

ALARMA DETECTORA DE ESCORPIONES UTILIZANDO PROCESAMIENTO DIGITAL DE IMÁGENES

Francisco Luis **Giambelluca**^{1,3}, Jorge Osio¹, Marcelo Cappelletti², José A. Rapallini¹, Luis A. Giambelluca³

¹ UIDET de Técnicas Analógico-Digitales [CeTAD] (Centro Asociado CIC - Facultad de Ingeniería – UNLP).- 116 y 48 - 2do piso, La Plata, Prov. de Bs. As. ² Grupo de Estudio de Materiales y Dispositivos Electrónicos (GEMyDE, Facultad de Ingeniería, UNLP). Calle 116 y 48, 1º piso, La Plata, Prov. de Bs. As. ³ CEPAVE, (CONICET - UNLP- CCT La Plata). Boulevard 120 e/61 y 64, La Plata, Prov. de Bs. As. - frangiambe@gmail.com

INTRODUCCIÓN

Teoría biológica

Durante muchos años los escorpiones han sido objeto de estudio, debido a su alta peligrosidad [1]. Todos los escorpiones emiten una fluorescencia de color cian (con una longitud de onda entre los 440 y los 490 nanómetros) cuando se iluminan con luz ultravioleta (UV) [2]. Este fenómeno fue descubierto casi simultáneamente en 1954 por el zoólogo italiano M. Pavan y el zoólogo sudafricano R. F. Lawrence, y revolucionó el estudio de la biología y ecología de los escorpiones gracias a que fue posible localizarlos y observarlos por la noche usando lámparas de luz negra [3-5].

La intensidad de la fluorescencia aumenta con la edad del escorpión y la dureza de su cutícula y es más brillante en las zonas más duras. Una vez adquirida, la fluorescencia persiste incluso después de la muerte del escorpión ([6] y [7]).

Solución electrónica

Con el fin de poder detectar los escorpiones, se procedió a realizar un sistema de alarma utilizando el procesamiento digital de imágenes. Para realizar un programa de menor nivel de cómputo y que fuese a su vez eficiente, se utilizó la propiedad de fluorescencia antes mencionada.

Para desarrollar esta alarma se procedió a realizar el examen pixel a pixel de una imagen capturada desde una Webcam en busca del color característico de la fluorescencia. Para realizar el procesamiento de imágenes se utilizaron las librerías de código abierto Open CV en lenguaje de programación C [8-10], mediante el entorno de desarrollo Code::Blocks también de código abierto.

Los elementos utilizados para los ensayos son una computadora, una Webcam, una fuente de luz UV y un espacio oscuro donde poder visualizar la fluorescencia de los escorpiones. Dicho espacio va desde una caja cerrada para los primeros ensayos hasta una habitación oscura.



Fig. 1 Escorpión *Tityus trivittatus* con luz natural y UV con diferentes fondos.

En este trabajo se realizaron pruebas con ejemplares de *Tityus trivittatus* y con individuos de los órdenes Isopoda, Coleóptera y Blattodea como controles. En Fig. 1 se muestra un

mismo ejemplar de escorpión *Tityus trivittatus* iluminados con luz natural y con una fuente de luz UV con dos fondos diferentes para apreciar mejor la imagen.

PARTE EXPERIMENTAL

Para la detección de los escorpiones se utilizó el formato de imagen HSV, el cual discrimina el color, con un barrido a través de todos los píxeles de la imagen con el fin de encontrar el color deseado. Esto se consiguió mediante un doble ciclo *for()* para recorrer la imagen a lo alto y ancho píxel a píxel, y colocando una sentencia *if()* que compara cada píxel al valor deseado, se logró la detección cambiando una variable en caso favorable [11].

Para saber cual es el color que se busca se tuvo que crear un programa que tome una captura de la Webcam y muestre el color del píxel que se selecciona con el mouse. Utilizando este software se procedió a realizar un muestreo de múltiples puntos del escorpión, a fin de obtener suficientes muestras para obtener un intervalo de valores para el color a detectar. En Fig. 2 se presenta el registro de las muestras obtenidas del color del escorpión.



Fig. 2 Gráfico de valores obtenidos del color del escorpión.

De los datos medidos se obtuvo un valor medio de 58,88 y un desvío estándar de 2,75, por lo que los valores del color deben ir desde 56 hasta 62. Una vez obtenido estos resultados se procedió a realizar ensayos con el fin de verificar la detección.

Un inconveniente encontrado con esta metodología es la presencia de ruido en la imagen, por lo que la detección se daba en cualquier situación, aún en plena oscuridad. Para solucionar este inconveniente se debió reducir el ruido de las capturas obtenidas, se procedió entonces a realizar un promedio entre dos capturas seguidas, con el fin de que el ruido disminuya. Dicho procedimiento funcionó de manera eficiente dado que ahora la cámara puede permanecer en oscuridad total sin tener una detección errónea.

Con el fin de optimizar el funcionamiento del programa, se introdujo en el mismo un sistema de detección de movimiento por procesamiento digital de imágenes. Esta detección se obtuvo haciendo la diferencia entre dos imágenes y fijándonos que tal diferencia no supere un umbral mínimo, de forma tal que si se superase el umbral habría movimiento [12]. La diferencia entre imágenes se realizó en blanco y negro con el propósito de no utilizar mucho procesamiento. Las imágenes diferenciadas son sin ruido, es decir que para realizar este procedimiento se requirieron cuatro capturas (dos por cada imagen sin ruido).

La optimización del programa viene dada en que la idea es solo procesar en formato HSV aquellos píxeles que hayan tenido movimiento. Para ello es necesario conocer el área de píxeles que se ha movido, lo cual se logra creando una imagen blanco y negro en la cual los píxeles blancos son quienes han superado el umbral mínimo requerido para detectar movimiento, y los píxeles negros son los que no se han movido. Utilizando la función *cvFindContours()* encontramos los contornos de estas áreas de movimiento, luego establecemos un área mínima para no detectar puntos solitarios y utilizamos la función *cvBoundingRect()* para establecer un rectángulo mínimo que encierre esa área de movimiento. A continuación, utilizamos la función *cvRectangleR()* para dibujar este rectángulo en pantalla, se eligió el color azul para marcar el rectángulo de movimiento.

Encerrar los píxeles en un rectángulo permite tener los límites del mismo y poder analizar en ese rectángulo solo la presencia del color deseado. Si dentro del rectángulo de movimiento hay píxeles del color buscado se da la detección y se marca este rectángulo de color rojo. Este último puede superponer un rectángulo azul preexistente en caso de tener un píxel del color buscado dentro del movimiento, aparentando una detección sin previa detección de movimiento, pese a que esta se haya dado. La Fig. 3 muestra una imagen de ejemplo con un escorpión de fondo, donde se pueden apreciar los diferentes rectángulos de colores azul y rojo, además de un ejemplo de un rectángulo rojo no contenido dentro de otro azul. La Fig. 4 ilustra el diagrama de flujo del funcionamiento del programa para la detección de escorpiones.



Fig. 3 Imagen con escorpión de fondo.

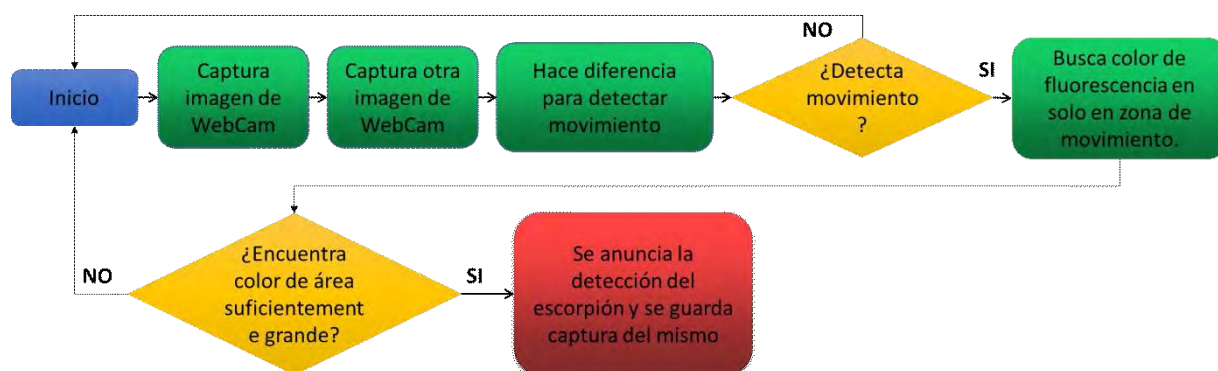


Fig. 4 Diagrama de flujo básico de funcionamiento.

Ante la presencia de falsas detecciones con la metodología anterior, debido principalmente a la presencia del color buscado en píxeles aislados dentro del movimiento, producto del ruido de la imagen y la inestabilidad del algoritmo, se procedió a realizar un procesamiento de las imágenes similar a la detección de movimiento, pero esta vez, para la detección del color de la fluorescencia. Para ello, se creó una nueva imagen en la cual los píxeles blancos corresponden al color de la fluorescencia buscada y los negros a los restantes colores, tal y como se observa en la Fig. 5.

Utilizando las mismas funciones que para la detección de movimiento se encontraron las áreas correspondientes a este color dentro de la imagen blanco y negro de la fluorescencia, generada previamente. Del resultado de dicha búsqueda de contornos, se filtraron las áreas más pequeñas para no detectar puntos solitarios (reducción de ruido), al igual que con la detección de movimiento y se procedió a realizar la detección con esta nueva metodología.



Fig. 5 Imagen de escorpión solo con color de fluorescencia.

RESULTADOS

Una vez finalizado el algoritmo se procedió a realizar ensayos en presencia de otros especímenes que puedan estar normalmente en el mismo ambiente que un escorpión. El primer ensayo de este tipo se realizó con un bicho bolita (*Porcellio laevis*), dicho ensayo demostró un inconveniente que se ilustra a continuación en la Fig. 6 con una imagen del color buscado como el caso anterior.



Fig. 6 Imagen de un escorpión (izquierda) y de un bicho bolita (derecha).

Como se observa en la Fig. 6, el escorpión presenta un color más uniforme mientras que el bicho bolita presenta el color en su contorno, por lo que esta presencia del color se da por el reflejo de su superficie. Este reflejo es el de la fuente de luz UV que en este caso es el de un tubo fluorescente que introduce mucho ruido al sistema.

Para eliminar este problema se evaluaron varias alternativas, entre ellas la utilización de un filtro polarizado para poner delante de la Webcam, dicho filtro se utiliza en fotografía para reducir los reflejos de las superficies; otra opción era utilizar procesamiento de imágenes con el fin de localizar estos reflejos y eliminarlos, pero dicha tarea requeriría una alta demanda de procesamiento. Finalmente, se optó por utilizar un método combinado de erosión y dilatación. Dicha elección se realizó porque al ser la fluorescencia del escorpión de color muy uniforme, realizar una dilatación uniría las áreas internas, y realizar una posterior erosión no sería tan nefasto como lo sería a una línea de borde de un reflejo. Posteriormente se haría una dilatación para evitar que queden áreas muy pequeñas que sean descartadas como una detección. Para llegar al nivel óptimo de iteraciones de dilataciones y erosiones se procedió a realizar ensayos con la imagen previamente mostrada. Los resultados de los ensayos son mostrados de la Fig. 7 hasta la Fig. 12, incrementando paulatinamente la cantidad de dilataciones y erosiones. En la Fig. 7 una erosión simple demuestra la degradación del escorpión que en casos de menor fluorescencia podría desaparecer perdiéndose la posibilidad de detección. En la Fig. 8 con la primera dilatación, aunque se resalta el brillo del bicho bolita, consolida la imagen del escorpión haciendo que se unan sus áreas internas, lo que ante futuras erosiones dificultará su desaparición. Sin embargo, el bicho bolita ante futuras erosiones se degradará considerablemente dado que su color sigue siendo líneas de contorno sin cuerpo. En la Fig. 9 luego de dos dilataciones y 4 erosiones pese a que el bicho bolita se ha degradado de forma considerable aún quedan rastros de él, por lo que ante una futura dilatación se volverá más grande ocasionando un posible falso positivo.



Fig. 7 Una erosión simple



Fig. 8 Una dilatación inicial



Fig. 9 Imagen resultado de 2 dilataciones y 4 erosiones.

Para solucionar la imagen anterior se realizaron dos erosiones más obteniendo el resultado deseado de eliminar al bicho bolita. Para compensar la degradación del escorpión se procederá a realizar dilataciones que le devuelvan algo de su forma y tamaño original.



Fig. 10 Imagen resultado de 2 dilataciones y 6 erosiones.



Fig. 11 Imagen resultado de 2 dilataciones, 6 erosiones y 4 dilataciones.

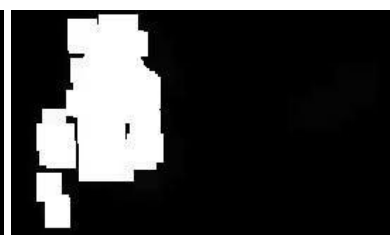


Fig. 12 Imagen resultado de 2 dilataciones, 6 erosiones y 8 dilataciones.

Aunque la Fig. 11 tiene un buen resultado su tamaño y densidad pueden mejorar, obteniéndose así la Fig 12. Esta se obtuvo luego de realizar 2 dilataciones, 6 erosiones y 8 dilataciones, muestra el resultado aceptable que se buscaba, en la cual no se observa al bicho bolita, y el tamaño y densidad del escorpión se pueden apreciar de manera adecuada, por lo que utilizaremos este método para procesar nuestra imagen del color deseado antes de encontrar los contornos y áreas para realizar la detección pertinente. Las funciones de OpenCV utilizadas para estos procesos son *cvErode()* y *cvDilate()*.

DISCUSIÓN

Una vez re-calibrado el programa se procedió a realizar una prueba con un escorpión vivo, el cual, pese a ser detectado ante su fluorescencia demostró falencias en el sistema de detección de movimiento, dado que se desplazó delante de la cámara a una velocidad considerable, suficiente para no detectar su movimiento. Para solucionar este problema se procedió a revisar el código.

Una vez realizada esta revisión, se detectaron diferentes inconvenientes que pueden ocasionar la lentitud del programa y por lo tanto la incapacidad de detectar un movimiento muy brusco.

Ante ello, lo primero que se quitó fue la reducción de ruido con el promedio de dos capturas que, aunque resulta eficiente utilizando la detección con las funciones de áreas y contornos, es totalmente innecesario dado que no se percibe diferencia entre la presencia o no de este sistema. Quitar este sistema de reducción de ruido disminuyó considerablemente el tiempo entre capturas, mejorando la detección de movimiento.

Se verificó, además, que los procesos de erosión y dilatación no ocasionan lentitud en el funcionamiento de la alarma quitándolos, sin obtener diferencia apreciable del desempeño, por lo que no debieron ser reemplazados por otro mecanismo para quitar los reflejos previamente mencionados. Otro componente del programa que ocasiona lentitud de procesamiento son las muestras en pantalla de diferentes imágenes del procesamiento digital de imágenes, las cuales se utilizan como testigos para verificar el funcionamiento del programa. Estos componentes serán reducidos considerablemente cuando el programa funcione de la manera esperada y no requiera correcciones adicionales.

Solucionado esto se procedió a realizar una nueva prueba con escorpión vivo el cual demostró movimientos de la misma velocidad que en el ensayo anterior, y esta vez fue detectado por el programa en su velocidad máxima.

CONCLUSIONES

En base a la propiedad de fluorescencia de los escorpiones, se desarrolló un sistema electrónico capaz de detectarlos, utilizando procesamiento digital de imágenes. Para ello se utilizó el formato de imagen HSV, el cual discrimina el color, con un barrido a través de todos los píxeles de la imagen con el fin de encontrar el color deseado. Además, se utilizó un método combinado de erosión y dilatación para mejorar la imagen detectada.

El sistema de alarma puede funcionar con cualquier orientación espacial debido a que la detección se da por la búsqueda del color particular de la fluorescencia de los escorpiones y no por detección de forma. La distancia máxima a la que podrá funcionar dependerá principalmente de la resolución de la cámara utilizada (mayor resolución es mayor distancia) y de la capacidad de la unidad de procesamiento de manejar la resolución de las imágenes tomadas con la misma.

Se debe destacar que el objetivo final de este sistema es la detección automática de escorpiones en instituciones educativas, por lo que será necesario realizar la implementación final sobre un dispositivo portátil como una Raspberry pi.

BIBLIOGRAFÍA

- [1] F. T. Abushama, "On the behavior and sensory physiology of the scorpion *Leiurus quinquestriatus*", *Animal Behaviour*, vol. 12, pp. 140-153, 1964.
- [2] G. R. Blass and D. D. Gaffin, "Light wavelength biases of scorpions", *Animal Behaviour*, vol. 76, pp. 365-373, 2008.
- [3] P. H. Brownell, *Sensory ecology and orientational behaviors*. In: *Scorpion Biology and Research* (Ed. by P. H. Brownell & G. A. Polis), pp. 159-183. Oxford: Oxford University Press, 2001.
- [4] E. A. Camp and D. D. Gaffin, "Escape Behavior mediated by negative phototaxis in the scorpion *Paruroctonus utahensis* (Scorpiones, Vaejovidae)", *Journal of Arachnology*, vol. 27, pp. 679-684, 1999.
- [5] A. Fasel, P. Muller, P. Suppan and E. Vauthey, "Photoluminescence of the African scorpion *Pandinus imperator*", *Journal of Photochemistry and Photobiology B: Biology*, vol. 39, pp. 96-98, 1997.
- [6] L. M. Frost, D. R. Butler, B. O'Dell and V. Fet, A coumarin as a fluorescent compound in scorpion cuticle. In: *Scorpions 2001; In Memoriam Gary A Polis* (Ed. by V. Fet & P. A. Selden), pp. 363-368. Burnham Beeches, Bucks: British Arachnological Society, 2001. [7] L. D. Honetschlager, "A new method for hunting scorpions", *Turtox News*, vol. 43, pp. 69, 1965.
- [8] D. Quing Chen, "A Basic Introduction to OpenCv for Image Processing", *University of Ottawa*, pp. 12, 2007.
- [9] R. Igual y C. Medrano, *Tutorial de Open CV, Laboratorio de visión por computadora*, 2008.
- [10] G. Bradski y A. Kaebler, *Learning Open CV*, Orreilly, ISBN: 978-0-596-51613-0, 2008. [11] A. Maldonado, "Diseño de módulo de Pre procesamiento y Extracción de características físicas de una imagen", *Escuela Politécnica Nacional*, Cap. 3 pp. 54-55, 2006.
- [12] C. F. Calderón, "Visión Artificial–Umbralización", *Universidad Pontificia Javeriana*, pp. 1-11, 2011.