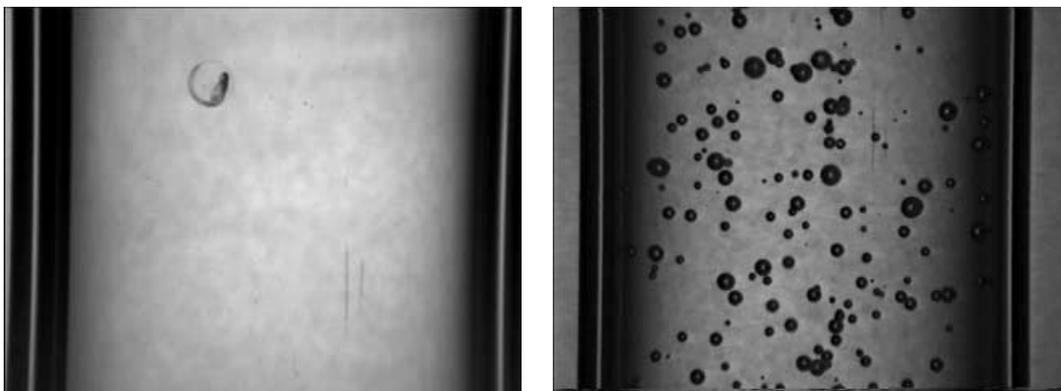


Fig. 3a,b. Two images of the CUFES flow. The *left image* shows a single sardine egg and the *right image* shows the CUFES flow full of air bubbles

Davis et al. (1997) have developed a machine vision system to image and classify oceanic plankters. The Video Plankton Recorder (VPR) images plankton *in situ* using a towed, dark-field imaging system. Video data are transmitted to the ship in real time. The VPR's artificial neural-network-based classifier can determine the family of imaged plankters mainly by contour quantified into Fourier shape descriptors (Tang et al. 1998). As the name implies, VPR's imaging and classification are specifically tuned for plankters that are opaque and vary in shape and are less suitable for transparent and round objects like fish eggs. Fish eggs cannot be differentiated from the background with dark-field illumination (dark background, bright object) and Fourier shape descriptors cannot distinguish between fish eggs and air bubbles.

Sieracki et al. (1998) have developed FlowCAM, a shore-based system to image microplankton. The size of an object in FlowCAM is limited to 20–200 μm and image capture is triggered by fluorescence level of objects (e.g., chlorophyll *a* in phytoplankton) in the field of view. FlowCAM can image the microplankton continuously to monitor the phytoplankton. FlowCAM is not suitable for fish eggs, because fish eggs do not fluoresce and are too large in size. Also, FlowCAM's imaging system cannot operate at speeds necessary for typical CUFES flow.

Tidd and Wilder (1998) describe a proof-of-concept prototype system for fish detection and classification. Their system images a volume $30 \times 30 \times 30$ cm every 4 s with a strobe. The system segments objects from the background and classifies these objects by size. The fish must be oriented in a certain position and direction for the imaging and classification to work. Obviously, the slow speed and the large size of target object makes this system unsuitable for fish eggs in CUFES flow.

In areas outside of oceanography, various real-time machine vision systems for inspection exist that share various features of REFLICS. For example, a research team at IBM has developed a produce (vegetable and fruit) imaging and automatic recognition system called VeggieVision for supermarket checkout use (Bolle et al. 1996). VeggieVision images the object passing through the register, segments the object from the background, extracts features (color, texture, histogram), and identifies the produce using a nearest-neighbor classifier. The system has been tested in an actual supermarket and can display in real time the identification of produce to the cashier. VeggieVision is meant to be used as a tool to help the cashier identify produce, not as a human replacement. Similarly, REFLICS would be used as a tool to extend the capabilities of its user and not entirely replace the human analysis in the fish egg survey process.

Imaging of small objects in the flow has been heavily developed for biological and medical flow cytometry, a technique for counting and sorting cells (scale: 10 μm). For example, Hüller et al. (1991) describe a PC-based image flow cytometry system. The system images cells flowing through a cytometry tube, processes the image, and saves the image to the hard disk. No object analysis is performed but the real-time imaging and processing of objects in flow cytometry is similar to REFLICS.

The general method that REFLICS uses to detect and identify fish eggs in flow images is a two-step approach.

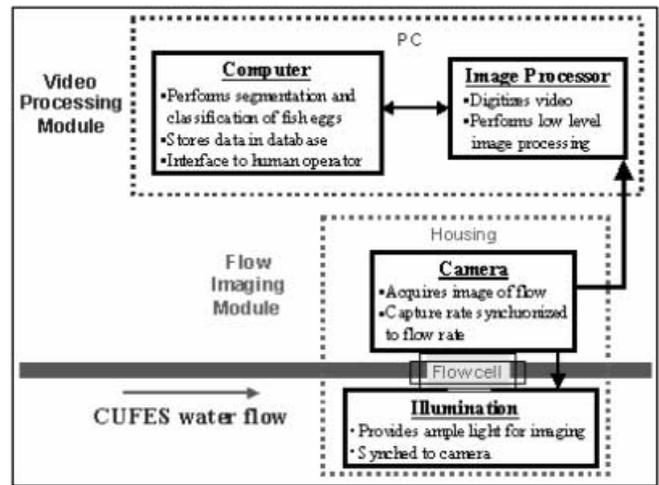


Fig. 4. REFLICS block diagram

First, it segments objects from the background image using low-level image-processing techniques. Second, it extracts features from segmented objects to classify them. This bottom-up approach to object detection and identification has been used in various applications, from aerial image analysis (Trivedi 1990) to automated potato grading (Heinemann et al. 1996). REFLICS implements this approach for finding objects in a flow illuminated by backlighting, extracting features suited for the identification of fish eggs and distinguishing them from other objects in the flow, and finally classifying these objects using the extracted features. Furthermore, REFLICS implements this approach rapidly, in real time, on a dual-processor computing platform with a specialized image-processing board.

4 Integrated system design

REFLICS is an integrated system consisting of a number of modules working together to achieve the objective of imaging, detecting, and classifying fish eggs in the CUFES flow. A block diagram of these modules and their interactions are shown in Fig. 4. The equipment diagram for REFLICS is shown in Fig. 5.

REFLICS is designed to address the challenges described in the introduction section for a CUFES machine vision system. The flow is imaged through a glass flow cell (imaging tube) about 20 mm in diameter with an appropriate optical lens to obtain enough resolution for fish-egg-sized objects and enough depth of field for the flow cell tube. The camera is a high-speed progressive-scan camera that can handle high-speed object motion. The illumination source is a powerful strobe synchronized to the camera in a backlighting configuration. The strobe illumination allows the camera to acquire a sharp (no motion blur) image of the flow. The flow imaging module is enclosed in a rugged housing to protect sensitive imaging equipment from the harsh outside environment. The Datacube MaxPCI™ pipeline-based image processor digitizes the analog video signal and performs low-level image-processing functions to very quickly segment and clean objects from the background. The host computer completes segmentation and performs classification on possible fish egg objects. The classifier uses many

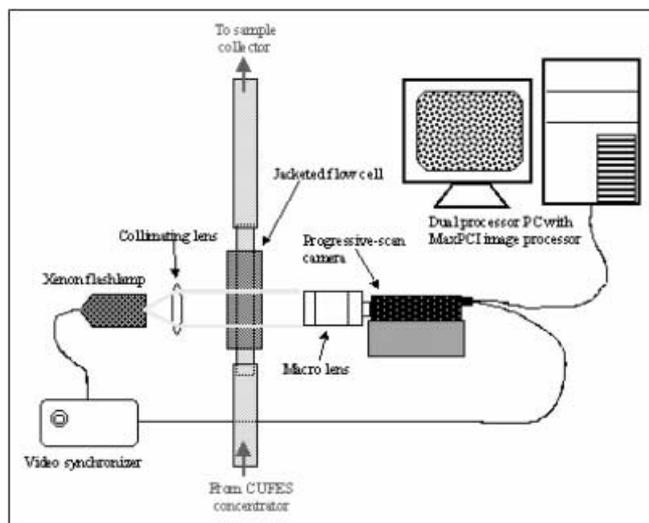


Fig. 5. REFLICS equipment diagram

features and a nearest-neighbor classifier to correctly identify fish eggs from other objects. The host computer is a fast dual-processor system to process the 60 frames/s video stream.

In the rest of this section, REFLICS's modules and components will be presented in more detail.

4.1 Flow imaging module

REFLICS's flow-imaging module must deal with small, translucent objects in a fast and dynamic flow and operate in a harsh environment. A backlit spherical fish egg looks round with a ring-like outer contour (chorion), translucent interior (perivitelline space and yolk sac), and a dark embryo. The dark embryo differs in size depending on the age of the egg and the view. The attached yolk sac is largely translucent. Figure 9 illustrates these features. The flow imaging module incorporates features to work with these conditions. This module is based, in part, on the *in situ* plankton camera of Ortner et al. (1981). Their system used strobe illumination in backlighting configuration, and images were captured on photographic film for later development and visual inspection.

The CUFES flow passes through a glass flow cell that is the same shape (circular) and size (20 mm diameter) as the tubing from the CUFES concentrator. The glass flow cell is jacketed with a glass box that is filled with water at room temperature. The water jacket serves two purposes. First, the planar face of the water jacket minimizes the refraction effect of the circular glass tube. Second, the water jacket reduces condensation problems when CUFES flow is very cold relative to room temperature. REFLICS's glass flow cell can be seen in Fig. 6.

Since the object of interest is small in size, the camera must fully image the flow cell to obtain the highest resolution possible. The flow-imaging module uses a macro lens that provides the appropriate magnification to image the 20-mm wide circular tube and has adjustable iris to obtain enough depth of field through the tube (20 mm depth). Since the flow-imaging module looks at a small field of view, objects



Fig. 6. Side view of REFLICS's glass flow cell. CUFES flow goes in and out from tubing protruding left and right. Water for the jacket flows in and out from smaller tubing protruding diagonally

in the flow move across the field very quickly. The flow-imaging module must be able to image every fast-moving object without blur. REFLICS uses a progressive-scan area camera (Pulnix 6701AN) as its image sensor that provides high-speed 60-Hz full-frame acquisition. To prevent motion blur, a xenon flashlamp strobe is synchronized to the camera's sync signal. The strobe generates an intense light pulse ($< 20 \mu\text{s}$) every $1/60$ s to "freeze" the objects in the field of view. Backlighting configuration (bright field illumination) is used to create a bright background and dark objects. Backlighting allows the fish egg's translucent interior to be imaged and air bubbles to be opaque in the image (Schroeder 1984).

To allow REFLICS to be used aboard ships, the flow-imaging module and support electronics are mounted in a custom-designed housing constructed from aluminum and PVC. The design incorporates features to minimize size such as using a first-surface mirror to deflect the image by 90 degrees. The CUFES flow is oriented upward in the housing to minimize bubble accumulation in the tubing. The interior of the ruggedized housing is compartmentalized to isolate the optics and electronics from the flow cell in case of water leakage. Cameras and illuminators are bolted down securely to prevent movement while in operation. The housing itself is mounted on shock-absorbing material to dampen vibrations from nearby the CUFES concentrator and ship's engines. Figure 7 shows the flow-imaging module of REFLICS installed in the housing aboard a research ship.

4.2 Video processing module

Video processing in REFLICS must be performed in real time to generate fish egg counts to the users at sea. To do so requires REFLICS to process every frame from the imaging module (60 frames/s). Since fish eggs are usually rare, REFLICS must miss as few eggs as possible (very low false negative rate). Every frame of video must be analyzed for fish eggs, which requires REFLICS to have a computing platform with a considerable amount of processing power to analyze the large amount of data the flow-imaging system



Fig. 7a,b. REFLICS flow-imaging module. **a** The housing (closed with gray PVC panels) mounted next to the CUFES's concentrator. **b** Inside of the housing, showing camera, flow tube, and illumination

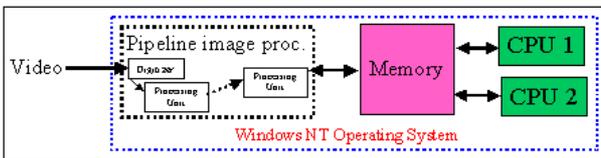


Fig. 8. Architectural diagram of REFLICS's video-processing module. The box with *outer dotted line* signifies the PC

produces. On the other hand, the computing hardware must be affordable to the end user.

REFLICS uses a dual-processor Intel Pentium™ II 350-MHz PC as its base system with a pipeline-based image processor card, Datacube MaxPCI™, to digitize the video and perform low-level image processing. Using a PC platform has several advantages. Current PCs have a tremendous amount of processing power, yet they are relatively low in cost. The Microsoft Windows NT operating system (OS) for the PC provides a graphical user interface (GUI) environment, easy-to-use development tools, and support for multiple processors at consumer-level cost. And finally, PCs are easy to maintain. The MaxPCI™ incorporates a high-speed image digitizer and performs low-level image processing in a pipeline fashion. Pipelining allows low-level image-processing functions, which do not have complex calculations but high data rates, to be performed in real-time. REFLICS's dual-processor PC and MaxPCI™ is a powerful video-processing module providing the required performance, while minimizing equipment, development, and maintenance costs.

Figure 8 shows the architectural diagram of REFLICS's video-processing module. The processed image stream from the MaxPCI™ is transferred to the PC's main memory. The data contained in the memory is accessible by the PC's two CPUs. The two CPUs work in parallel to further process the image stream to obtain the count of fish eggs in the video.

4.3 Object detection and identification algorithms

REFLICS uses a bottom-up approach of segmentation and classification to detect and identify fish eggs in CUFES flow

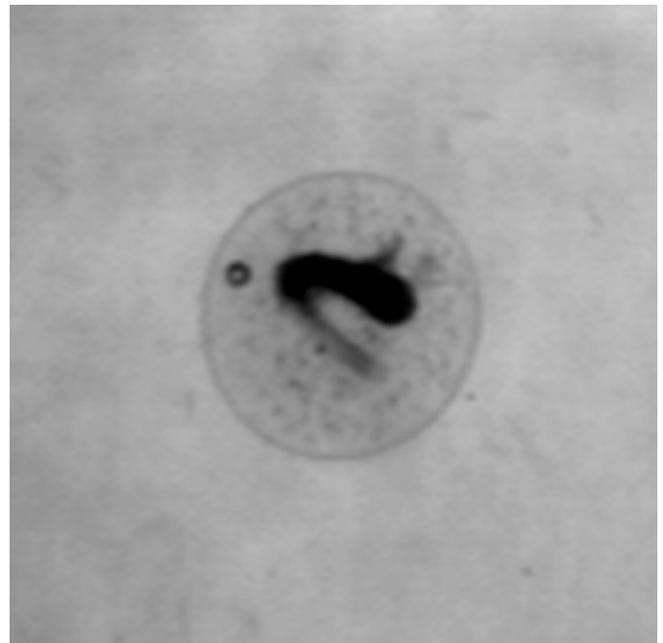


Fig. 9. Magnified view of a Pacific sardine egg from a REFLICS image

images. Specifically, REFLICS segments moving objects in the flow image and classifies objects as fish eggs or not using features extracted from these images.

Figure 9 shows an egg of a Pacific sardine late in its development imaged by REFLICS's imaging module. Features that are used to distinguish fish eggs from other objects in the flow are its size, its round shape, and its translucent interior with some dark areas. Since REFLICS needs to segment and classify in real time (60 frames/s), we choose segmentation and classification algorithms that are adequately robust and accurate while being fast on the available processing hardware.

The segmentation algorithm accepts images from the camera and finds regions of interest in the images which could be fish eggs. It does so by performing the following steps on each frame:

1. absolute differencing with the temporally stationary image to show only moving objects (background subtraction);
2. thresholding to binarize the image into foreground and background pixels;
3. morphological filtering to remove small noise pixels counted as foreground pixels;
4. updating temporally stationary image used in step 1 (only areas containing background pixels);
5. geometry-preserving run length (GPRL) encoding to mark lines containing foreground pixels; and
6. connectivity to group foreground pixels as labeled objects.

Background subtraction is a robust technique to isolate moving object pixels in the image from the temporally stationary image. In REFLICS, the temporally stationary image is the image of the backlit flow cell, which varies very slowly over time or not at all. REFLICS uses absolute differencing to remove the temporally stationary image as expressed in the equation below. I is the input flow image, BG is the temporally stationary image, and D is the background-subtracted, or the moving object, image.

$$D(x, y) = |I(x, y) - BG(x, y)| .$$

A statistical approach to background subtraction provides a more accurate result, but we used the absolute difference for two reasons. First, the statistical approach requires significant processing power that cannot be performed on the MaxPCTM image processor. The statistical background subtraction would need to be performed on the PC's CPUs and real-time segmentation would not be possible. Second, since the camera, flow cell, and illumination are fixed in position, the temporally stationary image changes little over time. Absolute differencing is sufficient and statistical background subtraction would only provide marginal improvement for REFLICS.

Thresholding binarizes the background-subtracted image into foreground and background pixels. Foreground pixels are pixels belonging to the objects moving with the flow (pixel value 1). Background pixels are pixels belonging to the temporally stationary image (pixel value 0). The operation is described by the following conditional equation, where B is the thresholded binary image and T is the threshold value.

$B(x, y) = 1$ if $D(x, y) > T$, else $B(x, y) = 0$, where $B = 1$ is foreground and $B = 0$ is background.

Since moving objects in the background-subtracted image D only have non-zero grayscale values, thresholding splits the image between foreground pixels for moving objects and background pixels for the stationary flow cell and illumination. Theoretically, if the temporally stationary image is absolutely stationary, the threshold value T can be set to 0. Due to electrical noise from sensor, cable, and digitizer, slight fluctuation in illumination, and physical vibration, the threshold value is set between 5 and 10 to limit the amount of noise pixels to be counted as foreground pixels.

The combination of background subtraction and thresholding works well to separate images of fish egg from the flow cell. Fish eggs are either opaque or translucent with

backlighting. There is enough contrast from the temporally stationary image for most pixels belonging to fish eggs to be counted as foreground pixels. Background subtraction and thresholding fail when fish eggs are imaged in a region of the flow cell where the field is dark (e.g., left and right edges of flow cell tubing due to refraction). In such cases, fish eggs do not contrast enough with the temporally stationary image and are not separated. REFLICS deals with this problem by creating a region of interest (ROI) and only analyzing for fish eggs in the ROI, and ignoring areas outside the ROI. Before running REFLICS, the operator will specify a ROI window in the flow image where fish eggs will be searched. The ROI window will exclude regions outside the flow cell and regions where the field is dark. Some fish eggs might be missed (false negative) by ignoring the dark areas of the flow cell, but the error can be estimated and corrected by comparing with human analysis data and statistical methods.

Morphological filtering by an erosion kernel removes very small foreground pixels that are still left in the image which are usually noise. The equation for the erosion operation is below. B is the binary image and K is the structuring element. For erosion, the structuring element is a 3×3 square kernel block of ones. The erosion operation has the effect of removing any foreground pixels that touch the background. This shrinks large objects by one pixel around the perimeter and also removes small noise pixels.

$$B \ominus K = \{(x, y) | K_{(x,y)} \subseteq B\} \text{ where } K = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} .$$

The temporally stationary image for background subtraction is updated to adapt to slight changes. The temporally stationary image is modified after processing each new frame by the following equation. BG_t is the current temporally stationary image and BG_{t+1} is the new temporally stationary image.

I is the input flow image and B is the thresholded binary image from above.

$$BG_{t+1}(x, y) = BG_t(x, y) + c$$

$$\begin{cases} c = 1 & \text{if } I_t(x, y) > BG_t(x, y) \text{ and } B_t(x, y) = 0 \\ c = -1 & \text{if } I_t(x, y) < BG_t(x, y) \text{ and } B_t(x, y) = 0 \\ c = 0 & \text{if } I_t(x, y) = BG_t(x, y) \text{ or } B_t(x, y) = 1 . \end{cases}$$

The temporally stationary image from the flow-imaging system does not significantly change over a short period of time but, over long time, it can change due to such factors as condensation, slight shifts due to vibration, and weakening light output from the strobe. When deployed, REFLICS would be required to work continuously for days. An adaptive feature such as this updating method is required.

GPRL encoding reduces processing time on the next operation, connectivity, by determining whether a row contains foreground pixels or not (Datacube 1994). Since the MaxPCITM image processor has dedicated hardware to perform GPRL encoding, there is no reduction in performance. GPRL encoding counts the number of background-to-foreground and foreground-to-background transitions for each line. Noting the number of transition for each line allows the connectivity operation to skip checking lines without

transitions. Each pixel is also encoded with a distance from the recent transition. This distance information allows the connectivity operation to find transition locations without checking every pixel in the image.

The connectivity operation takes the GPRL-encoded image and groups individual foreground pixels into a labeled object. An 8-way neighbor, block-based method is used for speed and accuracy. Each labeled object is stored as a list of foreground pixel blocks with a unique number label.

After obtaining segmented and labeled objects in the flow image, the classifier needs to determine whether or not each object is a fish egg. REFLICS performs classification by following these steps:

1. extract area feature from objects;
2. use size filters to eliminate objects too big or too small to be fish eggs;
3. extract other features, shape and histogram, from the remaining objects; and
4. use a nearest-neighbor classifier to determine whether object is a fish egg.

The object feature extraction operation first extracts only the area of objects. This is done by counting the number of pixels in each object. A bounds filter eliminates some of the object blobs by using the fact that the target fish eggs are at least and at most of a certain size. Thus, remaining small noise objects and very large objects are eliminated and processing power is not wasted in subsequent steps on these objects. Equations for these two steps are shown below. $Area_{object}$ is the size of an object in pixels. $Area_{min}$ and $Area_{max}$ are the minimum and maximum sizes allowed for fish eggs in the flow image. Minimum and maximum sizes are chosen by the REFLICS operator before running the system. Values would depend on how large fish eggs are in the REFLICS flow image.

$$Area_{object} = n_{object \text{ pixels}} = \sum_{\text{all object pixels}} 1$$

Remove *object* if $Area_{object} < Area_{min}$
or $Area_{object} > Area_{max}$

The remaining objects are put through another object feature extraction operation. This time, shape (contour and roundness/compactness) and histogram features are extracted. Using the object pixel information, a chain code of an object's outside contour is generated. The perimeter is measured from the chain code and roundness (or compactness) for the object is calculated by the following equation:

$$Roundness_{object} = \frac{Perimeter_{object}^2}{4\pi Area_{object}}$$

$Roundness_{object}$ is the quantitative measure of the object's roundness. A circular object like a sardine egg or an air bubble would have a roundness value close to 1 and an aspherical object like a copepod would have a value greater than 1. $Perimeter_{object}$ is the outside perimeter of the object and $Area_{object}$ is the area of the object.

An object's histogram is calculated from object pixels and the original grayscale image of the object. In the histogram calculation, only grayscale image pixels correspond-

ing to the object pixels are used. Non-object pixels (background pixels) are not counted in the histogram. The average intensity of the object is also calculated while calculating the histogram by summing the grayscale values of object pixels and dividing the sum by the total number of object pixels.

Finally, all these features (width, height, width-height ratio, area, roundness, histogram, average intensity) are used in a nearest-neighbor classifier to identify whether or not the segmented object is a fish egg of a specific type. All features for the unknown object are placed in a 262-element feature vector (256 elements for histogram and 6 other features) and the weighted Euclidean distance measure is calculated from the object's feature vector to feature vectors of N number of prototypes. The prototype with the closest distance to the object's feature vector is chosen as the identity of the object. The nearest-neighbor classifier adapted for REFLICS is expressed below where x is the unknown object, N is the number of prototypes, $d(\dots)$ is the Euclidean distance measure, w is the weight vector, f_x is the unknown object's feature vector, and f_j is the feature vector for the j -th prototype:

Unknown object x assigned to class j , where

$$\arg \min_{j=1 \text{ to } N} d(\vec{w} \cdot \vec{f}_x, \vec{f}_j)$$

The distance measure is weighted to normalize the features. For example, the area feature is larger in magnitude than the width-height ratio. The weight vector is also used to place emphasis on some feature element. The weight vector is determined initially by normalization and then is adjusted based on knowledge about fish egg classification and trial and error.

There are three main categories of prototypes corresponding to the three most common objects encountered in the image: fish eggs (spherical, translucent), air bubbles (spherical, opaque), and other types of particles. The fish egg category can be broken into many prototypes depending on species and age. The initial prototypes are generated by running REFLICS without the classifier turned on. The human operator classifies objects segmented by REFLICS, and these objects and their features are used as initial prototypes for REFLICS. When REFLICS is running with the nearest-neighbor classifier turned on, the prototypes are continuously updated. Newly classified object are compared with existing prototypes, and if the new classified object more evenly distributes that class, it is added as a new prototype for that class. Each class will be limited to a certain number of prototypes. If the addition of a new prototype exceeds the limit, the least used (has not been closest to an unknown object recently) prototype in that class is discarded. Feature extraction and nearest-neighbor classifier provide a fast, accurate, and robust identification of fish eggs.

4.4 Algorithm implementation

The processing algorithms are coded to take maximum advantage of the camera rate processing of the pipeline image processor and PC's two processors. Data-intensive calculations are mostly performed on the image processor and

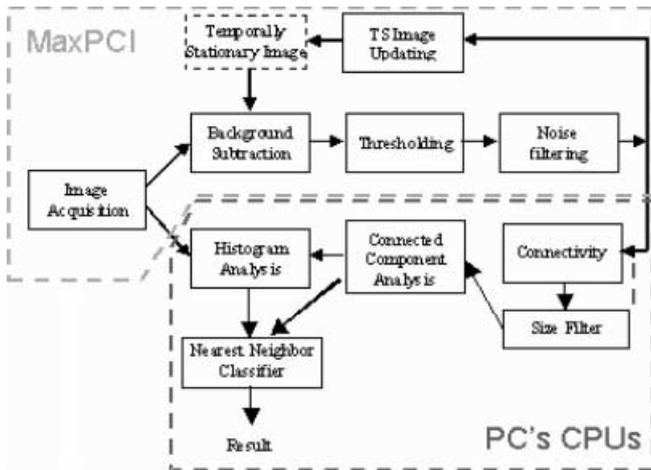


Fig. 10. Flow chart of REFLICS's segmentation and classification algorithms. Processing hardware used to perform the algorithms is denoted as well

complex calculations (connectivity, feature extraction, classification) are performed on the PC's two processors. The code is multithreaded to allow the two CPUs to work in parallel. Also, buffers and thread synchronization mechanisms are used between major processing steps to maximize processor usage and to process continuously without dropping frames. Figure 10 is a flow chart of segmentation and classification algorithms. It shows where each step occurs on the processing hardware.

The pipeline image processor, Datacube MaxPCI™, is a PC board consisting of an analog digitizer, triple-ported memory modules, processing units, and a sophisticated programmable interconnect. The MaxPCI™ is programmed by taking an image-processing algorithm and mapping it onto pipeline pipes. A pipeline pipe consists of an image data source memory, image data destination memory, processing units to perform the image processing, and the data-routing information for the interconnect. If the image-processing algorithm is too large for a single pipe, it can be spread out over several pipes. If more image-processing algorithms are required, they can be implemented in new pipes. Using more than one pipe incurs latency in the algorithm implementation (similar to a reduced instruction set computer's (RISC) pipeline latency), but images are still processed at camera rate. Once the initial latency passes, one processed image comes out of the MaxPCI™ for every input image going into it.

REFLICS's segmentation algorithm on the MaxPCI™ was implemented as shown in Fig. 11. There are four pipes in this implementation: acquisition (Pipe 1), background subtraction/thresholding/erosion/temporally stationary (TS) image updating (Pipe 2), GPRL encoding (Pipe 3), and image transfer (Pipe 4). Pipe 1 is a simple transfer of image data from the analog digitizer, which is connected to the REFLICS's imaging module, to a MaxPCI™ memory module.

Pipe 2 is complex and performs five tasks. The pipe takes the acquired image and performs background subtraction (absolute differencing) with the temporally stationary image, thresholds the background-subtracted image, erodes the thresholded binary image, and stores the resulting image in a memory module. The eroded and thresholded image is

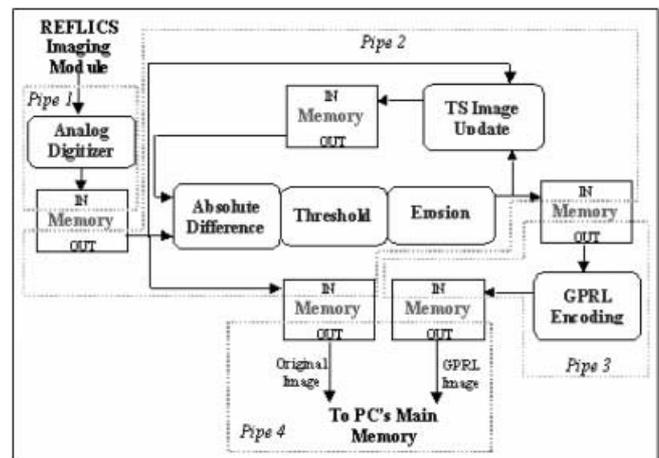


Fig. 11. REFLICS's segmentation algorithm implementation on pipeline-based image-processing hardware

used with the acquired image to update the temporally stationary image. The updated temporally stationary image is stored in the memory module dedicated to the background image. Finally, a copy of the acquired image is stored in another memory module.

Pipe 3 takes the binary image and generates a GPRL-encoded image. The GPRL image is stored in a memory module. Pipe 4 transfers the GPRL-encoded image along with the original acquired image saved in Pipe 2 to the host PC's memory. Further processing on the host PC completes the segmentation algorithm.

A pipeline processor implementation of the REFLICS segmentation algorithm incurs an initial latency of four frames (including one frame to transfer to the PC memory) but, after that delay, video is processed at a constant camera rate of 60 Hz. A pipeline processor limits the type of processing that can be performed but allows REFLICS to maintain the required processing speed for a large video data rate (~ 18 MB/s).

Figure 12 shows how segmentation and classification occurs step by step on three images. One image contains a fish egg, one contains a copepod, and another contains a small air bubble (a). Most of the flow cell image is eliminated with background subtraction (b) and thresholding (c). Erosion eliminates small noise pixels in images (d). The size filter eliminates the small round air bubble (e). The nearest-neighbor classifier eliminates the non-spherical copepod (f). The fish egg passes through and is correctly identified by REFLICS.

5 Experimental results and analysis

REFLICS is an integration of several components. Individual modules as well as the integrated system were extensively tested in our laboratory and shipboard at sea during research cruises. The imaging section has been fully developed and repeatedly tested in the lab and at sea. Computing hardware has been assembled, interfaced to the imaging system, and configured. The segmentation and classification algorithms have been prototyped and tested on images saved to hard disk and on Hi8 videotape. Real-time segmentation

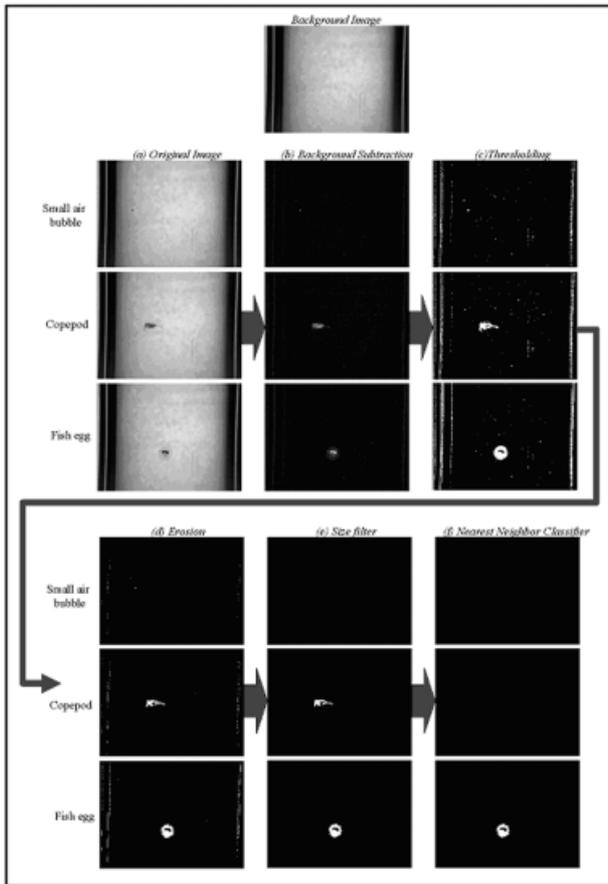


Fig. 12a-f. Illustration of algorithmic process flow in REFLICS segmentation and classification process. **a** Three images, an air bubble, a copepod, and a fish egg are inputs to the process. **b** Background-subtracted images are produced from the input images and the temporally stationary image. **c** The background-subtracted images are binarized to segment out separate moving objects. **d** Erosion removes small noise pixels. **e** Connectivity followed by size filter removes objects and noise smaller than fish eggs (air bubble removed). **f** Blob analysis (shape, histogram) followed by nearest-neighbor classification isolates fish egg (copepod removed). Only the fish egg remains

has been implemented and tested in the lab and at sea. The real-time nearest-neighbor classifier has been integrated into REFLICS and tested in the lab and at sea. To date, we have results and data from the imaging system, the classification algorithm, REFLICS's real-time segmenter, and REFLICS's integrated real-time segmenter and classifier from laboratory experiments, post-processing of data collected at sea with the REFLICS imaging module, and tests conducted at sea with the actual integrated REFLICS.

5.1 Experimental evaluation

To perform imaging system development and evaluation in the lab, we built a flow generation testbed (Fig. 13). Water flows by gravity from an upper 30-gal tank to a lower 30-gal tank and is pumped back up. The testbed can simulate flow between the concentrator and sample collector in CUFES up to 25l/min. Fish eggs, glass beads, and other objects can be introduced into the simulated flow. The REFLICS's imaging

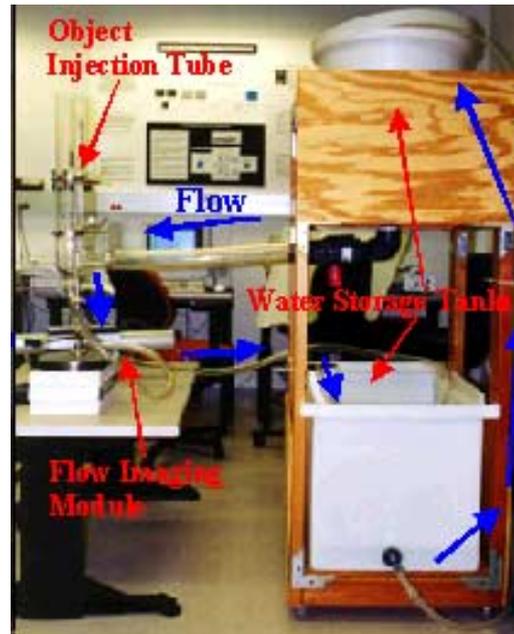


Fig. 13. Laboratory-based REFLICS flow testbed

system and real-time segmentation were tested using live eggs of white sea bass (*Atractoscion nobilis*) and glass beads in the testbed.

In addition to the flow testbed, we used CUFES flow images recorded on Hi8 videotape (30-Hz interlaced camera). We also wrote a Windows program for real-time display and capture to disk of video, and we have used this software and the imaging system to collect 60-Hz progressive-scan CUFES flow images. These two types of real flow images were used to test the prototype segmentation and classification algorithms.

These tests were useful in allowing us to refine the system hardware and software so that we could deploy REFLICS aboard three fish egg surveys in 1999 (Fig. 14) and two surveys in 2000. Ship-based studies involve significant efforts and preparations; however, without them, it is not possible to refine the system design and to assess its performance. We performed three sea tests for REFLICS development and testing in 1999. REFLICS's imaging module and real-time segmenter were tested at sea. Experiments conducted and data collected from these sea tests were used for further developing REFLICS in the lab.

5.2 Results and analysis

REFLICS's imaging system was tested off the coast of California during April 1999, the spawning season of the Pacific sardine. Using a high-capacity hard disk, we recorded several tens of thousands of real-time CUFES flow images at 60 frames/s. Figure 15 shows several images with fish eggs and other plankton. The imaging system has been shown to provide sufficient field of view and resolution. The 20-mm flow cell tube fills the width of the frame (636 pixels). A 1.5-mm diameter sardine egg is covered by a bounding box of approximately 50×50 pixels. The interior of a fish egg is translucent except for the embryo which is opaque. The

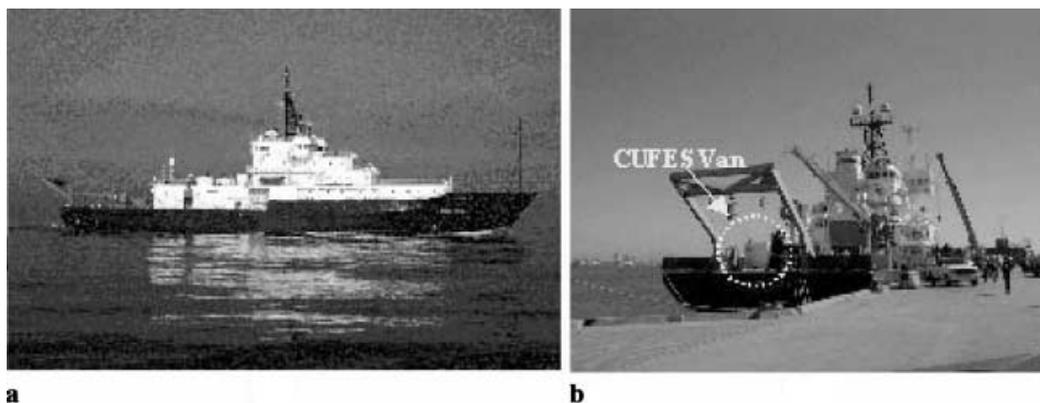


Fig. 14a–c. Testing REFLICS on a research cruise. **a** R/V Roger Revelle. **b** CUFES van and R/V Roger Revelle during installation on docks in San Diego. **c** REFLICS in CUFES van



Fig. 15. Images of sardine eggs (*top left and right*), copepod (*bottom left*) and euphausiid (*bottom right*)

outer egg membrane and embryo are resolved with enough detail for image analysis. Other objects, such as air bubbles, appear opaque. The frame rate of the system is fast enough to work with CUFES flow rates up to 15 l/min.

We tested REFLICS’s segmentation performance by imaging a moving test pattern with the 60 frames/s pro-

gressive-scan area camera and measuring the processing speed. Using a pattern that resembles a typical flow condition, i.e., a steady background with two to eight 50×50-pixel objects, the segmenter maintained a steady 60 frames/s processing speed. We increased the number of objects, with the processing speed held at 60 frames second, up to 15 objects

Table 1. REFLICS's segmentation speed versus number of fish-egg-sized objects

Number of moving objects	0	2	4	6	8	10	15	20	25
Processing Speed (Frames/s)	60.0	60.0	60.0	60.0	60.0	60.0	60.0	59.6	59.1

Table 2. Comparison between the number of fish eggs physically collected and identified by an expert and the number of fish eggs identified by REFLICS from a selected site during the April 1999 fish egg survey. Each run was 1 min in length

Eggs detected by REFLICS	14	16	11	16	12
Actual eggs collected and counted by an expert	15	18	11	17	10

(Table 1). Maximal concentrations of anchovy and sardine eggs in spawned patches are < 100 eggs/m⁻³ (Checkley et al. 2000a, b). Rarely are more than several plankters and never have more than two eggs been seen in a single frame. Thus, the segmentation algorithm, multithreading, and multiprocessor/image processor lets REFLICS achieve real-time segmentation under typical conditions of CUFES flow and the ambient plankton. The exception is when many bubbles occur in a single frame.

Collected CUFES flow images were used to test the non-real-time classifier. Instead of a video source, the classifier implementation used PPM image files (uncompressed grayscale image with header information) stored on the hard disk as the input image stream. Figure 16 shows a map of the abundance of Pacific sardine eggs from CUFES for a cruise in April 1999. Eggs were collected and counted at sea by experts. Comparison of results from human analysis and the non-real-time classifier were highly correlated (correlation coefficient = 0.9598) (Table 2). Discrepancies in number may be due to sample collecting and image capture not being properly synchronized. This will not be a problem in the eventual integrated onboard operation of REFLICS.

Using 3600 frame segments (1 min) containing sardine eggs and other objects captured during the April 1999 cruise, we tested the accuracy of the REFLICS non-real-time classifier with human analysis of the flow images. The results from the five segments are shown in Table 3. We used three prototypes: sardine egg, copepod, and salp. We did not include air bubbles, since none were present in these flow video segments.

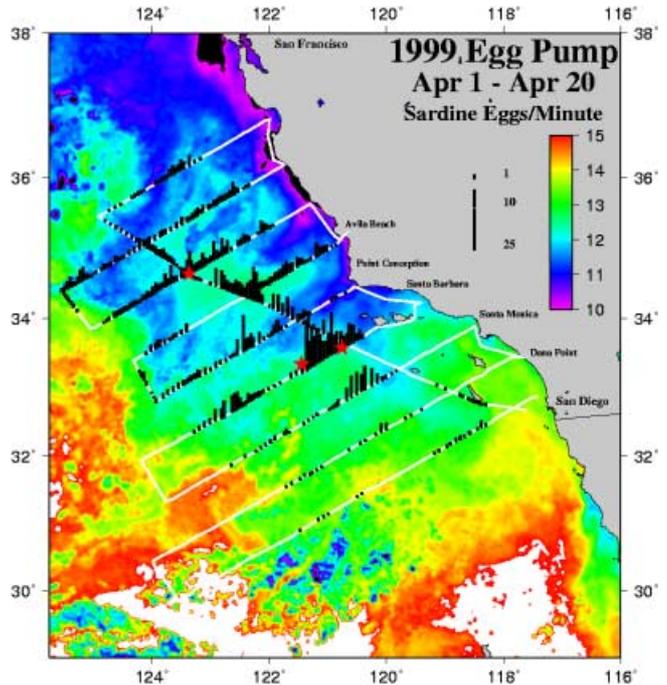
The non-real-time classifier achieves excellent true positive rate (egg present and detected) and the necessary false negative rate (egg present but not detected).

6 Concluding remarks

We presented REFLICS, a real-time machine vision system to detect and classify fish eggs in flowing water. REFLICS images, segments, and classifies at the speed necessary for objects in the flow of CUFES between the concentrator and sample collector. A progressive-scan area camera with a custom-designed flow cell and strobe backlighting allows

Table 3. Comparison of REFLICS non-real-time classifier with human analysis of flow images

Test #	Eggs imaged in video segment	True positive	False negative
1	14	14	0
2	16	16	0
3	13	11	2
4	19	16	3
5	12	12	0

**Fig. 16.** A map of the results from cruise 9904RR of CalCOFI, the California Cooperative Fisheries Investigation). CUFES sardine egg survey data are overlaid on sea surface temperature (SST). White lines indicate ship's cruise track. Height of vertical, black bars is proportional to the concentration of Pacific sardine eggs. Stars indicate collection locations of samples and images used in the segmentation and classification tests (summarized in Table 3). Egg counts are those made at sea and thus preliminary (courtesy R. Charter, Southwest Fisheries Science Center, NOAA)

the imaging of fish eggs in CUFES flow. A rugged housing allows the imaging system to function aboard ships at sea. A combination of background subtraction, morphological filter, GPRL encoding, connectivity, and background updating enables REFLICS to segment objects from the flow images at 60 frames/s using a dual-processor PC and a MaxPCI™ pipeline image processor. The nearest-neighbor classifier using moments, contour, and histogram features can identify fish eggs in the segmented images.

REFLICS can be improved in several ways. We are planning on implementing a neural-network-based classifier. The classifier will use the original raw image of the object as well as the extracted features to classify objects. A neural-network-based classifier such as a multilayer perceptron with back-propagation learning will allow supervised training from examples, classification using undefined features in the image, as well as human-defined features (area, average intensity, etc.), and REFLICS classification to be more robust. To train the neural-network classifier, we would

require many more fish egg images captured in subsequent survey cruises. We are also planning to shift the entire processing to the PC's CPU. Instead of dedicated and expensive image-processing hardware, the new REFLICS will use a framegrabber board and a fast multiprocessor PC. This configuration would further reduce the cost of REFLICS and enhance its versatility. The high-data-rate processing will still be an issue, but it would not be as significant as before with new technology such as faster processors (higher clock speed and new features like SIMD (single-instruction-multiple-data), faster memory (DDR-SDRAM, RAMBUS), and faster buses (64-bit PCI and AGP).

REFLICS will be a cost-effective tool for resource managers, oceanographers, and marine ecologists to obtain accurate, high-resolution, and real-time data on fish egg distribution and abundance in fish eggs surveys and studies. Such data, particularly when combined with ancillary data on the environment (e.g., water temperature, salinity, and chlorophyll a fluorescence), will be valuable for the assessment of stock size, spawning habitat, biology, and ecology of fish, for both basic and applied ends.

Acknowledgements. This research is funded by a grant from the National Sea Grant College Program, National Oceanic and Atmospheric Administration, U.S. Department of Commerce, under grant number NA66RG0477, project number 74-C-N through the California Sea Grant College System. The views expressed herein are those of the authors and do not necessarily reflect the views of NOAA or any of its sub-agencies. The U.S. Government is authorized to reproduce this paper and distribute it for governmental purposes. The authors would like to thank K. Alexandra Curtis, David Levenson, Dr. Kim Ng, Nils Lassiter, Kirk Ireson, Catherine Johnson, and Richard Capella for their participation and work on various aspects of the REFLICS project. The authors would like to thank the people of 9901RR, 9904JD, and 9909NH CalCOFI cruises, especially Ronald Dotson, David Griffith, Amy Hayes, and Thomas Hayward for allowing us to perform key developmental work at sea. We would also like to thank Hubbs-Sea World Research Center for providing live fish eggs for development work and Dr. Arun Dalmia of Datacube Inc. for his support during the initial phase of this project. Dr. Peter Ortner and Shailer Cummings of the Atlantic Oceanographic and Meteorological Laboratory, NOAA, Miami, partook in the initial application of video to CUFES and assisted us in strobe synchronization.

References

- Bolle RM, Connell JH, Haas N, Mohan R, Taubin G (1996) VeggieVision: A produce recognition system. Proc. Third IEEE Workshop on Applications of Computer Vision, pp. 244–51
- Checkley DM, Jr., Ortner PB, Settle LR, Cummings SR (1997) A continuous, underway fish egg sampler. *Fish Oceanogr* 6: 58–73
- Checkley DM, Jr., Dotson RC, Griffith DA (2000a) Continuous, underway sampling of eggs of Pacific sardine (*Sardinops sagax*) and northern anchovy (*Engraulis mordax*) in spring 1996 and 1997 off southern and central California. *Deep-Sea Res Part II-Topical Stud Oceanogr* 47: 1139–1155
- Checkley DM, Jr., Hunter JR, Motos L, van der Lingen CD (eds.) (2000b) Small Pelagic Fishes and Climate Change Program. Report of a workshop on the use of the Continuous, Underway Fish Egg Sampler (CUFES) for mapping spawning habitats of pelagic fish. *GLOBEC Rep* 14: 1–68
- Datacube (1994) SILL Programmer's Manual (1994) Datacube Inc., pp. 4–17
- Davis CS, Gallagher SM, Berman MS, Haury LR, Strickler JR (1997) The Video Plankton Recorder (VPR): design and initial results. *Arch Hydrobiol Beih Ergeb Limnol* 36: 67–81

- Heinemann PH, Pathare NP, Morrow CT (1996) An automated inspection station for machine-vision grading of potatoes. *Mach Vision Appl* 9: 14–19
- Hüller R, Päßgen W, Glossner E, Hummel P, Kachel V (1991) A PC-AT based video device for flow cytometrically triggered cell imaging in flow. *Comput Med Imaging Graphics* 15: 85–91
- Iwamoto S, Trivedi MM, Checkley DM (1998) Real-time detection and classification of objects in flowing water. Processing of 1998 Machine Vision Systems for Inspection and Metrology Conference. Proc SPIE 3521: 214–220
- Lasker RH (1985) An egg production method for estimating spawning biomass of pelagic fish: Application to the northern anchovy. NOAA Technical Report National Marine Fisheries Service 36
- Newman TS, Jain AK (1995) A survey of automated visual inspection. *Comput Vision Image Understanding* 61(2): 231–262
- Ortner PB, Hill LC, Edgerton HE (1981) *In-situ* silhouette photography of Gulf Stream zooplankton. *Deep-Sea Res* 28A: 1569–1576
- Schroeder HE (1984) Practical Illumination Concept and Technique for Machine Vision Application. Robots 8 Conference, pp. 14–43
- Sieracki CK, Sieracki ME, Yentsch CS (1998) An imaging-in-flow system for automated analysis of marine microplankton. *Mar Ecology-Progress Ser* 168: 285–296
- Smith PE (1973) The mortality and dispersal of sardine eggs and larvae. *Rapp PV Cons Int Explor Mer* 164: 282–292
- Tang X, Stewart WK, Vincent L, Huang H, Marra M, Gallagher SM, Davis CS (1998) Automatic Plankton Image Recognition. *Artif Intell Rev* 12: 177–199
- Tidd RA, Wilder J (1998) A Fish Detection and Classification System. Proc. of SPIE Conference on Machine Vision Systems for Inspection and Metrology VII, 141–148
- Trivedi MM (1990) Analysis of high-resolution aerial image, *Image Anal Appl* 281–305



Sadahiro Iwamoto earned his BS degree in Electrical Engineering from Cornell University in 1996 and his MS degree in Electrical and Computer Engineering from University of California, San Diego in 1999. His research interests include machine vision, image processing, computer graphics, and human computer interaction.



Dave Checkley Jr. is an Associate Professor of Biological Oceanography at the Scripps Institution of Oceanography (SIO) of the University of California, San Diego. He graduated from SIO, was a NATO Postdoctoral Fellow in Aberdeen, Scotland, and has served on oceanography faculties at the Universities of Alaska and Texas and North Carolina State University. His research concerns how the physical environment affects plankton and fish. He has studied zooplankton in Langmuir circulations from FLIP and fish spawning induced by winter storms off Cape Hatteras, and developed and uses CUFES to study spawning by anchovy and sardine worldwide.