Reconocimiento de objetos en una plataforma robótica móvil

Lorenzo García Tena[1], Humberto Sossa[2], Alejandro Alvarado[1], Osslan Vergara[1], Victor Manuel Hinostroza Zubia[1], Francisco Javier López Benavides[1]

[1]Ing. Industrial y Manufactura – Ing. Eléctrica y Computación de la Universidad Autónoma de Ciudad Juárez.
[2]Centro de Investigación en Computación del Instituto Politécnico Nacional

Resumen

La implementación de los algoritmos de reconocimiento de patrones SURF y SIFT en plataformas robóticas móviles permite el uso de los mismos en una serie de tareas amplias como lo pueden ser el acomodo de libros en una biblioteca de manera automática o la selección y aislamiento de material peligroso. Este trabajo se realizó utilizando la Interfaz de Programación de Aplicaciones (API) de una plataforma robótica móvil Robotino, las librerías de código libre OpenCV y el sensor de visión propio del robot. Ambos casos presentan ventajas y desventajas definidas ampliamente en la parte de desarrollo, así como los retos aún existentes en la investigación que continua en labor. Se busca ampliar los datos obtenidos implementando dichos algoritmos de manera autónoma en plataformas nuevas adquiridas y calibrando empíricamente cada algoritmo independientemente.

Palabras clave: Puntos característicos, descriptores, clasificación, algoritmo, SIFT, SURF, Robotino.

Introducción

En las últimas décadas las tecnologías de información se han convertido en herramientas clave para la realización de tareas automatizadas en la vida cotidiana del humano, así como en los procesos industriales. Entre las diferentes disciplinas que contribuyen a dicho proceso, la visión por computadora y el reconocimiento de patrones han sido ampliamente utilizados para aplicaciones industriales y especialmente para la visión en robots.

El seguimiento de objetos en el campo visual es un tema importante en las tareas multiobjetivo, particularmente en aplicaciones tales como las teleconferencias, la vigilancia y la interfaz hombre-máquina (Tzafestas, 2013). La tarea de reconocimiento de objetos consiste en determinar la posición de un objeto en imágenes de forma continua y fiable en escenas dinámicas y/o que presentan ruido constante (Kang, & Lee, 2002).

El reconocimiento de objetos, puede variar en las características específicas de acuerdo al elemento que es descrito como perceptor, esto es, existen propiedades que ayudan a definir a un objeto como tal utilizando diferentes criterios de clasificación según la percepción de visor. Un niño pequeño es capaz de reconocer un gran variedad de objetos, por ejemplo, puede generalizar la idea de un perro presentándole ejemplos de este animal, al mostrarle una raza específica no vista por el

niño antes, es aún capaz de reconocer el tipo de objeto que se le presenta e identificarlo como un perro. Por otro lado, insectos como las abejas son capaces de llevar a cabo reconocimiento visual para la navegación y encontrar su colmena ayudadas de la identificación de forma en las flores y plantas (Krapp, 2007).

Figura 1. Plataforma *Robotino*

El propósito principal del reconocimiento de objetos por medio de computadora es el tomar la información aparentemente no relevante para el ojo humano común que se presenta en imágenes de diferentes tipos y asociarla de alguna manera con un concepto dado.

Un sistema de reconocimiento de patrones completo consiste en:

- Un sensor que toma las observaciones a clasificar por medio de imagen.
- Un sistema de extracción de características que transforma la información observada en valores numéricos o simbólicos.
- Un sistema de clasificación o descripción que, basado en las características extraídas, clasifica la medición.

Una necesidad recurrente dentro de la industria es automatizar el proceso *pick-and-place* de tomar objetos, realizar algunas tareas, para después, colocar el objeto en una ubicación diferente (Bozma, & Kalalıoğlu, 2012). La mayoría de los elementos de *pick-and-place* están básicamente compuestos de sistemas como actuadores y sensores (Harada, Tsuji, Nagata, Yamanobe, & Onda, 2014). Los sensores están a cargo de la conducción de los actuadores del robot a la ubicación del objeto, para luego posiblemente ir a la orientación del siguiente objeto a ser tomado, lo anterior se relaciona como restricción directa del número de grados de libertad que presenta el robot.

El acopio de objetos puede ser muy complicado si el escenario no está bien estructurado y limitado, o en caso que las habilidades y funcionalidades del robot no sean lo suficientemente avanzadas. La automatización de la selección del objeto mediante el uso de cámaras es generalmente requerida para detectar y localizar objetos en la escena. Dichas tareas son cruciales para otras aplicaciones de visión por computadora, como lo son la recuperación de imágenes y vídeo, o la navegación autónoma de robots (Tsai & Song, 2009).

En el presente se describe la implementación de dos algoritmos para reconocimiento de patrones por medio de detectores de características en una plataforma robótica móvil presentándole como parte de la investigación diferentes

tipos de objetos, un acomodo aleatorio de ellos, así como un escenario no establecido parta la detección de los mismos.

El primer algoritmo de reconocimiento propuesto es el detector de características SIFT (*Scale-Invariant Feature Transform*), que se centra en buscar puntos característicos que cumplen criterios espacio-escalares. Los descriptores se calculan a través de la orientación de los gradientes de cada punto. El segundo es SURF (*Speeded Up Robust Features*), uno de los algoritmos más utilizados para la extracción de puntos de interés en el reconocimiento de imágenes. La extracción de los puntos la realiza detectando en primer lugar los posibles puntos de interés y su localización dentro de la imagen.

Trabajos Relacionados

En 2009 Du, Su, & Cai presentan un trabajo relacionado a la extracción de características usando SURF para reconocimiento facial, donde se destaca que SURF es un algoritmo de escala y rotación en el plano detector de invariantes y descriptor, el cual da un rendimiento comparable o incluso mejor con SIFT. Lo anterior debido a que SURF tiene sólo 64 dimensiones en general y un sistema de indexación se construye mediante el uso de la señal de la laplaciana, SURF es mucho más rápido que el SIFT 128-dimensional en el paso correspondiente. Por lo tanto en base a las ventajas mencionadas en dicho trabajo respecto a SURF, proponen explotar las características de SURF en el reconocimiento de rostros en dicho trabajo.

Las ventajas mencionadas en la investigación fueron reflejadas en el trabajo presente, sin aún tener valores suficientes para hacer un análisis estadístico, sin embargo, el funcionamiento mostrado por ambas implementaciones refleja las mismas tendencias en comparación de SURF y SIFT.

Por otra parte, en 2010 Valgren & Lilienthal presentan un trabajo donde abordan el problema del aire libre, la localización topológica por la apariencia, en especial durante largos períodos de tiempo en que los cambios estacionales alteran el aspecto del medio ambiente. Se investiga un método sencillo que se basa en características de la imagen de la zona de comparar pares de una sola imagen. En primer lugar, plantean los autores, se buscó los algoritmos dominantes que cuentan con características de manejo de imagen, SIFT o la más reciente SURF, concluyen son de lo más adecuado para esta tarea. Después, ponen a punto su algoritmo de localización en términos de precisión, y también introducen la restricción epipolar para mejorar aún más el resultado.

El algoritmo de localización final se aplica en múltiples conjuntos de datos, cada uno compuesto de un gran número de imágenes panorámicas, que fueron adquiridos durante un período de nueve meses con grandes cambios estacionales. La tasa de localización final en el juego de una sola imagen, con cambios de temporada es entre el 80% y el 95%.

Los resultados de sus experimentos iniciales mostraron que SURF, o más bien la versión "vertical" de SURF denotado U-

SURF, tuvo el mejor rendimiento. En general, con el algoritmo de U-SURF encontraron puntos clave más relevantes (es decir, puntos significativos que generan similitudes o emparejamientos válidos) y era mucho más rápido que SIFT. Tal vez esto en su caso no es sorprendente teniendo en cuenta la naturaleza aproximada del algoritmo SURF, junto al hecho que U-SURF deja fuera invariancia rotacional y también utiliza un descriptor más corto que SIFT y SURF-128.

SIFT (Scale Invariant Feature Transform).

Ha sido demostrado que la idea de utilizar características locales es el método más eficaz de reconocimiento visual y localización de robots.

El primero fue Christoph von der Malsburg usando filtros de Gabor orientados a escalas diferentes en el mismo gráfico. Después, fue David Lowe el que mezcló estas características de la escala, con su escala SIFT, y ha resultado tener resultados bastante notorios en el ámbito de reconocimiento.

El término SIFT proviene de Scale-Invariant Feature Transform. Es decir, es una transformación de la información que proporciona una imagen en coordenadas invariantes a la escala en el ámbito local (Chen & Hsieh, 2015). A partir de las características locales, se busca conseguir invariancia a la escala, orientación, parcialmente a cambios de iluminación, etc. También se puede utilizar para buscar correspondencias entre diferentes puntos de vista de una misma escena. Estas características locales se almacenan en los descriptores.

El algoritmo cuenta con varias características principales, según la literatura actual (Liu, Liu & Wang, 2015):

1. **Construcción del espacio de escalas:** La representación espacio-escala es un tipo especial de representación multi-escala que incluye un parámetro continuo de escala y preserva el mismo muestreo espacial para todas las escalas. Así, la representación espacio-escala de una señal es empotrar la señal original en una familia de señales de un sólo parámetro construidas mediante la convolución con señales gaussianas de varianza creciente.

2. **Detección de máximos y mínimos espacio-escala}:** El siguiente paso es la búsqueda de puntos en la imagen que puedan ser *puntos clave*. Se realiza usando diferencias de funciones gaussianas para hallar puntos interesantes que sean invariantes a la escala y a la orientación.

3. **Localización de los *puntos clave*:** De los puntos obtenidos en el apartado anterior se determinan la localización y la escala de los mismos, de los cuales se seleccionan los *puntos clave* basándose en la medida de la estabilidad de los mismos.

4. **Asignación de la orientación:** A cada localización del *punto clave* se le asigna una o más orientaciones, basado en las orientaciones de los gradientes locales de la imagen.

5. **Descriptores de los *puntos clave*:** Los gradientes locales se miden y se transforman en una representación que permite importantes niveles de la distorsión de la forma local y el cambio en la iluminación.

6. **Cálculo de correspondencias:** Al ya tener un descriptor, que es un conjunto de elementos que tienen las principales orientaciones de un punto clave, se deberá determinar si en dos imágenes existen correspondencias, es decir similitud, para lo cual se usa la diferencia euclideana.

SURF (Speed Up Robust Feature).

SURF es otro detector de variables locales, y fue presentado por primera vez por Herbet Bay en el 2006 y se inspira en el descriptor SIFT, pero presentando ciertas mejoras, como son:

- Velocidad de cálculo considerablemente superior sin ocasionar pérdida del rendimiento.
- Mayor robustez ante posibles transformaciones de la imagen.

Estas mejoras se consiguen mediante la reducción de la dimensionalidad y complejidad en el cálculo de los vectores de características de los puntos de interés obtenidos, mientras continúan siendo suficientemente característicos e igualmente repetitivos.

SURF es uno de los algoritmos más utilizados para extracción de puntos de interés en el reconocimiento de imágenes (Valgren & Lilienthal, 2010). La extracción de los puntos la realiza detectando en primer lugar los posibles puntos de interés y su localización dentro de la imagen.

Es mucho más rápido que el método SIFT, ya que los *puntos clave* contienen muchos menos descriptores debido a que la mayor cantidad de los descriptores son cero. Este descriptor se puede considerar una mejora debido a que la modificaciones que supondría en el código no serían excesivas, ya que el descriptor SURF utiliza la gran mayoría de las funciones que utiliza el descriptor SIFT (Huang, Chen, Shen, & He, 2015).

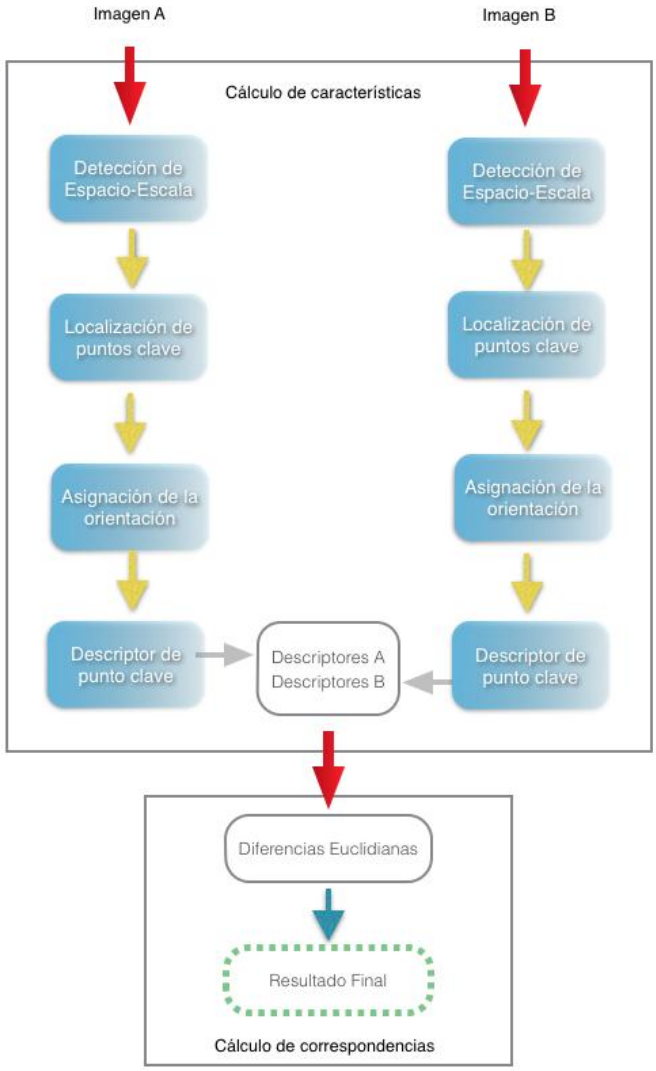

Figura 2. Diagrama de flujo de algoritmo *SURF*

El descriptor SURF hace uso de la matriz Hessiana, más concretamente, del valor del determinante de la matriz, para la localización y la escala de los puntos. El motivo para la utilización de la matriz Hessiana es respaldado por su rendimiento en cuanto a la velocidad de cálculo y a la precisión.

Lo realmente novedoso del detector incluido en el descriptor SURF respecto de otros detectores es que no utiliza diferentes medidas para el cálculo de la posición y la escala de los puntos de interés individualmente, sino que utiliza el valor del determinante de la matriz Hessiana en ambos casos (Miao, Wang, Shi, Lin, & Ruan, 2011).

Materiales y Métodos

Las herramientas para la creación de nuevas funcionalidades con las que trabaja el API del robot son un creador de bloques de función, el uso de un sistema de creación de proyectos *CMake* y el ambiente de desarrollo de software de *Microsoft Visual Studio*. Dichas herramientas son elementos no únicos para desarrollo de bloques a usar dentro del software del fabricante de la plataforma robótica, pueden ser usados otros elementos. Para el presente se usaron las descritas a continuación.

Function Block Manager

Para la creación de un nuevo bloque de función con fines de uso dentro de Robotino® View 2 se requiere crear un proyecto de Visual C++. Dicho software incluye la herramienta "Function Block Manager", la cual permite designar diferentes especificaciones al bloque tales como nombre del desarrollador, nombre del bloque, imagen de icono, entradas y salidas entre otros parámetros y a su vez, permite habilitar el bloque para su uso dentro de la interfaz gráfica una vez que el mismo haya sido generado, véase Figura 3 a continuación.

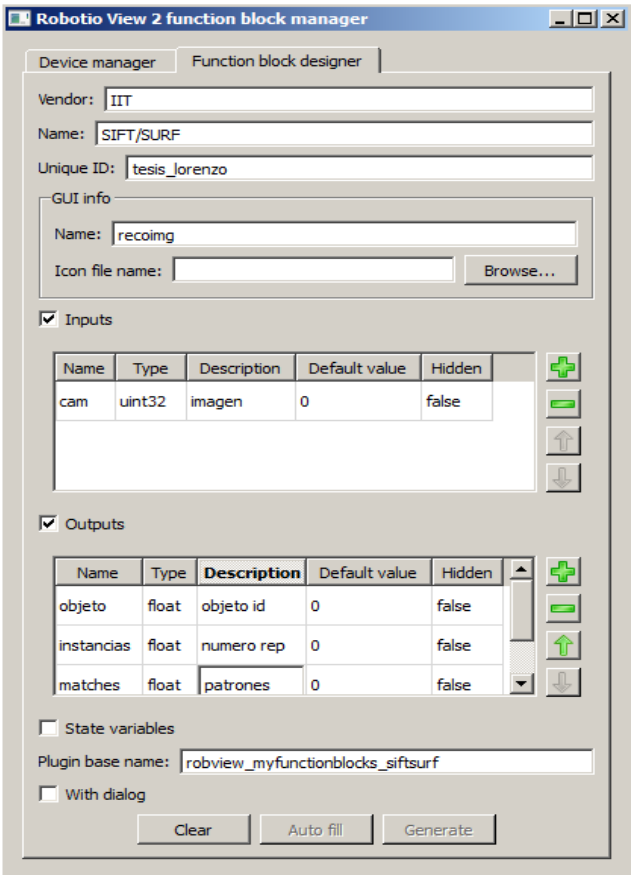

Figura 3. Herramienta *Function Block Manager*

La herramienta “Function Block Manager” genera el nuevo bloque de función y un subdirectorio en una carpeta específica del sistema el cual contiene un archivo de descripción XML, archivos de texto *CMake* y archivos raíz en lenguaje C++ sin función alguna.

CMake

Una vez generado el archivo de texto “CMakeLists” se emplea la interfaz gráfica de *CMake*, el cual es un sistema de código libre de creación, prueba y empaquetado de proyectos para múltiples plataformas de software. Ver Figura 4.

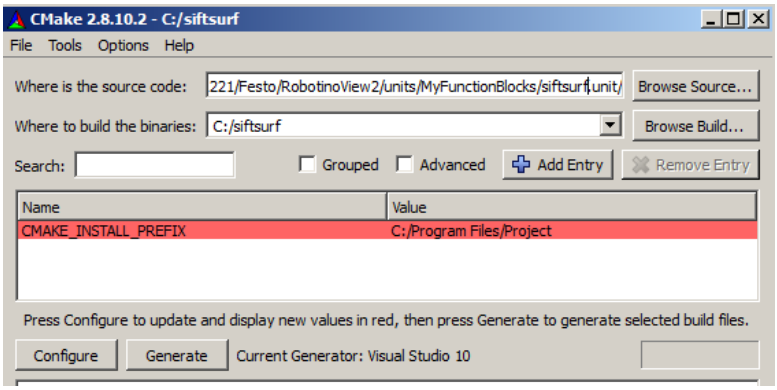

Figura 4. Herramienta *CMake*

Microsoft Visual C++

Visual C++ es un ambiente de desarrollo de aplicaciones que permite la implementación de código en lenguaje C++ para crear diferentes tipos de aplicaciones como librerías de enlace dinámico o ejecutables.

Resultados

Como base de datos se toman una serie de lomos de libros para ser reconocidos por los algoritmos. El las imágenes a continuación se aprecian los diferentes tamaños y patrones de acuerdo al libro en cuestión.

Las imágenes se presentan en blanco y negro debido a que es en escala de grises como se almacenan para ser comparadas y eventualmente recuperadas dentro de una imagen posterior presentada por el sensor del robot.

Figura 5. Base de datos de prueba.

SIFT

La detección de máximos y mínimos en espacio-escala es la etapa donde los puntos de interés, que se llaman puntos clave (punto clave) en el marco de SIFT, se detectan. Para ello, la imagen se procesa con filtros gaussianos a diferentes escalas, y luego se calcula la diferencia de los sucesivos puntos Gaussianos que se han encontrado en la imagen.

Los máximos y mínimos de la función proporcionan las características más estables, con lo que estos serán nuestros puntos clave.

Luego, el *espacio-escala* de detección da como resultado demasiados candidatos punto clave, algunos de los cuales son inestables. El siguiente paso en el algoritmo es realizar un ajuste detallado de los datos más cercanos para la localización exacta, la escala y proporción de curvaturas principales.

Esta información permite poder rechazar puntos que tienen bajo contraste (y por tanto son sensibles al ruido) o están mal localizados.

La asignación de una orientación a los *puntos clave* es muy importante, ya que si se consigue una orientación coherente basada en las propiedades locales de la imagen, el descriptor puede ser representado en relación de dicha orientación y por lo tanto ser invariante a la rotación.

Este enfoque contrasta con la de otros descriptores invariantes a la orientación, que buscan propiedades de las imágenes basadas en medidas invariantes a la rotación. La desventaja de este enfoque es que limita el número de descriptores que se pueden usar y rechaza mucha información de la imagen.

Una vez que ya se tienen todos los *puntos clave* de la imagen, se tiene que segmentar la vecindad del *punto clave* en

regiones de píxeles. Una vez que ya se ha dividido la vecindad del *punto clave*, se genera un histograma de orientación de gradiente para cada región. Para ello, se utiliza una ponderación gaussiana con un ancho sigma.

Esta construcción puede representar problemas, ya que en el caso de un pequeño desplazamiento espacial, la contribución de un pixel puede pasar de una casilla a otra, lo que provoca cambios repentinos del descriptor. Este desplazamiento también puede deberse al hecho de una pequeña rotación.

Las imágenes en la Figura 6 muestran resultados del uso de extractores y descriptores SIFT. Se llevaron a cabo acomodos aleatorios de los libros representados en la base de datos, obteniendo siempre un reconocimiento similar sin importar el acomodo de los mismos.

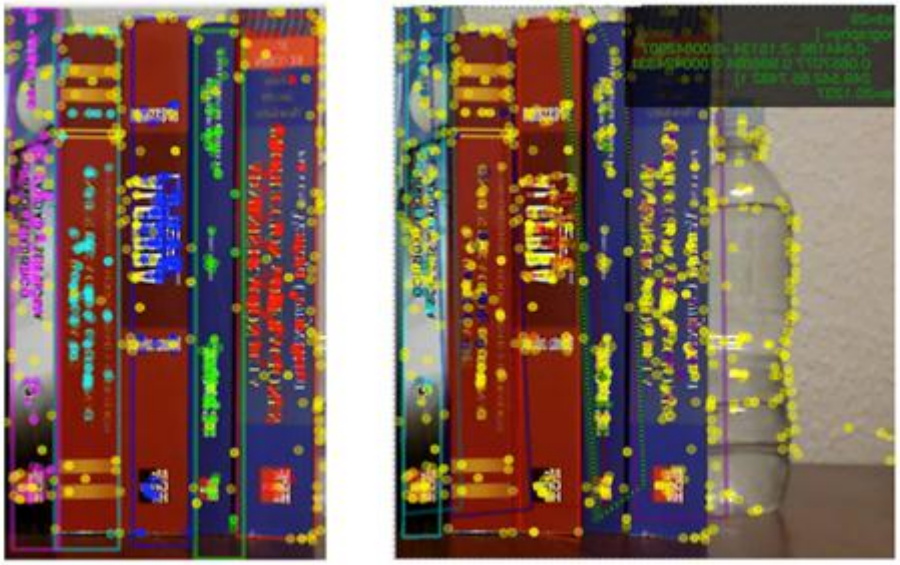

Figura 6. Resultados con detector y descriptor *SIFT*

SURF

El espacio escala para el descriptor SURF, al igual que en el caso del descriptor SIFT, está divido en octavas. Sin embargo, en el descriptor SURF, las octavas están compuestas por un número fijo de imágenes como resultado de la convolución de la misma imagen original con una serie de filtros cada vez más grande (Sykora, Kamencay, & Hudec, 2014). El incremento o paso de los filtros dentro de una misma octava es el doble respecto del paso de la octava anterior, al mismo tiempo que el primero de los filtros de cada octava es el segundo de la octava predecesora.

Después, para calcular la localización de todos los puntos de interés en todas las escalas, se procede mediante la eliminación de los puntos que no cumplan la condición de máximo en un vecindario de la ventana asignada. De esta manera, el máximo determinante de la matriz Hessiana

es interpolado en la escala y posición de la imagen.

La siguiente etapa en la creación del descriptor corresponde a la asignación de la orientación de cada uno de los puntos de interés obtenidos en la etapa anterior. Es en esta etapa donde se otorga al descriptor de cada punto la invariancia ante la rotación mediante la orientación del mismo.

Para los descriptores SURF se construye como primer paso una región cuadrada de tamaño dado alrededor del punto de interés y orientada en relación a la orientación calculada en la etapa anterior. Esta región es a su vez dividida en sub-regiones dentro de cada una de las cuales se calculan las respuestas de Haar de puntos con una separación de muestreo en ambas direcciones.

Por simplicidad, se consideran las respuestas de Haar en las direcciones horizontal y vertical respectivamente relativas a la orientación del punto de interés.

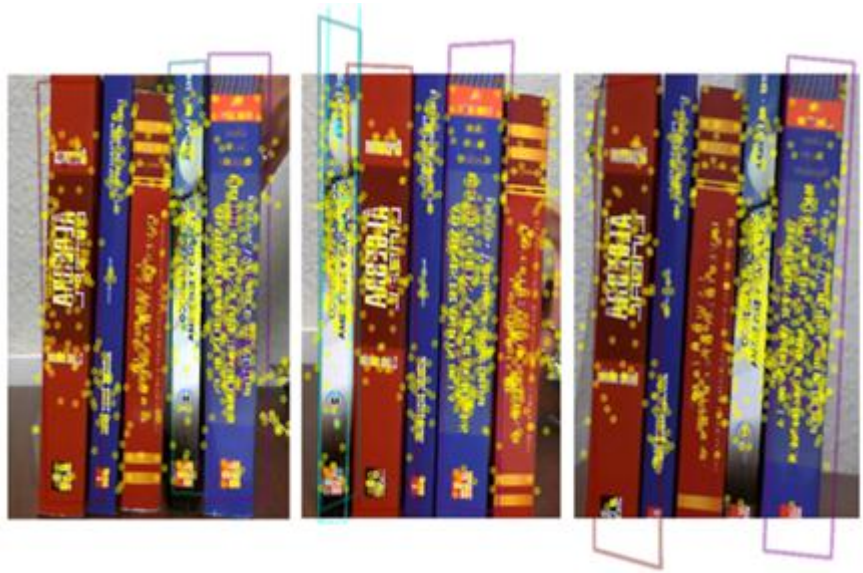

Figura 7. Resultados con detector y descriptor SURF

La aplicación de ambos algoritmos en la plataforma fue ejecutada desde un esquema maestro-esclavo, tal que el procesamiento del módulo de reconocimiento de objetos se lleva a cabo por la computadora que manda las instrucciones a ejecutar por el robot, no por el robot. Lo anterior debido al costo computacional que representa ejecutar las tareas de reconocimiento de patrones.

Conclusiones

En el caso particular de SIFT la aplicación de prueba resulto tener un reconocimiento con menor éxito que los presentados con SURF. Sin embargo en las pruebas también se mostraron resultados combinatorios, donde, la base de datos se tomaba con ambos algoritmos y la recuperación de patrones se veía beneficiada en rapidez. Esta tendencia solo fue comprobada tomando el uso de SURF para el análisis de la imagen entregada por el sensor de visión de la plataforma.

El propósito principal del trabajo fue cumplido ya que ambos algoritmos son capaces de ejecutar las tareas de recuperación, cada uno con limitaciones. Se pretende continuar con el trabajo durante verano y llegar a obtener dos puntos clave:

- Autonomía total de la plataforma robótica.
- Ajuste de algoritmos para mejorar resultados.

Para el primer punto, se debe integrar por completo el procesamiento dentro del robot. Para esto se está trabajando ya con las nuevas plataformas que ha adquirido la Universidad. Hasta ahora se han logrado hacer pruebas utilizando las librerías de *OpenCV* con resultados favorables.

Se espera adquirir la nueva API de dichas plataformas, ya que debido al firmware la versión de las plataformas con las que se realizó el presente son incompatibles para desarrollo de nuevos bloque de función.

El segundo punto involucra la obtención de tiempos y la delimitación de factores de ruido en la recuperación de patrones. Una vez tomados datos suficientes, se debe comprobar mediante un estudio estadístico que los resultados propuestos favorezcan el funcionamiento general del bloque de reconocimiento de patrones.

Uno de los puntos claves que pueden afectar el tiempo en que se logren correr loa algoritmos de manera totalmente independiente en la plataforma recientemente adquirida, es la limitante de no poder trabajar con varias *CF Cards* y por tanto, diferentes versiones de OS al mismo tiempo.

Dichas plataformas tienen una capacidad superior de procesamiento, pero no permiten llegar a la raíz de manera tan directa como las versiones anteriores.

Se sabe que la versión actual de *OpenCV* se puede ejecutar sin ningún problema, pero las pruebas se han ejecutado hasta ahora siguiendo pasos considerando una versión "lite" de dichas librerías.

Actualmente la universidad cuenta con 4 Robotinos, los cuales no son integrados en ningún curso impartido a nivel licenciatura. Se busca incorporar los trabajos realizados en tesis hasta el día de hoy, como prácticas regulares para alguna

de las clases del programa de mecatrónica en un futuro.

Referencias

Chen, C. C., & Hsieh, S. L. (2015). Using binarization and hashing for efficient SIFT matching. Journal of Visual Communication and Image Representation, 30, 86-93.

Du, G., Su, F., & Cai, A. (2009, October). Face recognition using SURF features. In Sixth International Symposium on Multispectral Image Processing and Pattern Recognition (pp. 749628-749628). International Society for Optics and Photonics.

Harada, K., Tsuji, T., Nagata, K., Yamanobe, N., & Onda, H. (2014). Validating an object placement planner for robotic pick-and-place tasks. Robotics and Autonomous Systems, 62(10), 1463-1477.

Huang, L., Chen, C., Shen, H., & He, B. (2015). Adaptive registration algorithm of color images based on SURF. Measurement, 66, 118-124.

Kang, S., & Lee, S. W. (2002). Real-time tracking of multiple objects in space-variant vision based on magnocellular visual pathway. Pattern recognition, 35(10), 2031-2040.

Krapp, H. G. (2007). Polarization vision: how insects find their way by watching the sky. Current biology, 17(14), R557-R560.

Liu, Y., Liu, S., & Wang, Z. (2015). Multi-focus image fusion with dense SIFT. Information Fusion, 23, 139-155.

Miao, Q., Wang, G., Shi, C., Lin, X., & Ruan, Z. (2011). A new framework for on-line object tracking based on SURF. Pattern Recognition Letters, 32(13), 1564-1571.

Sykora, P., Kamencay, P., & Hudec, R. (2014). Comparison of SIFT and SURF Methods for Use on Hand Gesture Recognition based on Depth Map. AASRI Procedia, 9, 19-24.

Tsai, C. Y., & Song, K. T. (2009). Dynamic visual tracking control of a mobile robot with image noise and occlusion robustness. Image and Vision Computing, 27(8), 1007-1022.

Tzafestas, S. G. (2013). Introduction to mobile robot control. Elsevier.

Valgren, C., & Lilienthal, A. J. (2010). SIFT, SURF & seasons: Appearance-based long-term localization in outdoor environments. Robotics and Autonomous Systems, 58(2), 149-156.