Sistema de detección y medición de peces en nubes de puntos para escalas de peces de hendidura vertical

Rico-Diaz, A.J.[a1], Pallas A.[a2], Rabuñal J.R.[a3], Puertas, J.[a4]

[a1, a2] Grupo RNASA-IMEDIR, Departamento de Tecnologías de la Información y las Comunicaciones. Universidade da Coruña, Campus de Elviña s/n. E-mail: a1angel.rico@udc.es

[a3] Centro de Innovación Tecnológica en Edificación e Ingeniería Civil (CITEEC), Campus de Elviña s/n. E-mail: juanra@udc.es

[a4]Grupo de Ingeniería del Agua y del Medio Ambiente, ETS de Ingenieros de Caminos, Canales y Puertos, Universidade da Coruña, Campus de Elviña s/n. E-mail: jeronimo.puertas@udc.es

Línea temática M | Tema monográfico.

RESUMEN

En este trabajo se pretende realizar la detección, medición y seguimiento de peces en entornos acuáticos y con baja luminosidad. Para ello se hará uso de un sensor 3D que emiten radiación infrarroja a la escena y a partir del comportamiento de la misma generan una matriz de puntos, llamada nube de puntos, que representa la superficie visible de la escena. La información que se recoge del sensor será posteriormente utilizada en un proceso de visión artificial siguiendo las fases de sustracción de fondo, la segmentación, la detección de peces y el seguimiento.
Palabras clave | Nube de puntos; Detección de peces; Kinect; Visión Artificial

INTRODUCCIÓN

El campo de la visión artificial en nubes de puntos es un campo completamente nuevo, desde los comienzos de la visión artificial se ha deseado modelar el entorno con la ayuda de sensores que pudieran, no solo obtener el color de los puntos de la imagen, sino también obtener medidas de la distancia a la que se encuentran de la cámara para poder dar forma a los objetos. Para realizar esta tarea se han utilizado variedad de trucos como el conocimiento previo de la escena, el uso de varias cámaras posicionadas estratégicamente para realizar la triangulación de puntos de referencia o la utilización de la perspectiva teniendo en cuenta ciertas limitaciones en la escena. No obstante, y pese a todos los estudios realizados no ha sido posible de realizar este tipo de aproximación tan solo con la imagen de manera genérica, eficiente, y sobre todo precisa. Con el lanzamiento de los sensores 3D es posible realizar una medición exhaustiva de cualquier tipo de escena, a 30 frames por segundo y con un margen de error del orden de micras. No fue hasta hace aproximadamente 10 años cuando los primeros sensores de este tipo comenzaron a ser lanzados al mercado a un precio desorbitado, afortunadamente el coste de este tipo de dispositivos se fue reduciendo drásticamente a medida que mejoraba la tecnología hasta llegar a hace 5 años que se comenzaron a comercializar versiones low cost de sensores 3D como la Kinect.

Con la bajada de precio de estos dispositivos, miles de desarrolladores en todo el mundo han investigado, experimentado y desarrollado soluciones que utilizan nubes de puntos en vez de imágenes para detectar los objetos. Este método le da un nuevo sentido a la resolución de problemas que antes se intentaban resolver a duras penas con tan solo la ayuda de una cámara, este tipo de sensores han tenido un éxito notable en campos como el de la robótica y son utilizados habitualmente en la industria para tareas como el control de calidad o el montaje en la cadena de producción.

Los sensores 3D (Bernardini y Rushmeier 2002) o escáneres 3D son dispositivos que mapean la escena generando una nube de puntos que se corresponde con la superficie de los objetos que se encuentran en ella, existen diferentes clases de sensores 3D que utilizan diferentes métodos:

Los sensores de contacto son un grupo de sensores 3D que utilizan métodos que obtienen la forma y posición del objeto a partir del contacto de una punta de acero o zafiro con el objeto, se utilizan generalmente con un brazo mecánico y tienen una precisión del orden de una micra, son muy utilizados en procesos industriales de fabricación y sus principales desventajas son su lentitud que existe la posibilidad de dañar objetos delicados.

Por su parte, los sensores sin contacto son aquellos sensores 3D que utilizan el comportamiento de radiaciones como por ejemplo la luz o los infrarrojos en la escena para obtener por medio de diferentes algoritmos la superficie de los objetos que se encuentran en ella. Pueden ser pasivos o activos.

Los pasivos son aquellos sensores 3D que no emiten ningún tipo de radiación sino que se apoyan en la reflejada en la escena, utilizan generalmente la luz visible, que suele ser muy abundante en el ambiente y hacen uso de una o varias cámaras lo que abarata su coste ya que no es necesario hardware especial. Como ejemplos de los sensores pasivos se pueden nombrar los estereoscópicos y los de silueta. Los primeros, utilizan un método basado en el funcionamiento de la visión humana mediante dos cámaras separadas por una distancia fijada y comparan las diferencias de las imágenes entrantes en cada una de ellas, siempre que se pueda establecer relaciones entre píxeles de una imagen con otra se puede saber la distancia a la que este se encuentra por medio de triangulación. Los de silueta obtienen varias imágenes de un objeto desde diferentes puntos de vista con un fondo conocido o fácil de extraer, el fondo es sustraído de la imagen y las distintas siluetas del objeto son cruzadas para obtener la superficie del mismo, el principal problema de este método es que no es capaz de mapear concavidades.

Los sensores activos son los sensores 3D que emiten algún tipo de radiación y generan la nube de puntos a partir del reflejo de la radiación emitida en la escena, su velocidad y precisión hacen que sean los más utilizados actualmente, el principal problema es que necesitan hardware especializado y en algunos casos el coste de estos dispositivos es muy elevado. Entre este tipo de sensores se pueden nombrar los Time of Flight (Li, 2014), los de luz estructurada (Fofi et al., 2004) y los de triangulación.

Los tiempo de vuelo o Time of Flight emiten algún tipo de radiación como puede ser un láser o luz infrarroja hacia la escena y a partir del tiempo que tarda esta radiación en ser capturada nuevamente por el sensor y conociendo la velocidad de la luz es posible calcular la distancia a la que se encuentra cada uno de los puntos de la escena, existen dos tipos de sensores que utilizan este método, los que utilizan un láser para la medición de la escena trabajan en tiempos del orden de pico-segundos entre que el láser es emitido y retorna, son conocidos por la precisión a la hora de realizar las mediciones pero su coste es muy elevado dado que se necesita un hardware de alta precisión.

El método de luz estructurada emite un patrón de radiación contra la escena que es capturado por un sensor que obtiene la distancia de cada punto basándose en la deformación del patrón emitido, este procedimiento puede ser aplicado tan solo con una cámara y un proyector pero requiere de una calibración previa del proyector y del sensor para realizar una correcta medición. Existen dispositivos que cuentan con las dos partes del sistema y no necesitan calibración por parte del usuario lo que simplifica su utilización. A la hora de generar patrones existen dos métodos principales, uno de ellos es el método por interferencia láser que proyecta dos patrones de luz variables que utilizan el código gray y el método por proyección, que es el utilizado por la Kinect, que imprime un patrón ya conocido y es recogido por una cámara que evalúa su deformación.

El método de triangulación utiliza un láser para iluminar puntos del entorno y una cámara cuya posición y rotación con respecto al láser es conocida, por lo que conociendo un lado del triángulo y dado el punto láser visualizado en la cámara es posible triangular la posición de ese punto obteniendo así su distancia con respecto a la cámara.

A partir de las imágenes RGB-D (Red, Green, Blue, Depth) que se recogen de estos sensores se puede proceder al proceso de detección en la nube de puntos. En la mayoría de los casos el proceso realizado comienza con el aprendizaje y el filtrado del fondo de la escena, seguido por la segmentación de los objetos que se encuentran claramente separados, posteriormente se realiza una detección de los mismos a partir de una clasificación fruto de diferentes métodos de aprendizaje máquina tanto supervisados como no supervisados sobre vectores de características o puntos característicos de los objetos.

Ya en 1999 algunos investigadores como Johnson y Herbert (1999) utilizaban varias imágenes spin (imagen en la que los puntos tienen una dirección, que en este caso se trata de la normal del punto) desde diferentes perspectivas al objeto buscando separar objetos ya conocidos que se encuentran en medio de varios objetos desconocidos, para esto, a partir de los puntos de la imagen con sus normales (imagen spin) obtenían un conjunto de puntos característicos utilizando PCA y posteriormente realizaban una búsqueda en la escena de esos puntos característicos y para extraer los objetos aprendidos anteriormente.

Más recientemente se han llevado a cabo métodos más sofisticados como (Lai et al., 2014) en donde se utiliza un método de aprendizaje no supervisado para la clasificación de los puntos característicos de diferentes objetos comunes en una vivienda, a partir de la utilización de voxels (cubos que iteran sobre la nube de puntos al igual que las ventanas lo hacen sobre una imagen), y comparan diferentes métodos existentes con el propuesto por ellos, llamado HMP3D, que realiza un aprendizaje de puntos característicos a partir la utilización de varias capas que utilizan voxels de diferente tamaño, con estos puntos característicos generan un diccionario que representa el tipo de objeto y de esta manera lo diferencian del resto.

MATERIAL

El sensor Kinect (Andersen et al. 2009) (Figura 1) es un escáner 3D que utiliza el método de luz estructurada para medir la distancia de cada uno de los puntos de la escena por medio de la proyección de un patrón de luz infrarroja que es captada por una cámara, para ello utiliza un emisor de infrarrojos y una cámara infrarroja, adicionalmente cuenta con una cámara RGB. Fue lanzada en 2010 por Microsoft como dispositivo de su consola XBOX 360 para dotar a los juegos de interfaces naturales y atraer a los usuarios que buscaban una experiencia de juego diferente.

Figura 1 | Dispositivo Kinect utilizado.

El hecho de que una gran empresa se ponga a fabricar en masa este tipo de dispositivos hizo que sus precios descendieran drásticamente en comparación con sensores similares en existentes en el mercado, al poco tiempo, la empresa creadora de la tecnología usada por Kinect liberó sus drivers, lo que atrajo a gran cantidad de desarrolladores que buscaban un escáner 3D asequible para realizar sus proyectos o simplemente divertirse utilizando el PC como herramienta.

En 2011 debido al gran éxito del dispositivo entre los desarrolladores Microsoft se unió a la corriente lanzando una nueva versión del dispositivo llamada "Kinect for Windows" a un precio muy superior al de la anterior y con prácticamente las mismas funcionalidades, pero dirigido al uso comercial. Microsoft también lanzó su propio SDK (https://msdn.microsoft.com/en-us/library/hh855347.aspx) para desarrollar con la Kinect y la principal diferencia entre la Kinect de XBOX 360 y la "Kinect for Windows" es que utilizando SDK de Microsoft en la primera existe una limitación inferior de 80 cm de distancia y en la segunda contamos con un modo llamado "Near Mode" que reduce esta limitación a 40 cm.

MÉTODO

En este trabajo se van a utilizar dos tipos de frames que sirve el sensor, así como la combinación de los mismos. El frame infrarrojo que se trata de una matriz de compuesta por píxeles que representan en 8 bits la intensidad de la luz infrarroja reflejada en ese punto, obtenido con una frecuencia de 30 por segundo. El sensor emite un patrón de luz infrarroja que sirve para medir la distancia a los puntos de la imagen y este patrón y su variación es claramente visible en el frame de infrarrojo.

El otro tipo de frame que se va a utilizar es el de profundidad y que se trata del más característico en este tipo de sensores. A partir del patrón emitido, el sensor calcula la distancia de cada uno de los píxeles obtenidos en el frame infrarrojo. Con esta información genera una matriz que representa esta distancia codificada en 16 bits. La frecuencia de obtención de este frame también es de 30 por segundo.

La combinación entre el frame infrarrojo y el frame de profundidad parece trivial, ya que el de profundidad es calculado directamente a partir del infrarrojo y cada píxel I(N,M) del infrarrojo representa el mismo punto que su píxel homólogo P(N,M) en el de profundidad, por lo que superponiendo ambos frames podemos realizar operaciones con ellos u obtener un nuevo frame con dos canales.

Calibración del sensor

Para la calibración del sensor se ha tenido en cuenta el conocimiento que se ha obtenido en el estudio previo de los análisis del sensor realizados por terceros (Andersen et al. 2009). El proceso de calibración del sensor está orientado a calibrar la respuesta del frame de profundidad con respecto al entorno. Tiene como objetivo comprobar de manera formal el rango del sensor, el significado de sus mediciones, su margen de error, su relación con el frame de infrarrojos y la capacidad de reflejo que tienen los diferentes objetos del patrón infrarrojo emitido, ya que si no lo reflejan correctamente se producen zonas no alcanzables o intermitentes.

A mayores, se tiene la complejidad de que los objetos se encuentran tras un cristal y en un medio que puede producir distorsiones. Por esta razón se han realizado diferentes tipos de calibraciones variando la distancia y el ángulo del sensor con respecto al cristal e introduciendo varios peces en diferentes ángulos y posiciones dentro de la pecera. Con este procedimiento se desea establecer un método para calcular la distancia real de un objeto dentro de la pecera teniendo en cuenta cuantos centímetros hay entre el cristal de la pecera y el objeto, para esto se ha colgado un objeto del techo para que se mantenga inmóvil en una posición determinada y se ha colocado la Kinect a 84 cm de él, posteriormente se ha introducido la pecera de tal manera que el objeto quedara dentro de ella. Se han tomado las mediciones y capturas pertinentes a lo largo del proceso, moviendo tan solo la pecera para variar la distancia del objeto al cristal, variando así la cantidad de agua que existe entre ellos. Los resultados han indicado que la Kinect subestima la distancia real al objeto cuando existe agua de por medio.

A partir de los resultados de este experimento, se ha aprendido la función (Ecuación 1) para el cálculo de la distancia real al objeto conociendo la distancia a la pecera y la medición de la Kinect, esta función es útil para realizar correcciones pero tan solo se trata de una aproximación ya que las pruebas solo se han realizado en las zonas centrales de la imagen:

$$Dist(Sensor,P1) = Dist(Sensor,Cristal)+(Medicion(P1)-Dist(Sensor,Cristal))*1,35 \quad (1)$$

La constante obtenida de las mediciones para la corrección de la distancia recorrida por la luz dentro del agua es muy similar al índice de refracción del agua por lo que se entiende que la causa de que la Kinect subestime la distancia es la refracción del agua.

Además, se ha comprobado que la distancia óptima a la que debe encontrarse el sensor del cristal es de entre 60 y 70 cm.

Medición de los peces

A partir de la máscara del pez (imagen que engloban los pixeles que pertenecen al pez) se extrae el contorno del objeto con la función findContours de OpenCV (Bradski y Kaehler 2008) y se busca la elipse que minimice la distancia a los puntos del contorno obtenido con la función fitEllipse, tras esto se obtienen los 4 extremos de los dos ejes de la elipse.

Posteriormente se procede a obtener los puntos extremos del objeto y su posición en el espacio real. Para realizar esta tarea se buscan los puntos del contorno más cercanos a los puntos extremos de los ejes de la elipse y estos son asignados como puntos extremos del objeto, la función findContours obtiene siempre contornos pertenecientes al objeto por lo que obteniendo el valor de los puntos extremos del objeto en el frame de profundidad obtendremos las distancias de estos 4 puntos al plano de la Kinect.

Con la distancia de los puntos a la Kinect y su posición en el frame de profundidad utilizamos la función NuiImageTransform (http://opencv.org/) proporcionada por el SDK para obtener las coordenadas de los puntos con respecto a la Kinect en el espacio real.

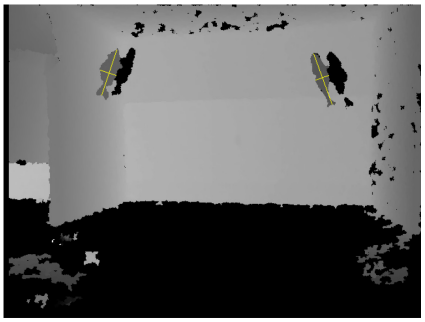

Figura 2 | Frame con medición de objetos.

A partir de las posiciones reales se calcula la distancia euclídea entre los puntos extremos obteniendo así la longitud y la anchura del objeto. A mayores hay que tener en cuenta la corrección que hay que realizar por encontrarse el objeto en el medio acuático y que se ha calculado en la etapa de calibración del sensor.

En la Tabla 1 se muestran los resultados de diversos experimentos realizados para comprobar el funcionamiento de la técnica.

Tabla 1 | Resultados de mediciones realizadas.

	Posición (x,y,z) (mm)	Distancia a pecera (mm)	Tamaño Real (mm)	Medición Estándar (mm)	Medición con corrección (mm)
Experimento 1	(121,-14,650)	650	110 x 30	99 x 27	100 x 28
Experimento 2	(4,-24,832	650	100 x 32	96 x 28	104 x 31
Experimento 3	(-64,10,657)	647	110 x 30	101 x 29	106 x 32
Experimento 4	(-14,-17,836)	647	100 x 32	82 x 27	87 x 29
Experimento 5	(206,28,647)	647	100 x 32	94 x 32	99 x 33
Experimento 6	(-78,-24,678)	647	100 x 32	114 x 34	115 x 34
Experimento 7	(-81,-4,648)	613	110 x 30	100 x 34	102 x 35

Sustracción del fondo

Se necesita realizar una sustracción del fondo para poder centrar los recursos en las regiones donde exista movimiento, que serán las que no pertenecen al fondo y donde se encuentren los objetos de la escena.

La sustracción de fondo es una disciplina ampliamente conocida en el ámbito de la visión artificial y es utilizada como preprocesado en numerosos problemas de detección de objetos. Esta disciplina se basa en filtrar aquellas áreas de la escena que pertenecen al fondo. Este método permite descartar gran parte de la información, obtener las formas de los objetos de la escena y centrar los recursos de detección en las áreas donde estos se encuentran. El problema se basa en la obtención de una máscara en la que únicamente los píxeles pertenecientes a los objetos de la escena estén iluminados, conociendo el fondo y la imagen actual de la escena. En ciertas situaciones, en las que el fondo no puede ser pasado como entrada al algoritmo, existe el problema de aprendizaje del fondo, común en aproximaciones en la que existe un bajo control sobre la escena, esta es dinámica o el sensor cambia su posición.

Para la substracción de fondo en el frame de profundidad (Figura 3a) se ha utilizado un método de aprendizaje por votación modificado para poder ser utilizado en tiempo real. Este tipo de aproximación está basada en aquellos algoritmos que aprenden estructuras por medio de votación de candidatos, el algoritmo de Hough es un ejemplo de ello, ya que aprende los círculos en una imagen a partir de la votación que realiza cada uno de los puntos del borde a posibles círculos con diferentes centros y radios.

La metodolgía genérica consiste en obtener frames de la escena en intervalos regulares (k frames) y cada uno de los píxeles del frame vota a una serie de candidatos a fondo establecidos por (valor píxel (+-) n). Se asigna a n el valor de 3 ya que es la variación que puede tener un píxel estable del frame de profundidad. Para cada píxel existe una lista de urnas, las urnas son registros que contienen el valor, el número de votos y una marca que indica si está abierta o cerrada. Esta lista de urnas se inicializa vacía, pero cuando se produce el primer voto se añaden varias urnas a la lista con los valores circundantes (una por cada valor circundante), un voto y marcadas como abierta (en el caso de ser 0 solo se añade su urna). A continuación, a medida que van sucediéndose nuevos votos se comparan los valores circundantes del valor con las urnas de la lista, en el caso de que coincida con una de ellas la urna aumenta en uno sus votos y se marca como abierta, en el caso de no coincidir con ninguna se añade una nueva urna. El algoritmo utiliza también un intervalo de refresco (N) que será múltiplo del intervalo de captura de frames, una vez cada (k*N frames) el algoritmo eliminara de las listas de urnas aquellas que se encuentren cerradas y cerrará las que se encuentren abiertas. Tras esto se realizará el recuento y se establecerá como fondo el valor más votado, y aquí se encuentra la ventaja del algoritmo, en el caso de que el valor más votado sea 0 (no alcanzable) se asignara el segundo valor más votado si cumple la característica de tener al menos un 20% de los votos que tiene el 0. Este método es capaz de dar la importancia que se merece a aquellos valores que han sido referenciados constantemente durante largos periodos de tiempo y aunque en un intervalo determinado la frecuencia del fondo sea menor esto no será un problema ya que persiste el número de votos. No obstante si estos valores no son referenciados en los N frames obtenidos antes del cierre de las urnas los elimina, lo que produce que se puedan aprender fondos de manera dinámica. Este método también soluciona el problema de las regiones intermitentes, ya que tanto el 0 (no alcanzable) y el fondo real persistirán a lo largo del tiempo logrando gran cantidad de votos, y a la hora de clasificar el 0 como fondo sabremos si se trata de una región no alcanzable o una intermitente mirando los votos del segundo más votado.

El algoritmo mencionado anteriormente es demasiado complejo computacionalmente para un sistema en tiempo real, por lo que se han realizado una serie de modificaciones que agilizan su comportamiento y mantienen su filosofía. La modificación principal se produce en el sistema de votación, cuando un nuevo píxel es capturado tan solo vota por su valor y no se realizan votaciones en todo el intervalo que lo rodea (n*2+1 votaciones), pero a la hora de introducir el voto las urnas no solo representan su valor sino que representan su valor y n valores superiores e inferiores, de manera que cuando entra un nuevo valor que no cumple las restricciones (valor+-n) de las urnas existentes se crea una urna como en el proceso anterior, pero cuando el valor cumple las restricciones de alguna de las urnas esta es actualizada en número de votos, marcada como abierta y su valor es modificado 0.1*nuevoValor + 0.9*antiguoValor.

Con la modificación realizada se agiliza el proceso ya que no es necesario almacenar una urna para cada uno de los valores y además cuando se realiza la votación tan solo es necesario actualizar una urna. Teniendo en cuenta que todo este proceso se realiza para cada uno de los píxeles del frame es importante optimizar el proceso lo máximo posible, pero sin perder eficacia, en este caso el algoritmo funciona aproximadamente al doble de velocidad y los resultados obtenidos son tan buenos como en la aproximación principal.

Tras realizar este proceso se genera la máscara que indica movimiento restándole al frame el fondo y quedarse con aquellos píxeles que superen un determinado umbral que se ha asignado en 15, en este caso, debido a las características del frame de se han añadido a mayores las restricciones de que si el fondo es 0 y el frame es distinto de 0 existe movimiento, si el frame es 0 nunca existe movimiento y existe movimiento si el frame se encuentra más cerca del fondo y no al contrario (Figura 3c).

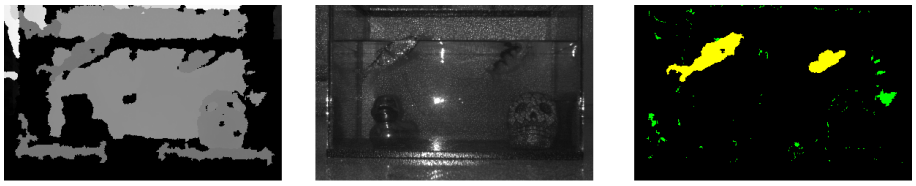

Figura 3 | Resultado de la sustracción de fondo (a) frame de profundidad (b) frame de infrarrojos (c) frame resultado de la sustracción utilizando el método de votación.

Segmentación

El primer paso del proceso consiste obtener todas las regiones de movimiento y filtrar aquellas que sean demasiado pequeñas, para ello se ha aplicado la función de OpenCV findContours (obtiene los contornos de una imagen binaria) a la máscara de entrada y se han filtrado los contornos que no superen un área de 100 píxeles. Posteriormente para cada una de las regiones de movimiento se obtiene el marco mínimo que contiene a cada uno de los contornos, a partir de este marco se ha obtenido una imagen de profundidad de la región recortada del frame de profundidad original y se ha generado un blob de la región, que se trata de una máscara del tamaño del marco en la que se encuentra marcada el área interior al contorno (sin restos de otras regiones que pueden estar incluidas en el marco) (Figura 4c).

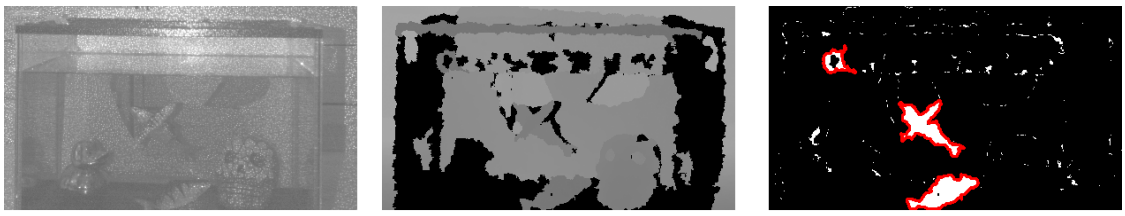

Figura 4 | Proceso de segmentación: (a) frame infrarrojo, (b) frame de profundidad, (c) resultado de la segmentación

Para separar los objetos que se encuentran dentro de una misma región, se ha utilizado el detector de bordes Canny (Canny, 1986) sobre la imagen de profundidad teniendo en cuenta que el margen de error del frame de profundidad para los valores distintos de 0 es bastante bajo. Sin embargo OpenCV tan solo ha implementado Canny para imágenes de 8 bits por lo que es necesario reducir el histograma para pasar de 16 bits a 8 bits. La pérdida de información es evidente por lo que esta reducción tan solo se aplica para realizar la detección de bordes y posteriormente se seguirá trabajando con la imagen de 16 bits. No obstante para realizar una correcta detección de bordes la reducción del histograma se ha realizado de manera que se pierda el mínimo de información. Para esto se han puesto a 0 los valores de la imagen de profundidad de la región que no pertenecen al blob y se obtenido el máximo y el mínimo (distinto de 0) de la misma. Si la diferencia entre ellos es menor que 255 el máximo se asigna a mínimo+255 y posteriormente se calcula el multiplicador utilizado para la reducción mult = (max-min)/255, si no se

realiza este proceso se el valor de los píxeles resultantes en la imagen de 8 bits dependerá de la heterogeneidad de la de 16 bits y no se mantendrá el sentido de los datos ni se podrá establecer correctamente un umbral para Canny.

Detección

Una vez logrado el objetivo de separar y medir todos los objetos candidatos a ser peces, si no se aplica ningún tipo de filtro adicional, existe la posibilidad de que algún objeto extraño que se encuentre dentro de la pecera o una parte del fondo que se mueva generen un falso positivo. Para evitar que esto suceda se ha de realizar un proceso de detección basado en la obtención de características clave de los objetos obtenidos.

Las siluetas obtenidas de la fase de segmentación cuentan con su imagen de profundidad, su máscara y su contorno. Se puede hacer uso de estos tres elementos para la extracción de características que sirvan para posteriormente evaluar si se trata de un pez, se trata de una parte de un pez y es necesario combinarla con otra o tan solo es un objeto extraño de la escena.

El proceso de obtención de características se encuentra relacionado con el proceso de medición dado que algunas de las características obtenidas han sido explicadas en ese apartado. Las características que se utilizan para la detección son: la elipse que minimiza el error de los puntos del contorno, el área en píxeles ocupada por el contorno, los extremos de la silueta en el espacio real, la longitud y la anchura de la silueta, la posición de la silueta en el espacio real, la envolvente convexa y el área de la envolvente convexa.

Seguimiento

La fase de seguimiento es la encargada de mantener la persistencia de la información de las detecciones del sistema, en las fases anteriores se obtienen los peces que se encuentran en el frame actual sin tener en cuenta los frames anteriores y posteriores pero una aplicación que cuenta con la posibilidad de obtener varios frames a lo largo del tiempo esta´ desaprovechada si no realiza ningún tipo de seguimiento para mantener la persistencia y la coherencia de las detecciones realizadas entre frames sucesivos.

Se ha implementado un método simple de seguimiento basado en el emparejamiento de los objetos detectados en el frame N con aquellos que se encuentren más cercanos a ellos en el frame N+1 (Figura 5), se ha utilizado la ventaja de que se conoce la posición en el mundo real y por lo tanto se ha dado prioridad a los objetos más cercanos a la cámara a la hora de realizar el emparejamiento. A continuación, se explicará el proceso partiendo desde el primer frame y detallando las operaciones realizadas.

En primer lugar se obtienen los peces detectados en la fase de detección, nos encontramos en el primer frame y no tenemos con que emparejarlos por lo que para cada detección guardamos una firma del pez con la posición, la altura, la anchura y los 4 puntos extremos de la silueta, les asignamos un color al azar y un contador que tiene como valor 5 unidades.

Con la llegada de las detecciones del segundo frame el sistema ordena la lista de firmas de peces almacenada poniendo al principio aquellas que se encuentren en una posición más cercana a la cámara (eje z menor). Tras realizar la ordenación itera sobre todas las detecciones del segundo frame buscando aquella que minimice la suma de las distancias de sus puntos extremos.

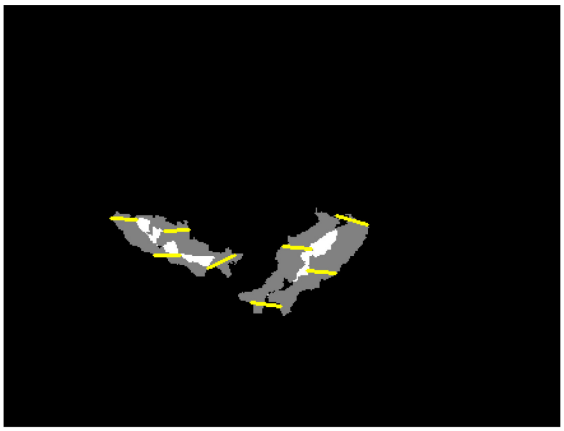

Figura 5 | Seguimiento de dos siluetas en frames consecutivos por medio de la minimización de la distancia de sus puntos extremos

Tras esto se comprueba que la silueta seleccionada cumple que la suma de las distancias de sus puntos extremos a los puntos extremos de la firma es menor que la suma de la longitud y la anchura del pez. Si se cumple esta restricción la nueva detección es asignada a la firma y la firma es actualizada con los nuevos valores de posición, tamaño y puntos extremos, tras asignar la firma o en el caso de no cumplir las restricciones, se pasa a la siguiente firma y se repite el proceso sin utilizar las detecciones ya asignadas.

Cada vez que una firma es actualizada se reinicia su contador a 5 y cada frame que pasa sin que una firma sea actualizada el contador disminuye en 1, en el caso de llegar a 0 esta firma es eliminada del sistema.

Adicionalmente se mantiene un contador de la cantidad de veces que la firma ha sido actualizada y no será mostrada por pantalla a no ser que este contador supere las 3 unidades, esto sirve para evitar falsos positivos en frames determinados que realizan una mala medición, o la sustracción de fondo genera artefactos que pueden parecer peces y el detector no es capaz de filtrarlos. Las posibilidades de la ocurrencia de esto son extremadamente bajas, pero teniendo en cuenta que trabajamos a 30 frames por segundo si ocurriera un error en un frame de cada 500 habría un error cada 16 segundos, en cambio utilizando este método la probabilidad de error disminuye drásticamente.

Al final del proceso si existen detecciones que no han sido asignadas se generan nuevas firmas con sus datos ya que se entiende que son peces nuevos que han aparecido.

RESULTADOS

Las pruebas realizadas con la Kinect en cualquiera de los dos escenarios deben cumplir los siguientes objetivos:

- Comprobar que en cada frame seleccionado todos los peces son detectados y separados en el caso de estar solapados. Si existe solapamiento solo es necesario que sea detectado el que se encuentra más cerca.
- Comprobar que los peces detectados son medidos correctamente con un margen de error de 1 cm de altura y 1 cm de anchura.
- Realizar un avance de máximo 50 frames para comprobar que los peces detectados son seguidos correctamente al menos hasta dejar de ser visibles en el frame de profundidad o ser solapados por otro pez.

Las primeras pruebas se realizan en un entorno artificial (Tabla 2) compuesto por una pecera y 1 ó 2 peces artificiales moviéndose dentro de ella (Figura 6). En esta prueba se obtendrán los instantes en los que un pez es detectado y el frame anterior a perderlo, con esto se comparará la medición del frame final con la real y se calculará el número de frames en los que el pez ha sido seguido y la razón de que haya sido perdido.

Tabla 2 | Resultados de experimentos en entorno artificial.

	Nº de frames	Medida real (mm)	Medida del Sistema (mm)
Experimento 1	43	110 x 30	111 x 24
Experimento 2	29	110 x 30	111 x 29
Experimento 3	33	110 x 30	113 x 31
Experimento 4	49	110 x 30	112 x 31
Experimento 5	46	110 x 30	114 x 31
Experimento 6	29	110 x 30	111 x 30
Experimento 7	60	100 x 32	102 x 28

Los resultados en las mediciones son buenos pero el sensor tiene problemas para detectar a los peces en ciertas posiciones por lo que no es capaz de seguirlos durante demasiado tiempo.

Figura 6 | Imagen del resultado en las pruebas en entorno artificial.

Las pruebas en entorno real se realizan en un tanque de agua lleno de peces de tamaño mucho mayor (más del doble) al de los utilizados en las pruebas anteriores (Figura 7). Estos ensayos se llevaron a cabo en el Aquarium Finisterrae de A Coruña.

Tabla 3 | Resultados de experimentos en entorno real.

	Nº de frames	Medida del Sistema (mm)
Experimento 1	12	216 x 37
Experimento 2	25	223 x 57
Experimento 3	23	239 x 54
Experimento 4	27	180 x 40

En estas pruebas no es posible verificar (Tabla 3) que las mediciones son correctas pero son muy estables en frames en los que los peces se ven completamente por lo que teniendo en cuenta que en el entorno artificial el sistema mide casi perfectamente se asume que aquí también lo hace, a la hora de evaluar el seguimiento funciona mucho mejor que en las pruebas en el entorno artificial debido a que el tamaño de los peces hace que ruido del entorno no sea suficiente como para recortarlos.

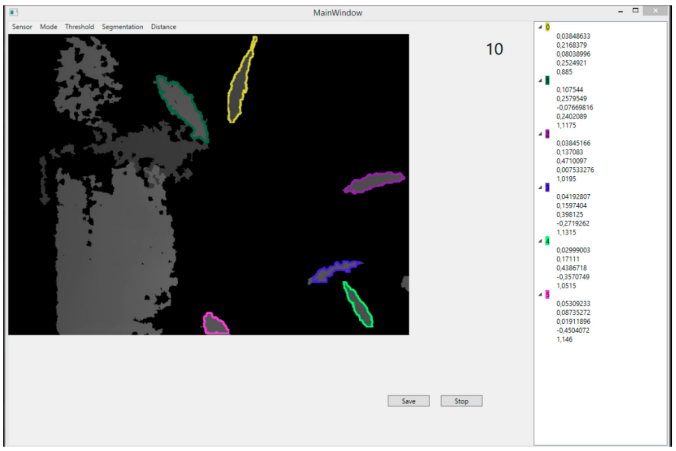

Figura 7 | Imagen del resultado en las pruebas en entorno real.

CONCLUSIONES

Se ha diseñado e implementado un sistema que aporta una solución a la detección, medición y seguimiento de peces utilizando sensores 3D. Cabe señalar que la Kinect está limitada en cuanto a visión y rango dentro del agua pero sus mediciones son muy precisas.

Para probar la técnica se han realizado ensayos en un entorno artificial, como punto de partida, y una vez comprobado su correcto funcionamiento se han trasladado estos experimentos a un entorno con peces reales.

Esta técnica puede ser implementada en zonas de paso de peces, como pueden ser las escalas de hendidura vertical, o en granjas de peces y piscifactorías.

AGRADECIMIENTOS

Este trabajo ha sido cofinanciado con fondos FEDER y por el Ministerio Español de Economía y Competitividad. Subprograma estatal de formación del Programa Estatal de Promoción de Talento y su Empleabilidad en I+D, en el marco del Plan Estatal de Investigación Científica y técnica y de Innovación 2013-2016 (FPI Convocatoria 2013) (Ref. del proyecto CGL2012-34688 Ref. de la ayuda BES-2013-063444). Los autores también quieren agradecer al personal directivo y técnico del Aquarium Finisterrae de A Coruña por su apoyo, asistencia técnica y permitir realizar la experimentación en sus acuarios.

REFERENCIAS

Andersen, M.R., Jensen, T., Lisouski, P., Mortensen, A.K., Hansen, M.K., Gregersen, T. y Ahrendt, P. 2012. Kinect depth sensor evaluation for computer vision applications.

Bernardini, F. y Rushmeier, H. 2002. The 3D model acquisition pipeline. In Computer graphics forum, Anonymous Wiley Online Library, , 149-172.

Bradski, G. And Kaehler, A. 2008. Learning OpenCV: Computer vision with the OpenCV library. .O'Reilly Media, Inc.", .

Canny, J. 1986. A computational approach to edge detection. Pattern Analysis and Machine Intelligence, IEEE Transactions on 679-698. .

Fofi, D., Sliwa, T. y Voisin, Y. 2004. A comparative survey on invisible structured light. In Electronic Imaging 2004, Anonymous International Society for Optics and Photonics, , 90-98.

Johnson, A.E. y Hebert, M. 1999. Using spin images for efficient object recognition in cluttered 3D scenes. Pattern Analysis and Machine Intelligence, IEEE Transactions on 21, 433-449. .

Lai, K., Bo, L. y Fox, D. 2014. Unsupervised feature learning for 3d scene labeling. In Robotics and Automation (ICRA), 2014 IEEE International Conference on, Anonymous IEEE, , 3050-3057.

Li, L. 2014. Time-of-Flight Camera–An Introduction. Technical White Paper, May .

Kinect for Windows SDK 1.5, 1.6, 1.7, 1.8. [consulta: 10 Junio 2017]. https://msdn.microsoft.com/en-us/library/hh855347.aspx

OPENCV (OPEN SOURCE COMPUTER VISION). [consulta: 10 Junio 2017]. http://opencv.org/