Identificación y seguimiento de personas usando kinect por parte de un robot seguidor*

Òscar Franco Xavier Perez-Sala Cecilio Angulo
UPC - Universitat Politècnica de Catalunya
CETpD - Centre d'Estudis Tecnòlogics per atenció a la Dependència i la Vida Autònoma
Rambla de l'Exposició 59-69 Planta 2, 08800 Vilanova i la Geltrú
{oscar.franco, xavier.perez-sala, cecilio.angulo}@upc.edu

Abstract

El presente trabajo aborda uno de los problemas que surgen en la interacción entre un humano y un robot durante la navegación conjunta: el seguimiento de una persona por parte de un robot, así como la posibilidad de intercambiar sus roles de guía - seguidor. Se ofrece como solución para la identificación y seguimiento del humano por parte del robot un algoritmo basado en la evaluación de su posición en función de las condiciones dinámicas y su identificación por medio del histograma de color. El sistema se ha demostrado fiable en la experimentación cuando el sensor se procesa desde un ordenador personal en situación de reposo. Sin embargo, los movimientos bruscos del robot empleado, su reducida velocidad de movimiento y la baja velocidad de procesado de su ordenador de a bordo, provoca que el porcentaje de éxito en el seguimiento se reduzca cuando se halla en condiciones de reales de seguimiento de una persona.

1. Introducción

El posicionamiento permanente de un robot, en una zona cercana a una persona, es de gran utilidad para diversas aplicaciones de interaccion, pero aún es uno de los problemas a resolver en navegación robótica. Un ejemplo de estas aplicaciones, en el marco de la robótica asistencial, es el que se presenta en [Endo *et al.*, 2009]: se utiliza un robot móvil para mantener un tanque de oxígeno cerca del paciente que lo requiere, evitando el esfuerzo que le supone arrastrarlo. Otro ejemplo sería el que se discute en [Kristou *et al.*, 2009; Laumond, 1993], donde se plantea el uso de una plataforma robótica móvil de seguimiento para el transporte de maletas de viajeros en un aeropuerto. Por último, por citar otro escenario habitual, en [Thrun *et al.*, 1999] se da uso a un robot que sirve de guía a un grupo de personas por un museo.

*Este trabajo ha sido realizado dentro del marco del proyecto ACROSS (TSI-020301-2009-27), aprobado por el subprograma Avanza I+D dentro de la convocatoria de ayudas de Acción Estratégica de Telecomunicaciones y Sociedad de la Información 2009, financiado por el Ministerio de Industria, Turismo y Comercio (MITYC) y el Fondo Europeo de Desarrollo Regional (FEDER).

Para conseguir de froma efectiva el seguimiento de una persona por parte de un robot, éste debe conocer primero la posición de la persona con la que interactúa. Existen diferentes aproximaciones en la literatura a este problema: cuando sea posible actuar sobre el entorno de trabajo, se pueden establecer marcas en el usuario (visuales, RFID...) que provean de información al robot; por otra parte, cuando el problema se pretende solucionar sin modificar el entorno, se puede considerar la fusión de información aportada por sensores telemétricos (Laser Range Finders (LRFs), kinect..) a los datos relativos al sistema de visión del robot. La propuesta de solución presentada en este trabajo se basa en el uso de kinect como sensor principal de la plataforma robótica [Kinect, 2011]. El sensor comercial kinect se trata de un sistema de visión en profundidad, RGB-Depth, que permite, de manera sencilla, segmentar la imagen gracias a la información de profundidad del sensor. En la Figura 1, se pueden observar las imágenes obtenidas de los dos sensores de kinect para un mismo instante de tiempo. La imagen superior (Figura 1(a)) es la que se obtiene con el sensor CMOS en RGB, mientras que la imagen inferior (Figura 1(b)) es el resultado de convertir los datos del sensor de profundidad a escala de grises.

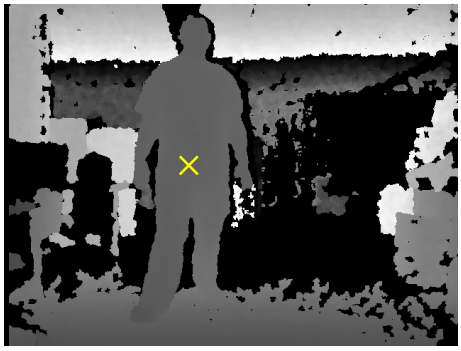
La aparición en el mercado de un sensor con estas buenas prestaciones a un precio muy asequible ha disparado el número de implementaciones y aplicaciones que utilizan este sistema [Leyvand *et al.*, 2011]. De entre todas ellas, es de destacar el middleware PrimeSense NITE (Natural interaction) [NITE, 2011], el cual permite comunicarse con los sensores de audio, vídeo y sensor de profundidad de kinect. A su vez, este middleware proporciona una API que facilita el desarrollo de aplicaciones que funcionen con interacción natural, por gestos y/o movimientos corporales. Además, haciendo uso del framework de OpenNI [OpenNI, 2011], se obtiene una primera descripción semántica del entorno a un coste computacional muy bajo.

En la aplicación que nos ocupa en este trabajo, el middleware original permitirá extraer elementos diferenciados, clasificados como posibles personas, a partir de la descripción semántica. Sin embargo, el algoritmo original, de código cerrado, es poco robusto: detecta objetos o muros como personas y deja personas sin detectar. Además, no es muy fiable en cuanto a la identificación de los usuarios en sucesivos fotogramas o 'frames'. Para solventar estos problemas se propondrá un algoritmo que permita comprobar en modo continuo

(a) Sensor RGB

(b) Sensor Depth

Figura 1: Imágenes obtenidas por los sensores de Kinect.



(a) Usuario detectado en plano Depth

(b) Usuario segmentado mediante el algoritmo

Figura 2: Segmentación del usuario utilizando el nuevo algoritmo, tomando como base el de OpenNI.

la coherencia de la posición medida del usuario mediante la evaluación de sus condiciones dinámicas y la comprobación de su identidad por histograma de color.

Más concretamente, el algoritmo propuesto consiste en:

1. Identificar el usuario con el que interactuar, mediante:
   - Posición inicial.
   - Histograma del color.
2. Para cada frame:
   a) Estimar la posición del usuario relativa al sensor.
      1) Obtener las coordenadas cartesianas de todos los usuarios detectados mediante OpenNI.
      2) Comprobar si la identificación del usuario llevada a cabo por el módulo OpenNI es correcta:
         - *Criterio a priori*: Evaluar la fiabilidad de la posición obtenida, en función del estado (posición y velocidad) anterior de los usuarios.
         - *Criterio a posteriori*: De ser necesario, comprobar la identidad del usuario comparando histogramas de color.

   b) Mantener el robot a una distancia objetivo del usuario.

La extracción del histograma de color de un usuario se realiza a través de las dos cámaras de kinect. Estas, deberán estar previamente calibradas para poder relacionar los píxeles entre los planos de las imágenes RGB y Depth.

En el caso que el usuario sea reconocido por el algoritmo base de OpenNI, como en la Figura 2(a), el middleware genera una máscara etiquetando los píxeles del plano de profundidad que pertenecerían al usuario. Al aplicar la máscara, y realizando el cambio de plano correspondiente al de la imagen RGB, se obtiene la imagen de la Figura 2(b). Son estos últimos los píxeles que se usarán para el cómputo del histograma de color.

De este modo, el sistema aporta una detección robusta de usuarios sin necesidad de introducir marcas en el entorno, ni tampoco de inicializarse ('set-up') a través de una *pose* específica del usuario, como sucede en la aplicación original, que resulta poco cómoda y poco efectiva para la interacción.

2. Desarrollo algorítmico

El algoritmo propuesto, a ser implementado sobre un robot dentro de un sistema de control general, debe permitir detectar y contornear a un usuario para su posterior reconocimiento. Este sistema, además, informará de la posición del usuario detectado respecto al robot. El algoritmo servirá para los dos objetivos duales propuestos, tanto para que el robot pueda seguir a la persona, como para que la persona pueda seguir al robot sin que ésta se aleje demasiado.

En ambos casos el funcionamiento será muy parecido, ya que el robot deberá permanecer a una distancia objetivo del usuario. Sin embargo, en el caso que el robot siga a la persona, la consigna de movimiento se determinará exclusivamente

a partir de la posición relativa del robot. Mientras que, en el otro caso, la consigna que recibirá el robot la determinará un sistema superior de control navegación; entonces, de forma autónoma, será el propio robot quien modifique esta consigna, valorando su distancia con el usuario.

2.1. Inicialización

Cuando el sistema inicia la aplicación, el robot espera en reposo hasta la aparición de un usuario en su campo de visión. Este evento inicia un proceso de adquisición de las características más relevantes del usuario objetivo. En primer lugar se obtiene información relativa al espacio, en píxeles, ocupado por el usuario detectado en las imágenes capturadas. El algoritmo espera hasta que el usuario aparece totalmente dentro del campo de visión y no queda cortado por los márgenes de la imagen. Estas condiciones se cumplen de forma óptima en el caso que el usuario se sitúe en medio del rango horizontal de visión del sensor, a una distancia aproximada de 2 metros, dependiendo de la altura del usuario.

Una vez se ha identificado un candidato a usuario en la zona esperada, se dimensiona un cubo en el que se hallará el sujeto. Se pueden configurar diferentes umbrales para determinar si el volumen detectado podría pertenece a una persona. Como resultado de la experimentación llevada a cabo, se han establecido unos márgenes en anchura (0,4~1,5m) y en altura (1,5~2m) con el fin de eliminar de la identificación posibles objetos como muros o sillas. Se debe tener en cuenta, sin embargo, que el movimiento de las articulaciones por parte del usuario puede hacer variar de forma radical el volumen del cuerpo detectado. Es por ello que esta fase del algoritmo será la de menor prioridad en caso de discrepancia con otras validaciones durante la tarea de reconocimiento.

Cuando se han validado las dimensiones del candidato detectado, se toman datos relativos al color de su vestimenta. El cálculo del histograma de color se realiza sobre la imagen donde el usuario está segmentado a través de la máscara del middleware. Para asegurar mayor fiabilidad en los datos del histograma, la imagen de RGB se transforma a HSV (*Hue Saturation Value*) debido a su mayor robustez ante cambios de iluminación, puesto que desacopla la información de la tonalidad de color (*Hue*), de la información de intensidad de gris (*Value*) y de la distancia al eje blanco-negro (*Saturation*) [Gonzalez and Woods, 2001]. El vector resultante del histograma es el vector de características que se utilizará para el posterior reconocimiento del usuario. El vector se compone de dos partes: la primera almacena aquellos puntos con unas condiciones de saturación y brillo bajas, cómo los producidos por colores blancos, negros o grises, los cuales ofrecen poca fiabilidad; mientras que la segunda parte del vector almacena la información de aquellos píxeles donde la información de color es fiable, es decir, el resto de colores.

Una vez se ha identificado un usuario por parte del sistema implementado en la plataforma robótica, el robot modifica su consigna de velocidad en función de la distancia al objetivo. En el caso de servir de guía, esta consigna de velocidad deberá de controlarse de forma compartida con la proviniente del módulo de navegación.

Una vez completado el proceso inicial de detección e identificación del usuario 'target', en las iteraciones posteriores debe de ir comprobándose que existe una navegación conjunta del binomio persona-robot, manteniéndose la distancia objetivo. Para ello, se recogerán los indicadores de posición tridimensionales de todas las personas detectadas en el campo de visión y se aplicarán dos criterios que permitan discriminar la persona 'target' del resto de personas presentes en la escena.

2.2. Criterio a priori

En toda iteración, el primer paso del algoritmo de seguimiento será comprobar si el etiquetado del middleware es coherente en función de la posición actual y la anterior, que se considera conocida y completamente fiable. Para determinar si la posición actual corresponde a nuestro usuario, se ha utilizado como métrica una función gaussiana, con media en la posición anterior y como varianza la distancia que se supone pueda haberse desplazado el usuario en el tiempo transcurrido entre frames consecutivos. Registrando datos de un único usuario en movimiento, y adquiriendo imágenes a 30 fps, se ha determinado que una velocidad máxima de $10km/h$ supone una desviación de unos $30mm$ entre frames consecutivos.

Si el middleware original de OpenNI ha cambiado las etiquetas de identificación de usuarios, el cálculo anterior nos dará un resultado negativo en el seguimiento. La aparición / desaparición de nuevos candidatos a usuario provoca que, cada cierto tiempo, el middleware indexe de nuevo los usuarios detectados. Este suceso obliga a comprobar de nuevo si alguno de los usuarios es el 'target'. Para ello, se solicitará al middleware las coordenadas de todos los usuarios en el marco de la imagen y se aplicará la función gaussiana propuesta a todas ellas. En el caso de que ninguna de las personas en la escena ofrezca la suficiente certeza de que es un único candidato que se puede haber desplazado de la antigua posición a la actual, se deberá proceder de nuevo a su identificación, esta vez mediante otra fuente de información: la cámara RGB.

2.3. Criterio a posteriori

En caso que el seguimiento dinámico falle, se comprobará si el histograma de color almacenado de la persona 'target' coincide con alguno de los histogramas de los usuarios detectados. Para poder extraer la información del histograma de color de los distintos usuarios del frame, se recoge del middleware una imagen como la de profundidad, pero etiquetada con el identificador ('Id') de usuario que ha sido asignado. Realizando el cambio de plano correspondiente a RGB, se toma cada uno de los píxeles que pertenecen al usuario que se desea comparar con el modelo de la persona 'target' y se recopila el número de píxeles para cada una de las semillas con las que se generó el histograma. Para evaluar la similitud entre histogramas se calculará la distancia del coseno para dos vectores:

$$CosSim(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}^T \mathbf{y}}{\|\mathbf{x}\|_2 \|\mathbf{y}\|_2}$$

Como primera opción, se comprobará para el usuario que el criterio a priori (el basado en la dinámica del movimiento) ha establecido como más probable. En el caso que ambos descriptores de color no coincidan lo suficiente, es decir, su similitud por la distancia del coseno no supere un umbral, se

deberá comprobar para el resto de candidatos que se encuentren en el marco de la imagen.

3. Experimentación

Para simplificar la experimentación y centrarla en el objeto de estudio del presente trabajo, se supondrá que el robot circula en un escenario libre de obstáculos. Igualmente, se ha supuesto que el procesador incorporado en el robot es capaz de ejecutar en paralelo las tareas de localización y navegación necesarias, de las que recibe una consigna objetivo de desplazamiento, en caso de servir de guía.

El robot utilizado para el desarrollo de la plataforma es un robot móvil WiFiBot Lab [WiFiBot, 2011]. Un robot no holonómico de cuatro ruedas con rotación diferencial. Dispone de un sistema embebido con un procesador Atom de 1,6GHz basado en el chipset Intel 945 que se comunica via puerto serie con un circuito que controla motores, encoders e infrarrojos mediante un dspic. El funcionamiento del sistema se resume en: recoger los datos del sensor kinect, evaluar la posición o histograma de los candidatos y mandar la consigna de velocidad vía puerto serie, acorde a la posición de usuario validado. La experimentación ha consistido en poner en marcha el aplicativo sobre la plataforma y comprobar que el sistema es capaz de identificar a la persona que queremos seguir y de mantenerse a una distancia de interés respecto la misma.

En la Figura 3, se puede observar la plataforma en funcionamiento con la función seguimiento activada. Se desplaza hasta llegar a la distancia objetivo. Esta se ha definido en $1,80m$ con una zona muerta de $+/-0,1m$.

Figura 3: Imágenes del sistema en funcionamiento

En la Figura 4, se observan las imágenes de profundidad y RGB capturados en un instante de ejecución durante la experimentación desde el punto de vista del robot. En la tercera ventana se observa el usuario seleccionado, segmentado y extraído del resto de la imagen.

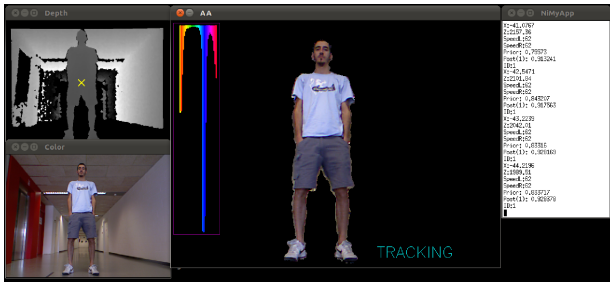


Figura 4: Imágenes que intervienen en un frame

En el Cuadro 1 se muestra la salida del sistema para el frame de la Figura 4. Los campos reflejan, en primer lugar, la posición relativa X, Z en 2D del usuario y la velocidad establecida en cada lado. En segundo lugar, el resultado del criterio a priori y como es inferior al umbral establecido (0,90), el resultado del criterio a posteriori del usuario que el histograma tiene máxima correspondencia. Para finalizar nos aparece el ID que hemos seleccionado como nuestro usuario.

Etiqueta	Valor
X	-44.2196
Z	1989.51
SpeedL	62
SpeedR	62
Prior	0.833717
Post(1)	0.928378
ID	1

Cuadro 1: Console Output

Durante la fase inicial de experimentación sobre el robot se ha observado una reducción de la frecuencia de actualización de los datos provenientes de kinect. Por tanto, la frecuencia de ejecución del algoritmo, que se basó originalmente en 30 fps, pasó a convertirse en 5 fps debido al procesador que monta el robot, lo que ocasiona, si el usuario se desplaza rápido, la pérdida de su seguimiento por parte del middleware original de OpenNI. Este aumento del tiempo de muestreo viene condicionado por dos motivos. En su mayor parte, por la menor capacidad computacional del ordenador embebido que contiene la plataforma robótica. En segundo lugar, porque el framework OpenNI, junto el sistema kinect, no es robusto a movimientos bruscos del sensor, los cuales vienen provocados por el movimiento del robot. Cuando el movimiento del robot, y por tanto del sensor, no es suficientemente constante, el sensor deja de entregar datos al procesador. No obstante, una vez que los datos en el sensor están disponibles, el tiempo de acceso a éstos es realmente bajo y casi no supone retardo en el movimiento del mismo.

Para evitar el segundo de los problemas, el de estabilidad en el sensor, provocados por el movimiento del robot, la captura de imágenes se realizará condicionada a que el robot esté parado o su movimiento se produzca con una aceleración casi nula, es decir velocidad constante. En los casos que este condicionante provoque que la evaluación de frames sea muy discontinua, no se podrá asegurar el seguimiento del usuario mediante las condiciones dinámicas establecidas en la evaluación a priori, por lo que se aumenta la latencia de uso del algoritmo de identificación propuesto. Esta situación también se ve acentuada cuanto más usuarios existen en el campo de visión. Si bien es cierto que se comprueba primero el candidato más probable en cuanto a la posición relativa al sensor, si éste no es el objetivo, entonces se vuelve a evaluar para los restantes candidatos.

Otro problema encontrado durante la fase experimentación de ejecución del algoritmo en la plataforma móvil, es que en determinadas situaciones lumínicas, coincidentes con los cambios de fuentes de luz, el histograma de color de un mismo sujeto sufre considerables variaciones.

Finalmente, también se ha experimentado que la velocidad

máxima del robot disponible no es excesivamente rápida, por lo que para la aplicación propuesta de seguimiento, la persona que hace de guía debe de reducir su velocidad de tránsito, para así adecuarlo a la velocidad máxima del robot.

4. Conclusiones y trabajo futuro

Se ha presentado un algoritmo que permite la identificación y el seguimiento de una persona objetivo por parte de un robot. Para ello, usando kinect como sensor, se han desarrollado dos algoritmos, uno dinámico y otro por histograma de color, que permiten comprobar de forma eficiente el seguimiento. El algoritmo, basado en el middleware de OpenNI, permite evitar la necesidad de identificar al usuario mediante una pose que resulta poco natural. Además, en la función de robot guía, permite que el robot pare cuando el usuario no mantiene la distancia establecida.

El sistema se ha demostrado fiable cuando el sensor se procesa desde un ordenador personal en situación de reposo. Sin embargo, los movimientos bruscos del robot empleado, su reducida velocidad de movimiento y la baja velocidad de procesado de su ordenador de a bordo, provoca que el porcentaje de éxito en el seguimiento se reduzca cuando se halla en condiciones de reales de seguimiento de una persona.

El sistema Kinect, por otra parte, no permite la sincronización de datos vía hardware [ROS, 2011]. El acceso a las dos fuentes de datos (las cámaras RGB y Depth) no siempre se logra realizar en el mismo instante. Ello provoca que la segmentación de la imagen no sea tan precisa como sería deseable y también perjudique el algoritmo aquí expuesto.

De igual forma, cuando el usuario aparece cortado en la escena, decrece la correspondencia de su histograma con el almacenado en el modelo, por lo que el porcentaje de acierto disminuye.

Por último, al ser usado el histograma de color como dato de identificación, también se debe tener en cuenta que si varios usuarios usan ropa muy similar en color, es muy probable que el sistema se confunda y no ejerza una identificación positiva.

Todos estos problemas planteados se traducen en una reducción considerable del acierto en la identificación del usuario. Se observan diferencias de hasta un 20~25 % respecto al modelo registrado en la fase de inicialización, lo que puede ocasionar un hurto de la identidad a seguir, a lo largo de la ejecución.

Una posible solución, planteada como trabajo futuro para solventar los problemas anteriores, sería modificar el algoritmo con el fin de ir adaptando el modelo del usuario objetivo a diferentes condiciones lumínicas. También se podría modificar el algoritmo para adquirir información del histograma de color, tanto cuando el usuario se da la vuelta como cuando el cuerpo no se vea completo en el marco de la imagen.

Además, la actualización podría sólo realizarse cuando el criterio a priori dé una tasa de acierto muy elevada y sólo haya un usuario detectado en la escena. Ésta consistirá en añadir otra muestra del histograma de color de sus píxeles al modelo.

También se podría estudiar la posibilidad de añadir características de la cara en el vector de identificación. No obstante, este proceso requiere una resolución que con el sistema kinect no se obtiene a partir de cierta distancia y tampoco seria de ayuda cuando el usuario esté de espaldas al sensor.

Finalmente, otra posibilidad de mejora del modelo de usuario sería recoger la información del color por partes. Una primera aproximación podría ser dividir el cálculo del histograma por un número establecido de franjas horizontales. Como alternativa se podría considerar obtener diferentes histogramas para las diferentes partes del cuerpo, considerando, por ejemplo, cada una de las extremidades y torso por separado.

Referencias

[Endo *et al.*, 2009] Gen Endo, Masatsugu Iribe, Atsushi Tani, Toshio Takubo, Edwardo F. Fukushima, and Shigeo Hirose. Study on a practical robotic follower to support daily life -development of a mobile robot with 'hyper-tether' for home oxygen therapy patients. volume 23, pages 316 – 323. Fuji Technology Press, 2009.

[Gonzalez and Woods, 2001] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 2001.

[Kinect, 2011] Kinect. http://www.xbox.com/es-es/xbox360/accessories/kinect/home, 2011.

[Kristou *et al.*, 2009] M. Kristou, A. Ohya, and S. Yuta. Panoramic vision and lrf sensor fusion based human identification and tracking for autonomous luggage cart. In *Robot and Human Interactive Communication, 2009. RO-MAN 2009. The 18th IEEE International Symposium on*, volume 27, pages 711 –716, October 2009.

[Laumond, 1993] J.-P. Laumond. Controllability of a multibody mobile robot. *IEEE Transactions on Robotics and Automation*, 9(6):755 –763, December 1993.

[Leyvand *et al.*, 2011] Tommer Leyvand, Casey Meekhof, Yi-Chen Wei, Jian Sun, and Baining Guo. Kinect identity: Technology and experience. *Computer*, 44:94–96, 2011.

[NITE, 2011] PrimeSense NITE. http://www.primesense.com/?p=515, 2011.

[OpenNI, 2011] OpenNI. http://www.openni.org/, 2011.

[ROS, 2011] ROS. http://www.ros.org/wiki/openni_camera, 2011.

[Thrun *et al.*, 1999] Sebastian Thrun, Maren Bennewitz, Wolfram Burgard, Armin B. Cremers, Frank Dellaert, Dieter Fox, Dirk Hähnel, Charles Rosenberg, Nicholas Roy, Jamieson Schulte, and Dirk Schulz. MINERVA: A second-generation museum tour-guide robot. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA'99)*, 1999.

[WiFiBot, 2011] WiFiBot. http://www.wifibot.com, 2011.