Aquatic Debris Monitoring Using Smartphone-Based Robotic Sensors

Yu Wang¹, Rui Tan², Guoliang Xing¹, Jianxun Wang¹, Xiaobo Tan¹, Xiaoming Liu¹, and Xiangmao Chang¹,³

¹ Michigan State University, USA
² Advanced Digital Sciences Center, Illinois at Singapore
³ Nanjing University of Aeronautics and Astronautics, China
Aquatic Debris Monitoring

• **Various sources**: garbage, tsunami, earthquake, etc.

• **Timely detection and removal are critical**

• **Challenges**:
  – Dynamic aquatic environments
  – Sporadic arrivals of debris objects
Existing Approaches

- **Beachgoers/fishermen**
  - Small-scale monitoring, labor intensive, unreliable
- **Patrol boats**
  - Costly, short-term monitoring
- **Remote sensing** (balloon and satellite)
  - High cost, low monitoring resolution
- **AUVs/sea-gliders**
  - Expensive ($50k), bulky, heavy
Smartphone-based Aquatic Robot

- **Multi-modality sensing**
  - Camera and various inertial sensors
- **Rich computation & storage**
  - Advanced computer vision algorithms
- **Network capability**
  - Utilize cloud via cellular & WiFi interfaces
- **Mobility**
  - Adapt to sporadic debris arrivals
- **Low-cost**

Yu Wang @ https://sites.google.com/site/wangyu329/
Problem Statement

• **Reliably detect debris objects**
  – Sporadic over large aquatic regions
  – A wide variety of artifacts
  – Camera is the only viable sensor

![Background](image1)
![New frame](image2)
![Detection result](image3)

• **Enable long-term autonomous monitoring**
  – Smartphone processing: 6 J/frame
  – Aquatic movement: 3 W

• **Provide sufficient sensing coverage**
Outline

• Motivation

• Smartphone-based debris monitoring
  – Vision-based debris detection
  – Dynamic task offloading
  – Coverage-based rotation scheduling

• Performance evaluation

• Conclusion
Vision-based Debris Detection

- Align images to a common coordinate system
  - Mitigate the impact of camera shaking
  - Correspond distinguishable features (e.g., sharp corners)

Image Registration → Background Subtraction → Debris Identification

Reference image

Image to register

Registration result

Yu Wang @ https://sites.google.com/site/wangyu329/
Vision-based Debris Detection

- Align images to a common coordinate system
  - Mitigate the impact of camera shaking
  - Correspond distinguishable features (e.g., sharp corners)

- Challenge in aquatic environments
  - Few detectable image features
Horizon-based Image Registration

• Horizon line
• Hough transform
  – Line extraction
  – Compute-intensive, >2 sec/frame
**Horizon-based Image Registration**

- **Horizon line**
- **Hough transform**
  - Line extraction
  - Compute-intensive, >2 sec/frame
- **Align horizon lines**
  - Shifting
  - Rotation
Background Subtraction

- **HSV color space**
  - Robust to illumination changes

- **Gaussian mixture model (GMM)**
  - For each pixel
  - K Gaussians (3-dimensional)

- **GMM update**
  - new pixel
  - Existing Gaussians
  - no match, replace Gaussian with the lowest weight
  - update matched Gaussian(s)
Debris Identification

• Morphology opening operation
  – Remove salt-and-pepper noise

• Size filter
  – Remove distant objects
    (detected when approaching closer)

• Speed filter
  – Remove non-debris objects
    (move actively)
Compute-Intensive Processing

- Local processing
- Cloud processing (entire frame)
- Hybrid offloading (partial frame)

97% is consumed by the Hough transform
Hybrid Offloading

- Upload partial frame with the horizon line
- Offload the Hough transform only
- Execute the rest processing locally

\[ \sum a_z \]

accumulated linear vertical acceleration

\[ \sum a_z > 0 \]
shift downward

\[ \sum a_z < 0 \]
shift upward
Minimizing Energy Consumption

- **Local processing**
  \[ E_{\text{local}} = \text{power} \times \text{delay} \]

- **Cloud processing**
  \[ E_{\text{cloud}} = \text{power} \times \frac{\text{frame\_size}}{\text{link\_speed}} \]

- **Hybrid offloading**
  \[ E_{\text{hybrid}} = a \times E_{\text{cloud}} + b \times E_{\text{local}} \]
  \[ a = \frac{\text{partial\_size}}{\text{frame\_size}} \]
  \[ b = 1 - \frac{\text{delay\_Hough}}{\text{delay}} \]
Real-time Debris Detection Pipeline

Video frames → network condition

- low speed
- high speed

Debris Identification

- Size/speed filters
- Opening operation

Horizon-based Image Registration

GMM update

Yu Wang @ https://sites.google.com/site/wangyu329/
Coverage-based Rotation Scheduling

- Camera model
  - Field of view (FOV)

- Thickness
  - Effective coverage of debris arrivals

- Rotation objective:
  Miss coverage rate $\omega(\beta) < \xi$, $\forall \beta \in [0, 2\pi]$

- Schedule:
  $\{(\text{camera}_1\text{orientation}, \text{time}_1\text{interval}), \ldots \}$

Yu Wang @ https://sites.google.com/site/wangyu329/
Experiment Settings

• Prototype system
  – Samsung Galaxy Nexus
  – Gliding robotic fish
  – 1.99MB storage, 10.2MB RAM
  – Frame: 720 × 480 @ 0.25 fps

• Experiment environments
  – Water tank: 15 feet × 10 feet
  – Debris object: coke can
  – Horizon line: white foam intersects with water
  – Waves are generated by a feed pump
Detection Performance

$K = 3$ gives the best trade-off between detection probability and system overhead.
Detection Performance

Image registration mitigates the impact of camera shaking

\[
\bar{a}_z = 0.17 \text{ m/s}^2
\]

\[
\bar{a}_z = 0.09 \text{ m/s}^2
\]
Detection Performance

Image registration mitigates the impact of camera shaking

\[
\overline{a_z} = 0.17 \text{ m/s}^2 \quad \overline{a_z} = 0.09 \text{ m/s}^2
\]

average linear vertical acceleration characterizes the camera shaking level

Yu Wang @ https://sites.google.com/site/wangyu329/
Energy Consumption & System Lifetime

**Dynamic task offloading saves energy and prolongs lifetime**

- Reduce energy consumption up to 45%
- Prolong system lifetime up to 30%

**Graphs:**
- Energy consumption vs. WiFi link speed (Mbps)
- Lifetime vs. Duty cycle (%) for local and hybrid processing

Yu Wang @ https://sites.google.com/site/wangyu329/
Conclusion and Future Work

• Smartphone-based debris monitoring
  – Vision-based debris detection algorithms
  – Dynamic task offloading scheme
  – Coverage-based rotation scheduling algorithm

• Future work
  – Evaluation in an inland lake
  – Multi-node coordination

Yu Wang @ https://sites.google.com/site/wangyu329/