

---

## 17. Ciência, Tecnologia e Inovação

### **Detección de objetos en entornos acuáticos mediante la línea de Horizonte**

Autor: Britez, David; david.britez94@gmail

Co-autor(es): Paredes, Gustavo; gdpr.014@gmail.com

Orientador: Arzamendia, Mario; mario617@gmail.com

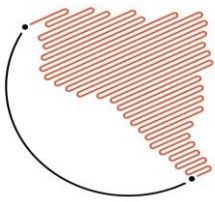
Laboratorio de Sistemas Distribuidos/ Facultad de Ingeniería

Universidad Nacional de Asunción

#### **Resumen**

Lipschutz(2013) propone el método H-EDHT (Edge detection and Hough transform based algorithm) que se utiliza para procesar imágenes obtenidas del lago Ypacarai a fin de detectar la línea de horizonte, y los objetos dentro del lago, para así dotar a un Vehículo de superficie de una herramienta para la evasión de obstáculos. Este método consiste en 5 pasos para la detección de línea de horizonte en imágenes marítimas, donde por lo general la línea más resaltante y de mayor longitud es la del horizonte, la complejidad de aplicar dicho método a imágenes dentro del lago radica en que, a diferencia del mar, en que hay otras líneas fuertes presente en la imagen como la de los cerros o árboles, por lo que una vez procesada la imagen por el método mencionado, se debe de hacer un análisis para discriminar la línea de horizonte de las demás, luego definir una región de interés donde buscar los obstáculos, mediante el algoritmo FindContours de OpenCV.

Palabras Claves: Horizonte, Detection, Hough.



## Introducción

Esta investigación es parte del proyecto “Vehículo Autónomo de Superficie (ASV) para el Estudio de Calidad del Agua en Lagos y Lagunas” perteneciente al Laboratorio de sistemas distribuidos de la Facultad de ingeniería de la Universidad Nacional de Asunción, Para lograr la autonomía del vehículo es necesario la detección de obstáculos, por ejemplo Larson(2006) utiliza un sistema de visión stereo, así como cámaras monoculares para la detección de líneas de horizonte y posterior reconocimiento de posibles obstáculos. La desventaja del sistema de visión stereo es que se requiere de una calibración precisa cada vez que se va a utilizar y cualquier perturbación afecta las mediciones del entorno.

Lipschutz (2013) realizó una investigación sobre varios algoritmos para la detección de línea de horizonte en imágenes IR en entornos marinos, resultando ser H-EDHT el más rápido.

Por lo mencionado anteriormente se desarrolló un algoritmo capaz de detectar objetos en la superficie del lago Ypakarai mediante el método H-EDHT.

## Objetivos

**Objetivo General:** Desarrollar e implementar un algoritmo capaz de detectar objetos en la superficie del lago Ypakarai

**Objetivos específicos:**

Detectar la línea de Horizonte

Definir una región de interés

Detectar objetos en la superficie del lago

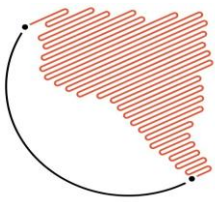
## Materiales y Métodos

Para el desarrollo del algoritmo se utilizó el lenguaje de programación Python 3.7.2, las librerías numpy y OpenCV.

Para la detección de la línea de horizonte se utilizó el método H-EDHT: Edge detección and Hough transform based algorithm o H-EDHT desarrollado por Lipschutz(2013) consiste en un pre-procesamiento de imágenes de 5 etapas a fin de poder diferenciar otros bordes de la línea de horizonte, las etapas son:

- Utilizar una erosión morfológica para reducir la probabilidad de detección de bordes débiles en las etapas posteriores
- Aplicar un filtro de medias o gaussiano para disminuir los detalles no relevantes en escala de grises.
- Aplicar la detección de bordes Canny.
- Aplicar la transformada de Hough.
- Seleccionar la línea de mayor longitud como la línea de Horizonte.

Una vez obtenida la línea de horizonte, definimos la región de interés como todos los pixeles por debajo de dicha línea, dentro de esta región de interés procedemos a realizar la detección de objetos en la superficie del lago mediante la función findContours de la librería OpenCV.



## Resultados y Discusiones

Fueron tomadas varias imágenes en el Lago Ypacarai para realizar las pruebas de detección de línea de Horizonte, una de ellas es la Figura 1



Figura 1: Imagen original

Aplicamos la erosión morfológica con un núcleo cuadrado a la Imagen original (Figura 1) y el filtro de medias, obteniéndose así la Figura 2.

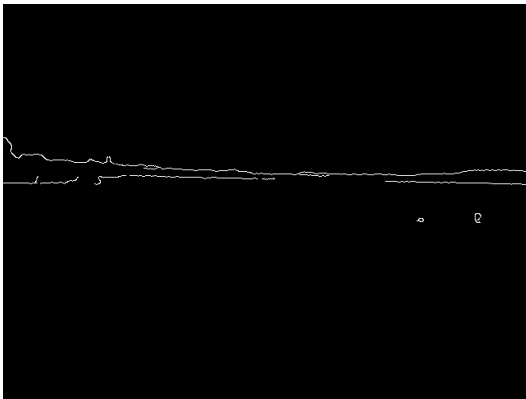


Figura 2: Imagen con filtro de medias, núcleo cuadrado

Al aplicar la transformada de Hough para la detección de líneas de la Figura 2 y dibujar dichas líneas sobre la imagen original se obtiene la Figura 3.



Figura 3: Imagen con detección de líneas, filtro de medias y núcleo cuadrado

Como se puede apreciar en la Figura 3 existen otras líneas resaltantes aparte de la línea de horizonte como ser el contorno de los cerros y árboles, además como las líneas de separación entre el agua y el suelo son fragmentos de la línea de horizonte, se procede a comparar la pendiente de cada línea con todas las restantes para hacer una reconstrucción de la misma.

Teniendo en cuenta que la ecuación de una recta puede escribirse como:

$$y = mx + b$$

donde  $m$  es la pendiente de la recta,  $x$  e  $y$  son las coordenadas de un punto cualquiera perteneciente a dicha recta y  $b$  es la ordenada en el origen, entonces podemos verificar si hay dos líneas con la misma pendiente y se procede a verificar si sus ordenadas en el origen coinciden, con lo que, si dos segmentos de recta cumplen con estas dos condiciones, entonces pertenecen a la misma recta, entonces se seleccionan los puntos opuestos de cada segmento podremos reconstruir todas las líneas rectas dentro de la imagen como se muestra en la Figura 4.

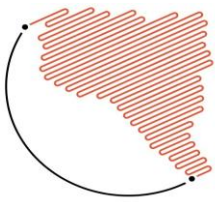


Figura 4: Imagen original con líneas rectas  
En esta imagen ya podemos seleccionar la línea de horizonte como la de mayor longitud.



Figura 5: Imagen original con línea de horizonte

Una vez obtenida la línea de Horizonte se puede determinar la región de interés como todos los píxeles debajo de dicha línea, obteniéndose así la Figura 6

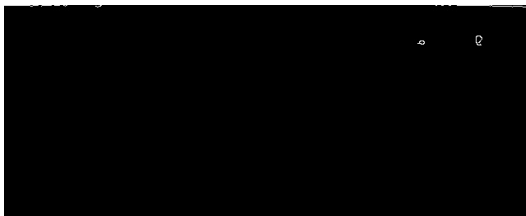


Figura 6: Región de interés, imagen binaria

Dentro de esta región se procede a utilizar la función Find contours de opencv, donde los contornos son todos los polígonos cerrados dentro de la imagen utilizada, una vez obtenidos todos los contornos se procede a dibujar dichos contornos en la imagen original obteniéndose así la Figura 7

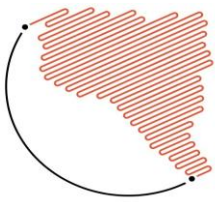


Figura 7: detección de objetos y línea de Horizonte  
Posteriormente se procedió a utilizar el algoritmo desarrollado en los 14 datasets utilizados por Paccaud(2018)



Figura 8: DS 1-14





Los errores y aciertos se ven de la siguiente forma:



Figura 9: Error en DS 1

Posteriormente el algoritmo fue adaptado para videos tomados en el lago Ypakarai



Figura 11: Imagen de video original



Figura 10: Acierto en DS 1

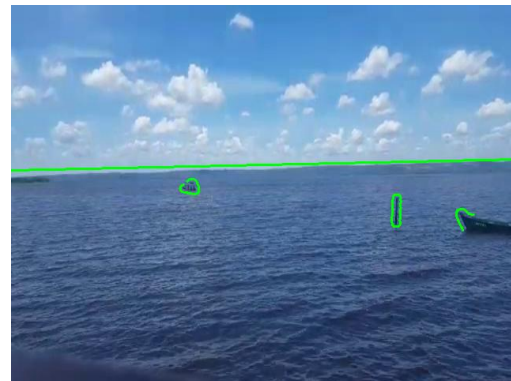
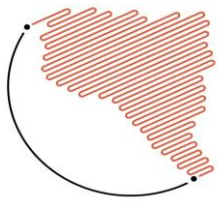


Figura 12: Detección de objetos en video

El algoritmo tuvo el siguiente porcentaje de acierto para cada uno de los datasets:

Dataset	Acierto(%)	Dataset	Acierto(%)
DS 1	97.85	DS 8	93
DS 2	100	DS 9	0
DS 3	0	DS 10	24.32
DS 4	98.09	DS 11	5.08
DS 5	29.75	DS 12	93.05
DS 6	99.63	DS 13	98.75
DS 7	100	DS 14	0

Tabla 1: Porcentaje de Aciertos



## Conclusiones

Dado que este método fue desarrollado para imágenes marítimas donde solo hay zona de agua y cielo es altamente preciso, pero para su extrapolación a imágenes de lagos se debe tener en cuenta la distancia a la orilla dado que existe una zona de agua, tierra y cielo, si se está lo suficientemente alejado de la orilla la zona de tierra en la imagen se reduce considerablemente, también como el caso de la Figura 11, por el contraste puede llegar a confundirse la zona de tierra con la del agua, que para fines prácticos es viable, también afecta la bruma y el reflejo del agua, esto se puede constatar en la Tabla 1 donde se tiene altos porcentajes de aciertos en imágenes donde la orilla se encuentra a mayor distancia que en otras, donde ocurren errores de acierto, esto abre la brecha a la utilización de dicho algoritmo para un vehículo de superficie en el lago Ypacarai con las limitaciones de no utilizar en días de bruma y mantener la distancia con la orilla para evitar errores.

## Referencia Bibliograficas

- Lipschutz, I., Gershikov, E., & Milgrom, B. (2013). New methods for horizon line detection in infrared and visible sea images. *Int. J. Comput. Eng. Res*, 3(3), 1197-1215.
- Larson, J., Bruch, M., & Ebken, J. (2006, May). Autonomous navigation and obstacle avoidance for unmanned surface vehicles. In *Unmanned Systems Technology VIII* (Vol. 6230, p. 623007). International Society for Optics and Photonics.
- Paccaud, P., & Barry, D. A. (2018). Obstacle detection for lake-deployed autonomous surface vehicles using RGB imagery. *PloS one*, 13(10), e0205319.