

Rastreo basado en vision por computador de mariposas en un ambiente controlado

Oscar Josué Ulate Alpízar, TEC

10 de junio de 2019

Resumen

Este artículo muestra el diseño y desarrollo de un disparador simultáneo por hardware para un conjunto de tres cámaras. Este se realiza utilizando el GPIO de las cámaras FLIR. A continuación se explica el desarrollo de esta implementación, sus puntos completados y sus limitaciones.

Introducción

El avance científico y la ingeniería se encuentra cada vez más unido. Se requiere de la interacción de ambas para realizar avances significativos en el terreno práctico de la ciencia. Es esta interacción la que requiere el CICANUM para monitorear el vuelo de las mariposas monarca.

Cada año las mariposas monarca en Canadá vuelan a México para su migración en la primavera. Vuelven a Canadá en el otoño. Esta es una de las migraciones más grandes del reino animal.

Se sabe que estas siguen los campos magnéticos terrestres para guiarse en sus vuelos migratorios, pero no se ha logrado responder el cómo. Es dentro de este contexto que se encuentra este proyecto. Se requiere sincronizar la captura de imágenes para conocer el punto exacto donde se encuentra las mariposas en un ambiente controlado.

Las cámaras disponibles son las FLIR Blackfly®S, de 1,3 MP de resolución, con solo un canal de color (escala de gris). Estas poseen un conector USB por el cuál se conectan a la computadora principal y una entrada de GPIO, que permite entre otras funciones,

controlar el disparador por hardware.

Antecedentes

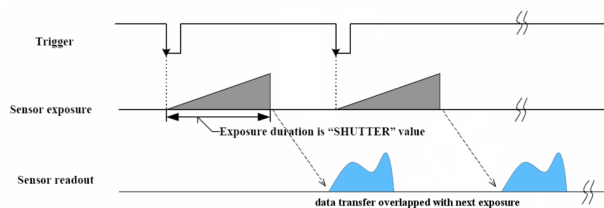


Figura 1: Diagrama de disparador sincrónico con overlapping [1].

En la Figura 1 se muestra el funcionamiento del disparador con overlapping explicado por FLIR, los fabricantes de las cámaras utilizadas. Como se muestra en el diagrama, en el *negedge* del disparador comienza la captura de la imagen. En una captura continua la cámara es capaz, de manera simétrica, de enviar los datos mientras la cámara trabaja en la exposición de la siguiente imagen. Es este el tipo de disparador usado para este proyecto.

Con respecto a el estudio de las mariposas, existen varios estudios interesantes donde se ha intentado buscar el funcionamiento de sus antenas [2]. En este artículo, los autores determinaron que las antenas de las mariposas funcionan con un reloj interno cada uno, y es el cerebro del animal, que sincroniza ambos para dirigir el vuelo.

En [3], Noda Akihito (y coautores), muestran el uso de cámaras con capturas sincronizadas de alta veloci-

dad. Con velocidades de captura a 500 capturas por segundo, sincronizan las cámaras por medio de conexión Ethernet para, por medio del protocolo PTP (Precision Time Protocol) sincronizar los relojes de las cámaras.

El acercamiento de Wand, Miao y Ling para sincronizar las cámaras es completamente distinto. En su artículo [4], sincronizan capturas de cámaras de uso convencional. Lo hacen buscando puntos característicos en las tomas para, con esta información, crear una "matriz fundamentalz de esta manera poder sincronizar los tiempos de los vídeos.

Un área de la visión por computadora que ha avanzado bastante y ha propiciado el avance de este tipo de tecnologías es los deportes. Muchas estadísticas se pueden obtener de cámaras que rastrean jugadores en un campo de juego. Kumar (et al.) en [5] proponen un arreglo de dos o más cámaras para seguir el recorrido de una pelota de tenis, además de la información de la calibración de las cámaras de manera similar a como en el proyecto completo al que este artículo pertenece.

Existen muchos antecedentes a este tipo de ingeniería. La sincronización de captura de imágenes es bastante común tanto en la academia como en la industria. Se utiliza para siempre que se desee poseer imágenes del mismo tiempo en el mismo espacio. A continuación se detalla la solución propuesta para este proyecto.

Descripción de la solución desarrollada

La solución propuesta se desvía considerablemente del plan original. La complejidad del software resultante es bastante simple y se compone de un solo script, el cuál contiene todos las funciones necesarias para capturar las imágenes, procesar las mismas y salvarlas en disco. A continuación se encuentra el diagrama UML y una descripción de cada una de las funciones principales.

Script
+ END_OF_CAPTURE: boolean
+ configure_main_camera_trigger(): void
+ configure_secundary_trigger(): void
+ retrieve_timestamp(image, nodemap): String
+ acquire_images (list): boolean
+ capture (list, int): boolean
+ main (): boolean

Figura 2: UML del Script de captura de las imágenes.

En la Figura 2 se observan las funciones que componen el Script y se detallan a continuación:

- **configure_main_camera_trigger(): void** Esta función es la encargada de configurar el disparador para la cámara principal. Esto es, modificar la configuración interna de la misma para que entregue la señal de disparo a las otrs dos cámaras del arreglo.
- **configure_secundary_camera(): void** Esta función configura ambas cámaras secundarias para que ajusten su configuración interna y obtengan la señal de disparo de la línea proveniente de la cámara principal.
- **retrieve_timestamp(): void** Esta función es la encargada de generar el nombre de cada imagen y su almacenamiento en disco. El nombre debe ser único, por lo que se genera a partir del ID de la cámara que la capturó y un identificador único que corresponde a un estampado de tiempo normalizado.
- **acquire_images(): void** Esta función recibe la lista de cámaras que posee el sistema conectado y para cada una de ellas comienza la captura de las imágenes. Obtiene la imagen con más antigüedad del *buffer* de cada cámara (modalidad FIFO) y posteriormente llama a la función *retrieve_timestamp()* para su guardado.

- **capture(): boolean** Esta función se encarga de inicializar las cámaras y desinicializarlas una vez terminada la captura de las imágenes.
- **main(): boolean** Esta es la función principal del algoritmo, es el punto de entrada del Script y orquesta su funcionamiento.

Del lado del hardware, el módulo diseñado es simplemente la conexión entre las cámaras para configurar y comunicar las señales del disparador de la cámara principal hacia las cámaras secundarias. El diagrama utilizado para este cableado se presenta a continuación en la Figura 3.

En la Figura siguiente se muestra en la parte superior el conector de GPIO de la cámara principal. Este consta de una resistencia *pull-up* de donde se genera la señal hacia las otras dos cámaras que se encuentran en la parte inferior de la imagen.

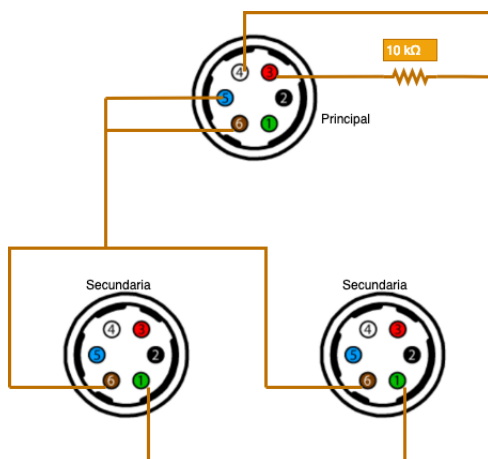


Figura 3: Circuito diseñado y soldado para el disparador por hardware.

Resultados obtenidos

Módulo de sincronización

Este es el principal módulo de este proyecto. Fue necesario implementar el circuito presentado en la sección anterior. Este módulo aún no se ha logrado que funcione de manera correcta como se requiere que

lo haga, pero se sigue trabajando en él hasta inclusive después de la entrega de este proyecto.

Otras alternativas que se pueden desarrollar, es el disparador por software, más simple de implementar, pero se teme que arroje resultados menos precisos que su contraparte por hardware. La principal dificultad de probar este módulo es lo hermético que es el circuito, lo que dificulta sus pruebas con instrumentos externos como osciloscopios, por ejemplo.

Estampado de tiempo

Es indispensable que todas las imágenes lleven consigo un estampado de tiempo para poder identificarlas y agruparlas las otras capturadas al mismo tiempo para su estudio.

La implementación original se basaba en un estampado de tiempo por software, pero este presenta la desventaja de que el estampado se genera una vez que la imagen llega a la computadora central, lo que puede generar tiempos incorrectos dado el tiempo de espera entre la captura y la generación del estampado por parte del algoritmo.

Se cambió entonces, a un estampado de tiempo por hardware. Las cámaras FLIR Blackfly®S permiten generar tal estampado al momento exacto de la captura. Este estampado funciona con la cuenta de un reloj interno que posee cada cámara. Este comienza a contar el tiempo desde el momento en que cada cámara fue conectada a la alimentación eléctrica. Esto presenta sus propios problemas, pues cada cámara se conecta en tiempos distintos a la alimentación y no se puede asegurar que siempre comiencen la cuenta todas al mismo tiempo. Por esta razón se tuvo que normalizar la cuenta del tiempo entre todas las cámaras.

Normalización del estampado de tiempo

Esta normalización se realizó de manera muy sencilla. Se captura en variables de software el tiempo de la primera imagen tomada por cada cámara. Este tiempo se le resta a todos los tiempos de las imágenes que se capturan posteriormente. Con esto, lo que se almacena y se asocia luego al nombre de cada imagen, es

la diferencia de tiempo desde la primer captura hasta la captura actual.

Identificación única de cada imagen

La identificación única se logra por medio de la unión del ID de cada cámara junto con el estampado de tiempo de cada imagen. Esta combinación que presenta la siguiente forma: *camera_id_timestamp*, se le asigna como nombre de guardado a cada imagen a la hora de su almacenamiento en disco.

Envío de imágenes

Este módulo no se realizó en este proyecto. Como se implementó todo el algoritmo en una sola computadora, no se realizó el módulo de envío y recepción de las imágenes por medio del protocolo TCP/IP como se tenía planeado.

Unión de módulo de sincronización de captura de imágenes con resto del proyecto

Queda pendiente el unir este módulo de sincronización con el resto del proyecto de captura de información del vuelo de las mariposas.

Conclusión

Este proyecto no se ha concluido al momento de la entrega de este proyecto. Se continúa trabajando hasta que se logre concluir todos los casos mencionados en la sección de Resultados obtenidos. Se logró concluir la mayoría de los casos que se propuso completar, salvo el sincronizador de las cámaras por hardware. Es este el punto que se continúa trabajando hasta que su funcionamiento sea correcto.

Se logró completar de manera satisfactoria los demás puntos establecidos al inicio del proyecto como importantes. El proyecto tiene como limitación actual que funciona únicamente con cámaras FLIR que puedan hacer uso del Spinnaker SDK, por lo que

una mejora que se puede realizar a futuro sería integrar un módulo para soporte a cualquier otro tipo de cámara que se desee.

Referencias

- [1] F. I. I. S. Inc., “Technical reference blackfly usb3 vision,” p. 44, 2018.
- [2] P. Guerra, C. Merlin, R. Gegear, and S. M. Reppert, “Discordant timing between antennae disrupts sun compass orientation in migratory monarch butterflies,” *Nature communications*, vol. 3, p. 958, 07 2012.
- [3] A. Noda, Y. Yamakawa, and M. Ishikawa, “Frame synchronization for networked high-speed vision systems,” in *SENSORS, 2014 IEEE*, Nov 2014, pp. 269–272.
- [4] J. Wang, Z. Miao, and Q. Liang, “Synchronization of cameras from human motion using feature points,” in *2014 12th International Conference on Signal Processing (ICSP)*, Oct 2014, pp. 1281–1284.
- [5] A. Kumar, P. S. Chavan, V. K. Sharatchandra, S. David, P. Kelly, and N. E. O’Connor, “3d estimation and visualization of motion in a multi-camera network for sports,” in *2011 Irish Machine Vision and Image Processing Conference*, Sep. 2011, pp. 15–19.