Universidad de Buenos Aires
Facultad de Ciencias Exactas y Naturales
Departamento de Computación

Seguimiento de Objetos en Secuencias de Imágenes RGB-D

Tesis presentada para optar al título de
Licenciado en Ciencias de la Computación

Mariano Bianchi

Director: Francisco Roberto Gómez Fernández
Buenos Aires, 2014

SEGUIMIENTO DE OBJETOS EN SECUENCIAS DE IMÁGENES RGB-D

Acá iría el abstract en español (aprox. 200 palabras).

Palabras claves: español, abstract, acá (no menos de 5).

OBJECT TRACKING USING RGB-D IMAGE SEQUENCES

Escribir acá el abstract IN ENGLISH ;) (aprox. 200 palabras).

Keywords: Escribir, ENGLISH, acá (no menos de 5).

AGRADECIMIENTOS

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce sapien ipsum, aliquet eget convallis at, adipiscing non odio. Donec porttitor tincidunt cursus. In tellus dui, varius sed scelerisque faucibus, sagittis non magna. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Mauris et luctus justo. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Mauris sit amet purus massa, sed sodales justo. Mauris id mi sed orci porttitor dictum. Donec vitae mi non leo consectetur tempus vel et sapien. Curabitur enim quam, sollicitudin id iaculis id, congue euismod diam. Sed in eros nec urna lacinia porttitor ut vitae nulla. Ut mattis, erat et laoreet feugiat, lacus urna hendrerit nisi, at tincidunt dui justo at felis. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Ut iaculis euismod magna et consequat. Mauris eu augue in ipsum elementum dictum. Sed accumsan, velit vel vehicula dignissim, nibh tellus consequat metus, vel fringilla neque dolor in dolor. Aliquam ac justo ut lectus iaculis pharetra vitae sed turpis. Aliquam pulvinar lorem vel ipsum auctor et hendrerit nisl molestie. Donec id felis nec ante placerat vehicula. Sed lacus risus, aliquet vel facilisis eu, placerat vitae augue.

Índice general

1. INTRODUCCIÓN

En la actualidad, las posibles aplicaciones de métodos de seguimiento o tracking son muchas y van desde el uso en la industria hasta juegos de consola. Un ejemplo de ello es la fabricación de barcos y autos mediante el uso de robots. Estas tareas se caracterizan por la necesidad de posicionar de manera precisa una herramienta sobre una pieza de trabajo. A través del uso de métodos de tracking se puede conocer la posición y pose de la pieza que se desea utilizar con respecto a la pose de la cámara y de esta forma saber cómo ubicar la herramienta necesaria para trabajar sobre la pieza en cuestión.

Otra área en donde se utiliza tracking de objetos es para la generación de estadísticas durante un partido de fútbol, tanto de jugadores como de un equipo, aunque las posibles aplicaciones en este contexto son mucho más amplias, como por ejemplo análisis de tácticas, verificación de las decisiones del árbitro, resúmenes automáticos de un partido, etc.

Actualmente existen sensores de profundidad que en conjunto con una cámara RGB pueden ser utilizados para detectar y seguir a una o más personas en tiempo real. De esta manera, mediante un sistema que procese las imágenes RGB-D de estos sensores, las personas puedan utilizar su cuerpo y sus movimientos para interactuar naturalmente con un dispositivo.

La utilización de sensores RGB-D se ha popularizado en los últimos años, cobrando un gran interés científico el estudio de aplicaciones y métodos capaces de procesar y entender la información que los mismos proveen.

La información de profundidad que nos provee un sensor RGB-D es un dato fundamental que nos posibilita encontrar la distancia de un objeto al sensor pudiendo recuperar su información 3D (tridimensional) junto a su textura RGB en tiempo real: 30 cuadros por segundo. El video RGB-D que se obtiene provee una gran ayuda al mejoramiento y desarrollo de nuevas técnicas de procesamiento de imágenes y video ya conocidas. En particular, es de interés en esta tesis, el seguimiento de objetos en secuencias de imágenes RGB-D.

Un sistema de seguimiento se puede dividir en tres etapas bien definidas:

1. Entrenamiento
2. Detección
3. Seguimiento cuadro a cuadro

La etapa de entrenamiento consiste en obtener una representación del objeto al cuál se pretende seguir. Para llevarla a cabo se puede utilizar un patrón (template) ya conocido o aprenderlo de imágenes capturadas del mismo objeto. Este template luego se utiliza en la detección para ubicar la representación del objeto dentro de una imagen cualquiera. Una vez conocido el template no se requiere de una nueva ejecución del entrenamiento.

La segunda etapa, la de detección, radica en encontrar dentro de un frame del video al objeto en cuestión utilizando el método de detección deseado, valiéndose de la información registrada en la etapa de entrenamiento. Esta etapa se ejecuta, con el propósito de encontrar en la imagen el objeto a seguir, al comienzo del sistema de seguimiento y cuando el seguimiento cuadro a cuadro falla. Dado que la etapa de detección suele ser la más costosa en términos de desempeño computacional es deseable que se ejecute la menor cantidad de veces posible.

Finalmente, la tercera etapa consiste en seguir cuadro a cuadro el objeto detectado en la etapa anterior. Es decir, teniendo la ubicación del objeto en un cuadro de video se desea identificar la posición del mismo objeto en el siguiente frame. Esta etapa es la más importante ya que es la que se ejecuta en cada frame del video. La eficiencia del método de seguimiento es lo que determinará que todo el sistema de seguimiento se consiga realizar eficientemente. Si la técnica de seguimiento tiene una efectividad baja, es decir, no logra identificar la nueva posición del objeto en el siguiente cuadro, se debe volver a la etapa de detección cuyo desempeño computacional es mayor.

1.1. Objetivos ¿va?

El objetivo principal de esta tesis es la implementación, estudio y evaluación de un sistema de seguimiento de objetos en secuencias de imágenes RGB-D, con las siguientes características:

- Performance Real-time: procesamiento de imágenes mayor a 10 cuadros por segundo
- Seguimiento de objetos tridimensionales con forma conocida previamente y de objetos aprendidos mediante una fase de entrenamiento previa
- Funcionamiento en sensores de profundidad de bajo costo (Kinect, XTion, etc.)

2. DESARROLLO

2.1. Trabajo relacionado

En el artículo [PLW11] se implementan las tres etapas de un sistema de seguimiento nombradas anteriormente. Cada una de estas etapas es abordada de distintas maneras según la literatura actual. La etapa de entrenamiento consiste en obtener una representación tridimensional del objeto al cuál se pretende seguir. En el artículo [DC99] se utiliza un entrenamiento off-line que consiste en obtener un modelo CAD (computer-aided design) del objeto que se desea seguir. Luego, en el artículo [PLW11] se presenta una etapa de entrenamiento novedosa que se realiza de manera on-line, en donde utiliza un marcador conocido para definir las coordenadas de los objetos y calibrar la cámara.

La etapa de detección tiene como objetivo obtener la ubicación del objeto a seguir en un frame dado. En el artículo [PLW11] utilizan el método propuesto en [HLI$^+$10] para detección de objetos en imágenes 2D y lo extienden para estimar la pose 3D. Otros métodos conocidos en la literatura son los propuestos en [Bru09, KRTA13].

La etapa de seguimiento 3D cuadro a cuadro es la más importante y de la que depende el éxito o fracaso de todo el sistema de seguimiento. En el artículo [PLW11] utilizan el algoritmo "Iterative Closest Point" (ICP) propuesto en [Zha94, BM92], refinando el resultado con datos de bordes tomados durante la fase de entrenamiento. El método utilizado por [DC99] se basa en la detección de bordes para realizar el seguimiento frame a frame.

2.2. Base de datos RGB-D

Durante el desarrollo de este trabajo se utilizaron datos de imágenes RGB-D anotados para aplicar los métodos estudiados y tener una referencia para hacer comparaciones y sacar conclusiones sobre su eficacia. Los datos fueron tomados del trabajo [LBRF11] en donde se creó una base de objetos y escenas anotada frame a frame. Esta base cuenta por un lado con varias escenas. Cada una de ellas consta de varios frames RGB con su respectiva información de profundidad. Además, la base provee información frame a frame de qué objetos aparecen y cuál es su ubicación en el plano RGB.

Por otra parte, la base provee imágenes RGB de los objetos anotados frame a frame en las escenas antes mencionadas. Para tomar estas imágenes los objetos fueron posados en una base circular giratoria y manteniendo la cámara en una posición fija se tomaron muestras con cierta regularidad cubriendo toda la circunferencia de cada objeto. Esto se hizo además desde distintas alturas permitiendo apreciar la profundidad del objeto y así obtener una mejor descripción del mismo. Cada una de estas imágenes es acompañada además por una máscara que segmenta al objeto buscado y la información de profundidad (nube de puntos) del objeto segmentado.

Los objetos elegidos para esta base se organizaron de una manera jerárquica tomada de las relaciones hiperónimo/hipónimo de WordNet. Cada objeto pertenece a una clase de objetos y hay varias instancias por cada clase. Por ejemplo, en la categoría "taza" existen varias instancias diferentes, que se corresponden simplemente a distintas tazas ya sea por forma o por color.

Existen distintas escenas que contienen a los objetos mencionados y en cada escena se combinan distintas clases de objetos y distintas instancias de la misma clase. De esta manera la base otorga la posibilidad de generar algoritmos capaces de identificar instancias de objetos particulares o familias de objetos según la clasificación antes mencionada.

2.3. Alignment prerejective

2.4. Iterative Closest Point (ICP)

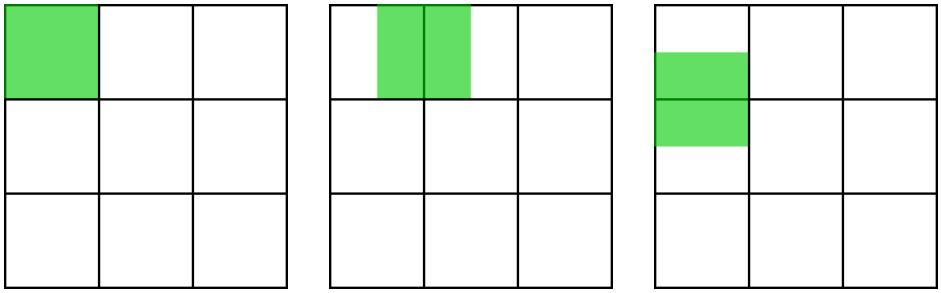
2.5. Esquema de seguimiento

Cosas a escribir:

- cómo separé las etapas en el código y por qué
- cómo se çomunican"

2.6. Método propuesto

Tomando como base las etapas antes mencionadas, proponemos distintos métodos para cada una de ellas. La primera etapa del sistema puede ser prescindible si contamos con el modelo 3D del objeto a seguir y una cámara calibrada. Este es el caso de estudio de esta tesis, ya que, con el propósito de poder evaluar cuantitativamente el seguimiento de objetos en secuencias de imágenes RGB-D, utilizamos la base de datos descripta en la sección 2.2. Para esta primer etapa existen varias posibilidades distintas que van desde utilizar una única nube de puntos hasta generar un modelo completo del objeto 3D alineando todas las nubes de puntos disponibles en la base. La tarea de generar un modelo completo excede el tema de estudio de esta tesis. Además el modelo resultante se utilizara únicamente para la etapa de detección por lo que se optó por el método más simple que es tomar una nube de puntos cualquiera del objeto a seguir como modelo 3D. Lo ideal sería que esta nube de puntos sea lo más completa posible, acercándose así al modelo 3D completo del objeto pero esto no es muy factible ni se puede obtener desde la base que usamos. Otra posibilidad es elegir de todas las nubes de puntos de cada objeto que otorga la base alguna cualquiera al azar. Esto resulta más realista pero sería un problema al momento de analizar los resultados ya que se agregaría una variante azarosa y se deberían correr muchas veces la misma prueba para lograr un análisis más acertado. Por estos motivos se decidió elegir una nube de puntos fija para cada objeto, en



(a) Primer frame de búsqueda *(b)* Segundo frame de búsqueda *(c)* Sexto frame de búsqueda

Fig. 2.1: Se busca en cada cuadrante de la grilla y en los recuadros del mismo tamaño que cubren los bordes de la grilla principal

particular, la primera según orden alfanumérico del nombre del archivo proveniente de la base.

Para la segunda etapa, la de detección, se utilizó el método descripto en la sección 2.3 refinando el resultado con ICP. La elección del mismo se realizó luego de correr varias pruebas que corroboraran la factibilidad del mismo. A través de estas pruebas se pudo observar que el método era robusto para ciertos valores de los parámetros y un objeto en particular pero que al cambiar de objeto los valores de los parámetros debían ser modificados para lograr una detección. También se observó en las pruebas que si la escena donde se buscaba el objeto era lo suficientemente pequeña, la búsqueda volvía a ser robusta no solo para un objeto en particular sino para aquellos elegidos para estas pruebas. Teniendo en cuenta esto se pensó en una variante para la detección que utilice el método elegido. Esta tenía como primer etapa obtener el alto y el ancho del modelo del objeto y multiplicarlos por un cierto valor, llamado **DETECTION_FRAME_SIZE**. Considerando estos valores dividimos la escena en cuadrantes de ese tamaño y corrimos el método de detección en cada cuadrante. Como el objeto a detectar puede haber quedado justo entre dos cuadrantes, la división se hizo de manera tal que estos cuadrantes se solapen entre si, como puede observarse en el gráfico 2.1

mbianchi: Notar que la división no se hizo en el eje de la profundidad ya que las pruebas preeliminares dieron buenos resultados de esta manera y hacer eso implicaba agregarle complejidad algoritmica al método.

o

mbianchi: Notar que esta división solo se hizo en los ejes “x” e “y” y no en el eje “z”

. La detección se corre en cada uno de estos cuadrantes y pueden suceder varias cosas:

- No se encontró el objeto en ningún cuadrante: en este caso el algoritmo indica que el objeto no se encuentra en el frame
- Se encontró el objeto en un cuadrante

- Se encontró el objeto en varios cuadrantes: el algoritmo devuelve la mejor alineación encontrada

Si la detección es positiva, se refina la alineación corriendo ICP entre el modelo del objeto transformado por el método "alignment prerejective" y el cuadrante de la escena donde fue encontrado el mismo. Para tratar de mejorar aún más el resultado y con el objetivo de comenzar el seguimiento en las mejores condiciones posibles, se intentan tomar los puntos del objeto buscado pertenecientes a la escena. Esto se realiza porque se asume que el objeto se va modificando frame a frame, ya sea por movimientos de la cámara o del objeto. Dado que el modelo con el que se cuenta es incompleto, como se explicó en la etapa de entrenamiento, es preferible contar con el objeto de la escena en vez del modelo del objeto. Una de las formas para obtener los puntos del modelo del objeto en la escena es utilizando un k-dtree. Se arma un k-dtree con los puntos provenientes del modelo alineado y se filtran uno a uno los puntos de la escena que se encuentren cerca de al menos un punto del modelo en un cierto radio de distancia. Este valor de radio es uno de los parámetros explorados durante las pruebas, llamado **LEAF_SIZE**. Los puntos que surjan de esta búsqueda son los considerados encontrados en la escena. Para que el algoritmo de búsqueda considere exitosa la detección, la cantidad de puntos filtrados de la escena debe ser mayor o igual al 50 % de los puntos del modelo original. Si todas estas etapas son superadas con éxito, se considera que el objeto fue encontrado y se pasa a la siguiente etapa, la de seguimiento. Si cualquiera de estos pasos fallara, se comienza nuevamente con la etapa de detección en el siguiente frame.

La tercera y última etapa

La detección se realizó utilizando [BKK+13] y corrigiendo con ICP. Cosas a escribir:

- qué sucedió al tratar de detectar en toda la escena
- cómo se hizo para dividir la escena en partes y detectar en cada una

La utilización del algoritmo ICP [Zha94, BM92] para realizar el seguimiento resulta natural e intuitiva. Por ello, es que en esta tesis se estudiará el algoritmo ICP y sus variantes [EBW04, SHT09], con el fin de evaluar cómo sus parámetros afectan cuantitativamente al sistema de seguimiento y la performance computacional del mismo. Asimismo, se evaluará la adaptabilidad del filtro de Kalman [WB95] para seguimiento de objetos 3D en imágenes RGB-D con posibilidad de desempeño en tiempo real. El filtro de Kalman es un filtro muy popular y estudiado extensivamente en la literatura [JU97, WVDM00] debido a su gran desempeño para realizar seguimiento en imágenes 2D. Por lo tanto, su aplicación en seguimiento de objetos 3D resulta de especial interés.

2.7. Elección de parámetros

	2	3	5	7
desk_1 - coffee_mug_5	30.71 %±5.9	47.58 %±4.49	42.42 %±5.1	41.41 %±5.21
desk_1 - cap_4	28.7 %±6	42.51 %±8.44	37.48 %±6.1	34.04 %±6.08
desk_2 - bowl_3	23.71 %±9.41	23.13 %±8.8	16.99 %±7.39	14.16 %±6.28

Tab. 2.1: Los valores corresponden al parámetro DET_FRAME_SIZE, que es en cuanto se multiplica el tamaño del bounding box del objeto para usarlo como frame de detección (ver sección 2.6)

	0.2	0.4	0.6	0.7	0.9
desk_1 - coffee_mug_5	36.99 %±5.41	48.82 %±4.66	47.14 %±5.17	44.69 %±5.42	33.18 %±6.5
desk_1 - cap_4	44.22 %±7.7	31.35 %±4.56	34.49 %±6.44	42.54 %±7.86	32.09 %±10.
desk_2 - bowl_3	23.18 %±8.59	24.39 %±9.42	21.46 %±8.94	16.06 %±6.62	15.27 %±6.8

Tab. 2.2: Los valores corresponden al parámetro DET_SIMILARITY_THRESHOLD utilizado en la detección, que está relacionado con la eliminación temprana de malas poses basándose en las distancias entre el objeto y la escena (ver sección 2.3)

	0.2	0.4	0.6	0.7	0.9
desk_1 - coffee_mug_5	30.11 %±6.22	32.11 %±6.67	31.78 %±4.2	36.82 %±5.24	43.6 %±5.49
desk_1 - cap_4	23.83 %±7.18	38.98 %±7.75	37.7 %±6.48	41.21 %±6.59	25.42 %±5.3
desk_2 - bowl_3	9.01 %±4.08	13.21 %±5.61	24.25 %±8.55	21.55 %±8.66	21 %±8.78

Tab. 2.3: Los valores corresponden al parámetro DET_INLIER_FRACTION utilizado en la detección, que es el porcentaje de puntos del modelo considerados suficientes para aceptar una hipótesis de pose del objeto como válida (ver sección 2.3)

	0.2	0.4	0.5	0.75
desk_1 - coffee_mug_5	36.77 %±4.76	51.07 %±5.08	39.94 %±5.62	47.45 %±6.93
desk_1 - cap_4	37.91 %±5.25	48.76 %±9.09	52.32 %±12.39	50.25 %±14.91
desk_2 - bowl_3	20.51 %±7.87	20.81 %±7.81	25.56 %±9.59	24.86 %±10.13

Tab. 2.4: Los valores corresponden al parámetro FIND_PERC_OBJ_MODEL, que es el porcentaje de puntos del modelo considerados suficientes para determinar que lo que se encontró es el objeto buscado (ver sección 2.6)

3. EXPERIMENTACIÓN

4. DISCUSIÓN

5. CONCLUSIONES

Bibliografía

[BKK+13] A.G. Buch, D. Kraft, J.-K. Kamarainen, H.G. Petersen, and N. Kruger. Pose estimation using local structure-specific shape and appearance context. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 2080–2087, May 2013.

[BM92] P.J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14(2):239–256, 1992.

[Bru09] Roberto Brunelli. *Template matching techniques in computer vision: theory and practice*. Wiley. com, 2009.

[DC99] Tom Drummond and Roberto Cipolla. Real-time tracking of complex structures with on-line camera calibration. In *BMVC*, pages 1–10. Citeseer, 1999.

[EBW04] Raúl San José Estépar, Anders Brun, and Carl-Fredrik Westin. Robust generalized total least squares iterative closest point registration. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2004*, pages 234–241. Springer, 2004.

[HLI+10] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Pascal Fua, and Nassir Navab. Dominant orientation templates for real-time detection of texture-less objects. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2257–2264. IEEE, 2010.

[JU97] Simon J Julier and Jeffrey K Uhlmann. New extension of the kalman filter to nonlinear systems. In *AeroSense'97*, pages 182–193. International Society for Optics and Photonics, 1997.

[KRTA13] S. Korman, D. Reichman, G. Tsur, and S. Avidan. Fast-match: Fast affine template matching. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2331–2338, 2013.

[LBRF11] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1817–1824. IEEE, 2011.

[PLW11] Youngmin Park, Vincent Lepetit, and Woontack Woo. Texture-less object tracking with online training using an rgb-d camera. In *Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on*, pages 121–126. IEEE, 2011.

[SHT09] Aleksandr Segal, Dirk Haehnel, and Sebastian Thrun. Generalized-icp. In *Robotics: Science and Systems*, volume 2, page 4, 2009.

[WB95] Greg Welch and Gary Bishop. An introduction to the kalman filter, 1995.

[WVDM00] Eric A Wan and Rudolph Van Der Merwe. The unscented kalman filter for nonlinear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, pages 153–158. IEEE, 2000.

[Zha94] Zhengyou Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, 1994.