

# Detección de Características de Imagen y Flujo Óptico en Espacios Confinados Acuáticos

Rony Caballero, PHD<sup>1</sup> y Aranzazu Berbey, PHD<sup>1</sup>

<sup>1</sup>Universidad Tecnológica de Panamá, Panamá, [rony.caballero@utp.ac.pa](mailto:rony.caballero@utp.ac.pa), [aranzazu.berbey@utp.ac.pa](mailto:aranzazu.berbey@utp.ac.pa)

**Resumen**— En este trabajo de investigación se compara y evalúa los algoritmos de detección de esquinas basados en el algoritmo de Harris, así como el de Shi-Tomasi. El estudio se realiza en condiciones acuáticas con poca y mucha turbiedad. Posteriormente se presenta y evalúa un algoritmo para el seguimiento de esquinas en espacios confinados acuáticos basado en el flujo óptico. Los resultados obtenidos por el mismo han resultado muy prometedores.

**Palabras clave**—detección de esquinas, flujo óptico, procesamiento de imágenes submarinas.

## I. INTRODUCCIÓN

La aplicación de vehículos submarinos de inspección con sistemas de video se ha extendido a múltiples aplicaciones [1][2][3][4]. Sin embargo, la mayoría de éstas se ha limitado a su utilización a aguas abiertas. Por otro lado, resulta de interés la utilización de estos vehículos en ambientes donde el agua se presenta turbia y con poca iluminación, como es el caso dentro de alcantarillas y otros espacios confinados. En este sentido, algunos investigadores resaltan la poca información relativa a la comparación y evaluación de técnicas de detección de características de imagen en condiciones similares [5][6]. En consecuencia, también resulta de interés evaluar la factibilidad de aplicar estas técnicas para determinar del flujo óptico en este tipo de ambientes.

El primer objetivo de este trabajo es presentar los resultados preliminares de una investigación orientada a evaluar la eficacia y robustez de los algoritmos de detección de esquinas en ambientes confinados acuáticos. En este sentido, investigaciones preliminares [5][6] apuntan a una limitación de la efectividad de los mismos, producto de los fenómenos de dispersión y absorción de la luz, así como de la presencia de partículas en el agua. Así, en el primero de estos trabajos se apoya en el modelo matemático de dispersión (PSF) de Dolin, aplicable a aguas oceánicas a distintas profundidades. Aquí, se evidencia que los algoritmos para la detección de esquinas, en este caso basados en el algoritmo de Harris [7], presentan peor desempeño que los algoritmos de detección de segmentos (Blobs), en este caso basados en el Hessiano – Laplaciano [7]. En el segundo trabajo, se basa en una prueba experimental bastante exhaustiva que trata de analizar los efectos de turbiedad en el agua. Es importante destacar aquí, que los resultados experimentales también evidencian que la detección de segmentos es más efectiva que la detección de esquinas. Este resultado tiene sentido si se considera que los algoritmos de esquinas funcionan mejor en la medida que las imágenes presenten buena textura y bordes

claramente definidos, lo que usualmente no resulta fácil en condiciones acuáticas. En este último trabajo también se trata de validar otras técnicas de detección de características, aunque queda claro que las basadas en el Hessiano son mejores. Si bien es cierto el algoritmo de Harris es el más utilizado en evaluaciones comparativas, tiene sus detractores [8]. Esto se debe, a que si bien es cierto este algoritmo ha sido muy exitoso en la práctica, no presenta un formalismo lo suficientemente riguroso para algunas demostraciones teóricas a diferencia de otros como el de Shi-Tomasi [7]. En este sentido resulta de interés aplicar y comparar este algoritmo con el de Harris en ambientes confinados acuáticos.

El segundo objetivo de este trabajo, es evaluar la metodología necesaria para la determinación del flujo óptico en estas condiciones. En este aspecto hay muy pocos estudios aplicados a ambientes confinados acuáticos [9] y se hace necesario evaluar hasta qué punto la metodología convencional de flujo óptico es aplicable. Es importante resaltar que se busca estimar el flujo óptico en tiempo real, por lo, tanto resulta conveniente evitar la utilización de algoritmos de detección de segmentos (Blobs). Esto se debe, a que estos algoritmos pueden demandar mucha carga computacional. Además, no debe olvidarse que es necesario aplicar el filtrado de valores atípicos (outliers) sobre los resultados del detector de esquinas, para entonces poder aplicarlas al algoritmo de determinación de flujo óptico.

En esta investigación, se comparan primeramente los algoritmos de detección de esquinas de Harris y Shi-Tomasi en imágenes borrosas propias de las imágenes acuáticas. Luego, se analizará la forma más conveniente de filtrado de estos resultados, basados en un algoritmo RANSAC (Random Sampling and Consensus) para luego aplicar el algoritmo de Lucas-Kanade para la estimación del flujo óptico. Posteriormente, se analizarán los resultados experimentales. Finalmente, se presentan las conclusiones de este trabajo.

## II. DETECTORES DE CARACTERÍSTICAS

Existen dos tipos de características que son de gran interés en el análisis de imágenes. Estos son los detectores de esquinas y los detectores de segmentos (Blobs).

### A. Detectores de Esquinas

Los detectores de esquinas se basan en la matriz de autocorrelación, la cual describe las relaciones de gradientes en torno a un punto. Esta matriz está definida por:

$$M = w * \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \quad (1)$$

Donde,  $I_x$  es el gradiente de la intensidad de la imagen en el sentido de  $x$ ,  $I_y$  es el gradiente de la intensidad de la imagen en el sentido de  $y$ ,  $w$  es un kernel de pesos. Es decir, esta matriz es el resultado de una convolución entre el tensor formado por los productos de gradiente de la imagen y el kernel  $w$ .

Es conocido que si se determinan los valores propios de  $M$  en torno a un pixel, estos valores propios pueden determinar si este pertenece a una esquina, un borde o una sección plana. Así por ejemplo si un valor propio es mucho mayor que otro, se dice que estamos en un borde. Por otro lado si ambos son comparables y son lo suficientemente grandes se dice que pertenece a un borde. Las secciones planas se caracterizan por tener valores propios más bien pequeños.

En este sentido Harris y Stephens proponen el indicador,

$$R = \det(M) - \alpha \text{traza}(M) = \lambda_0 \lambda_1 - \alpha(\lambda_0^2 + \lambda_1^2) \quad (2)$$

donde,  $\lambda_0$  y  $\lambda_1$  son el mínimo y máximo valor propio de  $M$  y  $\alpha$  es una constante que típicamente toma valores entre 0.04 y 0.06. Por otro lado Shi y Tomasi proponen utilizar como indicador únicamente el valor propio mínimo,

$$R = \lambda_0 \quad (3)$$

### B. Pruebas experimentales

Para evaluar la capacidad de detección de esquinas se procede a evaluar el desempeño de ambos algoritmos sobre una foto de prueba. A la fotografía se le aplicaran los dos algoritmos de esquinas primero fuera del agua, luego en el agua con poca turbiedad y luego en el agua con mayor turbiedad.



Fig. 1 Foto fuera del agua. \* Algoritmo de Harris  
o Algoritmo de Shi-Tomasi.

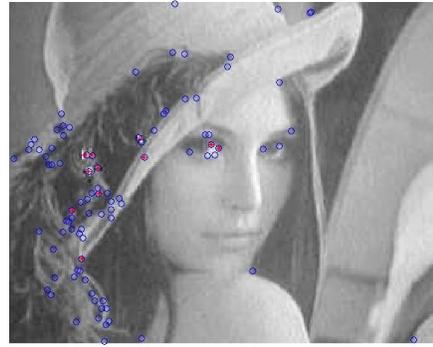


Fig. 2 Foto en el agua con poca turbiedad.  
\* Algoritmo de Harris o Algoritmo de Shi-Tomasi.

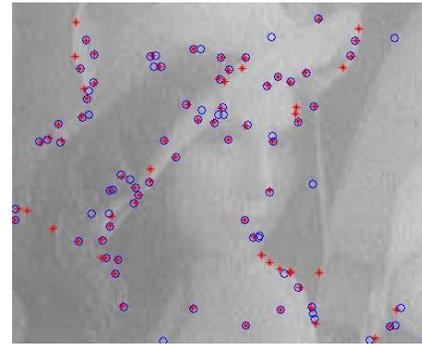


Fig. 3 Foto en el agua con mayor turbiedad.  
\* Algoritmo de Harris o Algoritmo de Shi-Tomasi.

Las imágenes en el agua se toman con una cámara que proporciona una resolución de 8 bits. Además, se aplica un filtro gaussiano 5x5 con  $\sigma=5/3$ .

Estos resultados preliminares parecen confirmar que los efectos de dispersión y absorción de luz en el agua afectan la efectividad de los algoritmos de detección de esquinas. Además, estos resultados parecen sugerir que algoritmo de Harris se ve mayormente limitado que el algoritmo de Shi-Tomasi en condiciones de poca turbiedad. No obstante en condiciones de mayor turbiedad ambos algoritmos se degradan, introducen errores en el posicionamiento de las esquinas y parecen confundir los bordes de la imagen con las esquinas.

Resulta de interés validar estos resultados con una imagen sintética de prueba, en este caso en el agua. Este segundo experimento parece confirmar nuevamente, que el algoritmo de Harris devuelve una mayor número de falsos positivos en condiciones de poca o ninguna turbiedad.

Para analizar más de cerca lo que sucede con las imágenes de las figuras 2 y 3 puede graficarse  $I_x$  vs  $I_y$ . Como puede apreciarse (véase la figura 5 y 6) los valores propios de  $M$  van decreciendo en la medida que la imagen se vuelve más borrosa. En consecuencia esto parece explicar que sea difícil diferenciar las esquinas de los bordes.

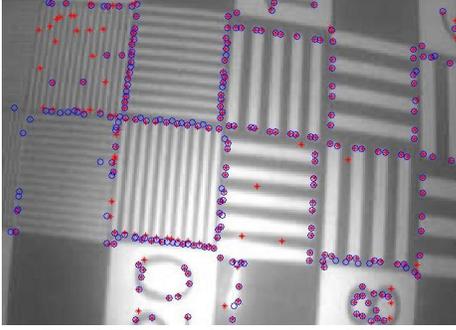


Fig. 4 Foto en el agua sin turbiedad.

\* Algoritmo de Harris o Algoritmo de Shi-Tomasi.

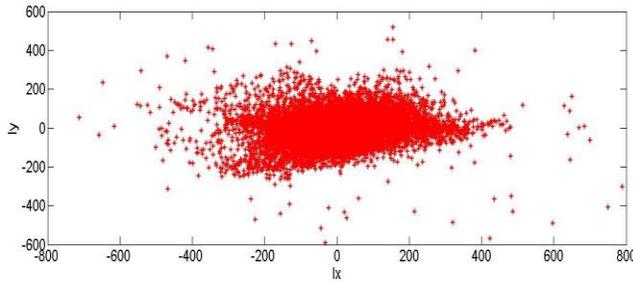


Fig. 5 Gradientes de la primera imagen en el agua con poca turbiedad  $0 < \lambda_0 < 1 \times 10^5$  y  $0 < \lambda_1 < 3.6 \times 10^5$

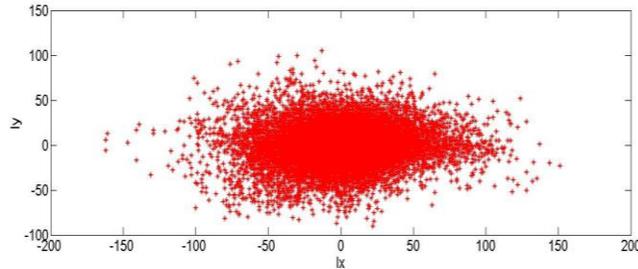


Fig. 6 Gradientes de la primera imagen en el agua con mayor turbiedad  $0 < \lambda_0 < 3.8 \times 10^3$  y  $0 < \lambda_1 < 1.2 \times 10^4$

Finalmente, después de estas pruebas se hace evidente la necesidad de un algoritmo que filtre la mayor cantidad de esquinas erróneas posibles.

### III. SEGUIMIENTO DE CARACTERÍSTICAS

#### A. Emparejamiento de Esquinas

Para determinar el flujo óptico del video resulta necesario identificar y emparejar características entre dos imágenes consecutivas. En otras palabras se debe identificar que esquinas se trasladan de una imagen a otra. No obstante, al aplicar los algoritmos de detección de esquinas a cada imagen resulta difícil establecer a priori la correspondencia de cada una en cada imagen consecutiva. Esto se complica más si el algoritmo de detección de esquinas proporciona esquinas falsas producto de las condiciones de la imagen. En

consecuencia, se debe aplicar algún tipo de filtrado a las esquinas para eliminar el mayor número posibles de esquinas falsas.

Una forma abordar este problema es aplicando el algoritmo RANSAC (Random Sample Consensus) [10][11]. Este algoritmo iterativo permite calcular parámetros de modelos matemáticos cuyos datos observados contienen valores atípicos. Este algoritmo es de tipo probabilístico, y permite obtener resultados razonables en la medida que se ejecuten mayor número de iteraciones. El algoritmo básico de RANSAC se detalla a continuación:

*Paso 1:* Establecer información de entrada :

*Esquinas:* el conjunto de esquinas sin filtrar

*Modelo:* el modelo para el cual se ajustan mejor las esquinas.

*n:* número mínimo de datos necesario para ajustar las esquinas.

*k:* el máximo número de iteraciones permitidas en el algoritmo.

*t:* un valor de umbral para determinar cuándo las esquinas ajustan al modelo.

*d:* el número de esquinas necesario para establecer que el algoritmo converge

*Paso 2:* Seleccionar *n* esquinas aleatoriamente

*Paso 3:* Ajustar los parámetros del *Modelo* a las *n* esquinas seleccionadas.

*Paso 4:* Si *d* esquinas ajustan dentro de *t* entonces ir al Paso 6.

*Paso 5:* Si el número de iteraciones no ha superado *k*, volver al Paso 2, sino informar que el algoritmo no converge.

*Paso 6:* Devolver los parámetros y las esquinas que ajustan al modelo.

La mayoría de los investigadores optan por el modelo:

$$\begin{bmatrix} x'_k \\ y'_k \end{bmatrix} = \begin{bmatrix} a & b \\ d & e \end{bmatrix} \begin{bmatrix} x_k \\ y_k \end{bmatrix} + \begin{bmatrix} c \\ f \end{bmatrix} \quad (4)$$

No obstante para ajustar este modelo se requiere un conjunto de seis esquinas. Tres provenientes de la imagen actual  $(x_1, y_1)$ ,  $(x_2, y_2)$ ,  $(x_3, y_3)$  y tres de la anterior  $(x'_1, y'_1)$ ,  $(x'_2, y'_2)$ ,  $(x'_3, y'_3)$ .

$$\begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} \quad (5)$$

$$\begin{bmatrix} y'_1 \\ y'_2 \\ y'_3 \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix} \begin{bmatrix} d \\ e \\ f \end{bmatrix} \quad (6)$$

En opinión de los autores de este artículo, la utilización de este modelo no resulta conveniente por las siguientes razones:

- Pruebas preliminares demuestran que ausencia de corrientes este modelo puede ser innecesariamente complejo. El agua es un medio fuertemente inercial, los cambios que se presentan entre dos imágenes consecutivas de video son muy pequeños.
- Las esquinas encontradas pueden presentar errores en el posicionamiento como evidencian los resultados anteriores, haciendo que se prefieran modelos lo más sencillos posibles, ya que son más robustos frente a los errores.

En consecuencia para el caso proponemos un modelo que solo involucra traslación de una imagen a la otra,

$$\begin{bmatrix} x'_k \\ y'_k \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \end{bmatrix} + \begin{bmatrix} c \\ f \end{bmatrix} \quad (7)$$

Este modelo es mucho más fácil de ajustar, requiere de solo una esquina en la imagen presente y otra en la anterior.

### B. Estimación del Flujo Óptico

Esta herramienta permite estimar el movimiento independiente de cada pixel entre dos imágenes consecutivas [7]. Este concepto está basado en el supuesto que cada pixel que se mueve entre dos imágenes consecutivas mantiene su intensidad aproximadamente constante.

$$I(x+u, y+v, t+1) = I(x, y, t) \quad (8)$$

Si se presentan pequeños desplazamientos esta expresión puede linealizarse aplicando series de Taylor, lo que lleva a:

$$\frac{\partial I(x, y, t)}{\partial x} u + \frac{\partial I(x, y, t)}{\partial y} v + \frac{\partial I(x, y, t)}{\partial t} = 0 \quad (9)$$

Donde  $u$  y  $v$  corresponde al flujo óptico y no es más que un campo de desplazamiento de pixeles. El problema con (9) es que existen múltiples soluciones para  $u$  y  $v$ .

En este sentido, Horn y Shunck proponen minimizar la integral,

$$\int_{\Omega} [\alpha^2 E_c^2 + E_b^2] d\Omega \quad (10a)$$

Donde,  $\alpha$  es un factor de regulación y

$$E_b^2 = \left( \frac{\partial I(x, y, t)}{\partial x} u + \frac{\partial I(x, y, t)}{\partial y} v + \frac{\partial I(x, y, t)}{\partial t} \right)^2 \quad (10b)$$

$$E_c^2 = \left( \frac{\partial u}{\partial x} \right)^2 + \left( \frac{\partial u}{\partial y} \right)^2 + \left( \frac{\partial v}{\partial x} \right)^2 + \left( \frac{\partial v}{\partial y} \right)^2$$

Este método funciona bien en presencia de flujo óptico denso. No obstante, no produce buenos resultados en condiciones de mucho ruido. Además, presenta el problema que demanda una alta carga computacional al tener que aplicar un método iterativo de Gauss-Seidel. En este sentido Lucas y Kanade proponen un método alternativo basado en los siguientes supuestos:

- Las dos imágenes consecutivas se encuentran separadas por un pequeño cambio de tiempo y los objetos mostrados en los mismos presentan pequeños desplazamientos.
- Los objetos mostrados en las imágenes presentan diferentes niveles de gris, que cambian suavemente.

Bajo estos supuestos Lucas y Kanade pueden asumir que un pixel de intensidad  $I(x,y)$  comparte la misma intensidad entre los pixeles vecinos. Bajo esta premisa a (9) ahora se le añaden  $n$  ecuaciones correspondientes a sus  $n$  pixeles vecinos.

$$\begin{aligned} I_{x1}u + I_{y1}v &= -I_{t1} \\ &\vdots \\ I_{xn+1}u + I_{yn+1}v &= -I_{tn+1} \end{aligned} \quad (11)$$

Donde,  $I_x = \frac{\partial I(x, y, t)}{\partial x}, I_y = \frac{\partial I(x, y, t)}{\partial y}, I_t = \frac{\partial I(x, y, t)}{\partial t}$

El flujo óptico en (11) puede calcularse fácilmente a partir de la pseudoinversa del sistema, lo que lo vuelve computacionalmente eficiente. Este algoritmo en particular es el preferido por la mayoría de los investigadores hoy en día y es el que seleccionaremos para este trabajo.

### C. Algoritmo Propuesto para el Seguimiento de Características

En este punto los autores proponen el siguiente algoritmo para el seguimiento de Características (Esquinas).

*Paso1:* Aplicar el algoritmo de detección de esquinas a las dos imágenes consecutivas del video.

*Paso2:* Aplicar el algoritmo de RANSAC a las dos imágenes de interés utilizando el modelo descrito por (7).

*Paso3:* Aplicar el algoritmo para la estimación del flujo óptico de Lucas-Kanade, que está basado en la solución del sistema (11).

#### IV. RESULTADOS EXPERIMENTALES

##### A. Equipo Experimental

Para evaluar el algoritmo de seguimiento de características se cuenta con una cámara a color submarina Inspecam a color (véase figura 7). Además, se realizan las pruebas en una tina del laboratorio de turbomaquinaria de la Universidad Tecnológica de Panamá.



Fig. 7 Cámara de video submarino a color

##### B. Pruebas Experimentales

Para evaluar el desempeño del algoritmo se procede a tomar un video en la tina, que cuenta con tuberías y otros elementos adicionales. Una vez obtenido el video se seccionan imágenes consecutivas con una calidad de 8 bits. Luego, de aplicar los algoritmos de detección de esquinas y RANSAC, se selecciona una vecindad de 10 píxeles a la izquierda y a la derecha, así como 10 píxeles hacia arriba y hacia abajo de cada píxel de esquina. Esto último define el ajuste del algoritmo de Lucas-Kanade. En la figura 8 y 9 se muestran las imágenes de prueba del algoritmo.

Primeramente, se probará el algoritmo, utilizando un detector de esquinas basado en Harris, pero con filtrado de las esquinas (véase figura 10). Luego, se prueba con el detector de Shi-Tomasi (véase figura 11), también con filtrando de esquinas.



Fig. 8 Primera imagen de la primera prueba



Fig. 9 Segunda imagen de la primera prueba



Fig. 10 Flujo óptico con detección de esquinas de Harris



Fig. 11 Flujo óptico con detección de esquinas de Shi-Tomasi

Después de esta prueba, podría concluirse que ambos devuelven un flujo óptico muy parecido. Esto puede

explicarse gracias al filtrado que se da en el algoritmo RANSAC, ya que elimino la mayoría de las falsas esquinas que devuelve el algoritmo de Harris.



Fig. 12 Primera imagen de la segunda prueba



Fig. 13 Segunda imagen de la segunda prueba

A continuación se realiza una segunda prueba en un área dentro de la tina con mejores esquinas (véase figura 12 y 13). En este caso la imagen presenta mayor gama de valores de gris, que en el caso anterior, por lo tanto es de suponerse que los resultados deben ser mejores. Además, al igual que en el caso anterior los resultados entre el flujo óptico obtenido con Harris y con Shi-Tomasi son muy similares.



Fig. 14 Flujo óptico con detección de esquinas de Harris



Fig. 15 Flujo óptico con detección de esquinas de Shi-Tomasi

## V. CONCLUSIONES

En este trabajo, queda evidenciado que trabajar en entornos acuáticos demanda ajustar en cierta medida la metodología convencional utilizada para procesar imágenes. Los resultados preliminares apuntan también, a que si bien es cierto el algoritmo de Shi-Tomasi se comportaba mejor en estos ambientes acuáticos, el algoritmo de Harris parece que puede utilizarse siempre y cuando se filtren las esquinas detectadas por el mismo.

Otro punto importante a destacar es que en la medida que la turbiedad aumenta, los valores propios de la matriz  $M$  que se utiliza en los detectores de esquinas decrecen de forma importante hasta hacer muy difícil diferenciar las esquinas de los bordes.

Si bien es cierto, el algoritmo de RANSAC es muy importante para el correcto apareamiento de esquinas, otras pruebas experimentales han mostrado su alto costo computacional. Además, sería conveniente modificarlo para que también pueda eliminar falsas esquinas de las imágenes.

Finalmente, el propuesto para el seguimiento de las características ha funcionado relativamente bien. Se evidencia en los resultados, algunas imprecisiones en la estimación del flujo óptico. Este problema se debe principalmente a la presencia de falsas esquinas que no pudieron ser eliminadas en algoritmo de RANSAC. En consecuencia, resulta de interés mejorar el algoritmo en este sentido.

#### AGRADECIMIENTOS

Los autores desean agradecer a la Secretaría Nacional de Ciencia y Tecnología de Panamá (SENACYT) y a la Universidad Tecnológica de Panamá por el apoyo recibido para esta investigación.

#### REFERENCES

- [1] J Batlle, P Ridao, R Garcia, M Carreras, X Cufi, A El-Fakdi, D Ribas, T Nicosevici, E Batlle, G Oliver, A Ortiz, J Antich. “*URIS: Underwater Robotic Intelligent System*”. Automation for the Maritime Industries 177-203
- [2] Choi, H.T. Hanai, A. ; Choi, S.K. ; Yuh, J. “*Development of an underwater robot, ODIN-III*”. International Conference on Intelligent Robots and Systems, 2003. (IROS 2003).
- [3] S. Krupinski, G. Allibert, M. Hua, T. Hamel. “*Pipeline tracking for fully-actuated autonomous underwater vehicle using visual servo control*”. American Control Conference, 2012.
- [4] R. Caballero, A Vega, A Berbey, J Rodríguez. “*Vehículo submarino de 6 grados de libertad para inspección interna de tuberías*”. Tenth LACCEI. Panamá City, Panamá, 2012.
- [5] K. Oliver, W. Hou, and S. Wang, “*Image feature detection and matching in underwater Conditions*”. Proceedings of SPIE - The International Society for Optical Engineering. April 2010
- [6] R. Garcia, N. Gracias. “*Detection of Interest Points in Turbid Underwater Images*”. OCEANS 2011 IEEE.
- [7] R. Szeliski, “*Computer Vision: Algorithms and Applications*”. 2010 Springer.
- [8] C.S. Kenney ; M. Zuliani ; B.S. Manjunath. “*An Axiomatic Approach to Corner Detection*”. Computer Vision and Pattern Recognition, 2005. CVPR 2005 CVPR 2005. IEEE Computer Society Conference.
- [9] C. Cheng, J. Chen. “*Application of Optical Flow to Station keeping of Underwater Vehicles* Conference: Proceedings of the Third ICSC Symposia on Intelligent Industrial Automation (IIA'99) and Soft Computing (SOCO'99), June 1-4, 1999, Genova, Italy.
- [10] L. Dung, C. Huang, Y. Wu. “*Implementation of RANSAC Algorithm for Feature-Based Image Registration*”. Journal of Computer and Communications, 2013.1,46-50.
- [11] I. Austvoll, B. Kwolek “*Region Covariance Matrix-Based Object Tracking with Occlusions Handling*” International Conference, ICCVG 2010, Warsaw, Poland, 2010.
- [12] B. Horn, B. Shunck. “*Determining Optical Flow*”- Artificial Inteligence, Volume 17, Issues 1–3, August 1981.
- [13] B. Lucas, T. Kanade. “*An Iterative Image Registration Technique with an Application to Stereo Vision*”. Proceedings of Imaging Understanding Workshop. 1981