2016

Reconocimiento y seguimiento de objetos con Matlab

TÉCNICAS AVANZADAS DE VISION POR COMPUTADOR

CARLOS MORILLO LOZANO - M15329

Contenido

1. Introducción

El trabajo consiste en el desarrollo de un algoritmo capaz de reconocer tres tipos de objetos previamente definidos en una escena de video y realizar un seguimiento de los mismos durante su tiempo en escena.

El lenguaje sobre el que se ha implementado el algoritmo es Matlab. Se ha elegido este entorno de trabajo frente a otros como OpenCV por la facilidad que presenta la herramienta dado su carácter de lenguaje interpretado. Si se buscase una mayor eficiencia computacional se habría optado por C++, sin embargo, para este caso será suficiente lo que puede entregar Matlab.

2. Acondicionamiento de la imagen

2.1. Captación de la imagen

La imagen se obtiene de un video previamente obtenido con las características iniciales estándares. Una vez introducido en Matlab, se pueden ver sus características. Algunas más relevantes son la calidad de 720p y 30FPS y el formato de video 'RGB24'.

```
>> vidObj = cargarVideo

vidObj =

  VideoReader with properties:

    General Properties:
            Name: 'desarrollo.mp4'
            Path: 'C:\Users\Carlos\Google Drive\MASTER\MASTER\Semestre 2\...'
        Duration: 91.6810
     CurrentTime: 0
             Tag: ''
        UserData: []

    Video Properties:
           Width: 1280
          Height: 720
       FrameRate: 30
     BitsPerPixel: 24
      VideoFormat: 'RGB24'
```

A continuación, será necesario comenzar a hacer una lectura de todos los frames que contiene el video para su posterior análisis. Cada frame será analizado hasta que el video finalice.

2.2. Segmentación de la escena

Antes de empezar a reconocer los objetos de la escena, es necesario analizarla para saber si hay alguno en ella. Para ello, se debe tratar de eliminar todos los elementos que no sean objeto tales como el fondo o el ruido.

En este caso las opciones son variadas, en la medida de lo posible debe optarse por una solución genérica para poder tener un sistema de reconocimiento adaptable a diferentes entornos y al ruido. En caso de que no haya solución única, será necesario hacer un código adaptable a cada situación con el fin de optimizar la localización de cada objeto.

Los objetos en cuestión son los siguientes:

Ilustración 1: Plantillas de los objetos

En este caso vemos características comunes entre los dos primeros, pero en el caso del último elemento esa relación desaparece. Es necesario realizar algunos test en distintos canales de la imagen, con el fin de obtener ese punto en común que evite gastar recursos innecesarios.

Segmentación A

Analizando la ilustración 2, se puede comprobar que los canales que mejor separan el objeto del entorno son el canal azul del RGB y el H del HSV; y los que mejor aíslan al objeto y al entorno del ruido (pero presentan menor contraste) son las capas Cb y Cr del canal YCbCr.

Tras estudiar los canales mostrados en todos los frames del video se eliminaron como opciones válidas la capa H por el ruido elevado que introducía ante brillos y la capa Cr. El hecho de que el teléfono destaque en el espectro de los azules es la característica común que se buscaba para poder realizar un algoritmo de segmentación genérico para los tres objetos. Con el fin de eliminar el ruido que presenta el canal azul, se multiplica la capa azul por la capa Cb ecualizada, combinando así las ventajas de cada una.

De esta forma se obtiene un algoritmo potenciador de azul bien aislado de reflejos y del entorno. Únicamente es sensible a reflejos en superficies de color negro, que son mínimos por su índice de reflexión bajo.

Segmentación B

Gracias al potenciador del color azul es posible detectar los objetos cuando se encuentran en un folio blanco, dado que la componente azul del color blanco es de 1 sobre 1. El problema que presenta esto es que el objeto realmente detectado es el folio completo, sin conocer la posición exacta en su interior. Por este motivo se ha desarrollado un algoritmo complementario que se ejecutará cuando el valor de blancos en el histograma supere ciertos valores.

En este caso (ver ilustración 3) se ha eliminado el color blanco del folio restando las capas Cb e Y del canal YCbCr. Tras ecualizar la imagen resultante y binarizarla, los objetos quedarán perfectamente definidos en el interior del folio y aislados del fondo (también detectado como objeto). Para eliminar el fondo se ha hecho uso de la función imclearborder de Matlab.

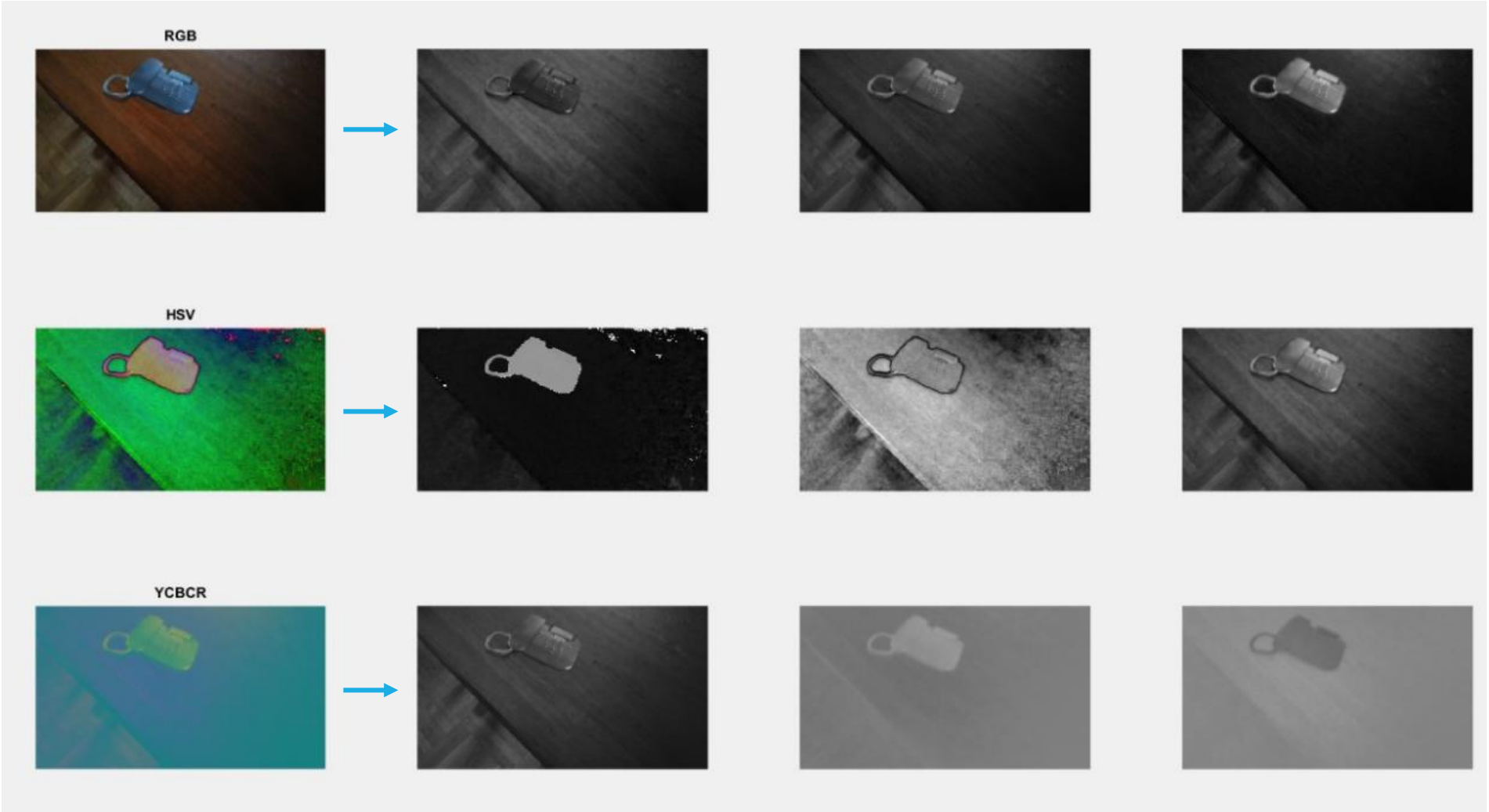

Ilustración 2: Canales de color de la imagen y sus 3 capas ordenadas por columnas

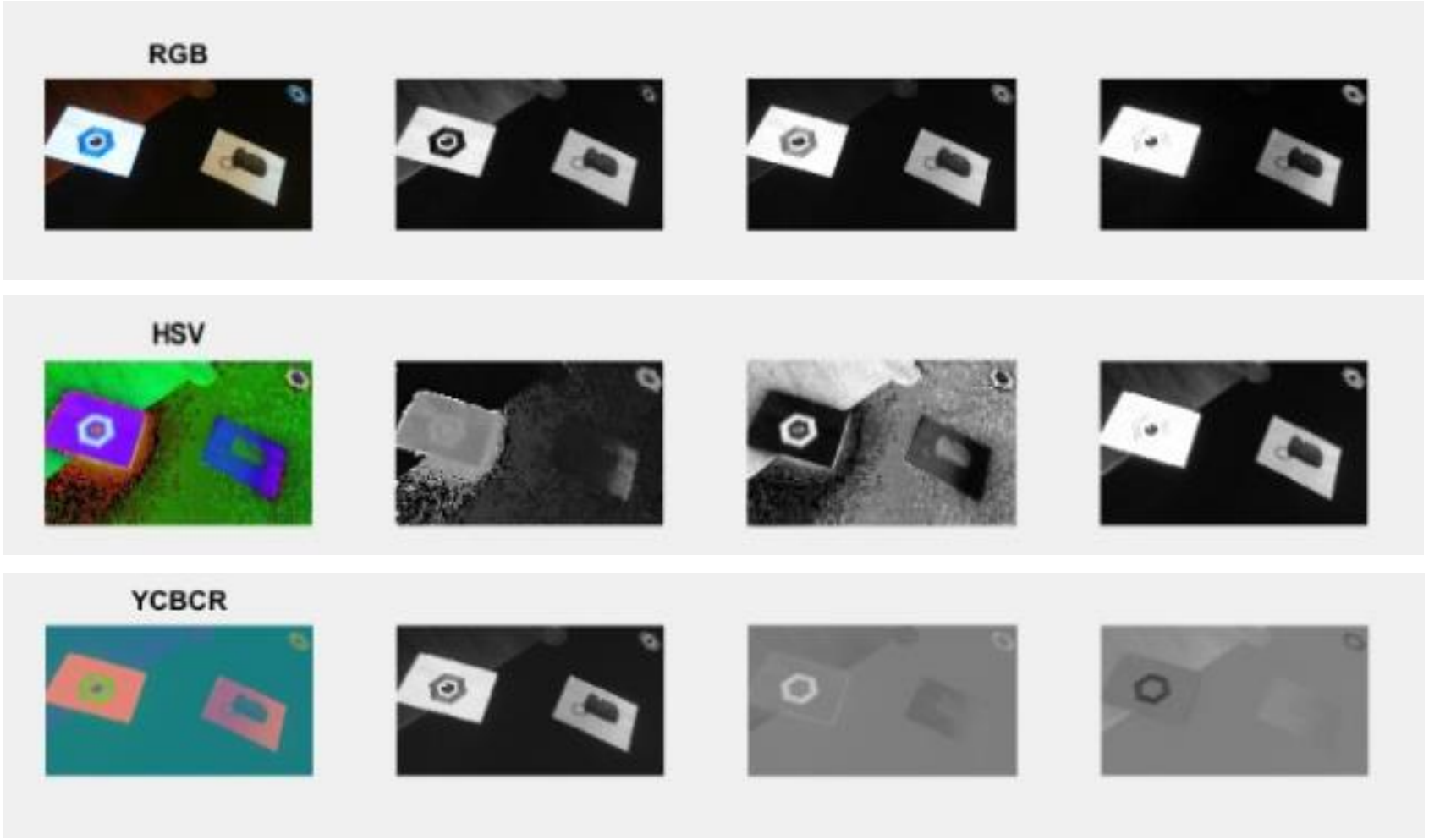

Ilustración 3: Canales de color de la imagen con folios

Resultado de la segmentación

Finalmente, el resultado queda de la siguiente forma en los dos casos respectivamente:

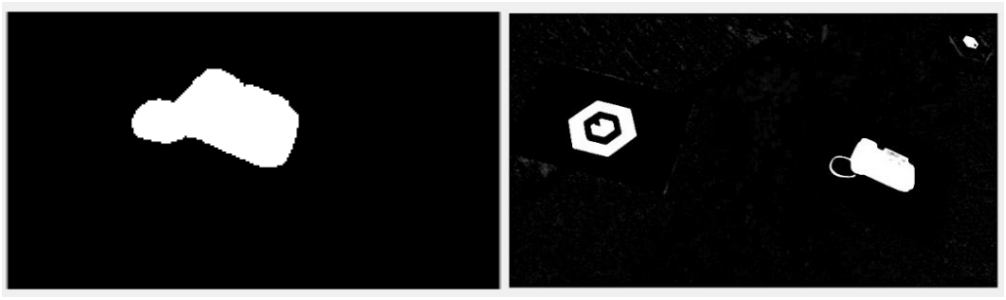

Ilustración 4: (Derecha) Segmentación sin folio. (Izquierda) Segmentación con folio

2.3. Estudio del objeto

Una vez se tiene el resultado de las imágenes anteriores, se pueden contabilizar los objetos y extraer propiedades mediante la función de Matlab regionprops. Además, se añadirán algunas características a cada objeto que serán útiles en su procesado. Las características son las siguientes:

```
>> lista_objetos

lista_objetos =

            Area: 17116
        Centroid: [293.7476 223.1198]
     BoundingBox: [174.5000 156.5000 217 125]
   EquivDiameter: 147.6237
        Etiqueta: ' '
            Util: 1
     Reemplazado: 0
       Precision: -21
        Tracking: [36x2 double]
```

La función de regionprops dará la información de:

- Area: área que ocupan los pixeles del objeto.
- Centroid: localización del centro del objeto.
- BoundingBox: region rectancular equivalente que ocupa el objeto en la escena.
- EquivDiameter: diámetro de la circunferencia circunscrita en el objeto.

Además, se han añadido otras características al objeto:

- Etiqueta: es el nombre que recibe el objeto, se asignará cuando se reconozca y se identifique.
- Util: parámetro que determina si el objeto (que se encuentra en una lista de objetos) sigue siendo necesario. En caso de que este valor sea ‘0’, será eliminado para liberar memoria.
- Reemplazado: indica si el objeto ha sido copiado a otro objeto equivalente.
- Precision: es un cuantificador de la calidad de la precisión del objeto.
- Tracking: es un vector que almacena los centroides del objeto en todo momento.

3. Reconocimiento del objeto

Una vez determinados y aislados todos los objetos de la escena, es necesario realizar su correcta etiquetación. Hay diversos métodos basados en la extracción de características como son SURF, HOG, BRISK, SIFT… en este caso se ha optado por SIFT dada su robustez y a la muy mala precisión que presentaban el resto de algoritmos más sencillos. Los métodos de machine learning también han sido probados, sin embargo, el entrenamiento debe ser muy riguroso y la distinción de los dos teléfonos (dado su enorme parecido), es prácticamente inviable por este método.

Para ahorrar tiempo de procesado de la imagen, dado que SIFT es bastante costoso, se minimizará las veces que se llama a dicha función. Para ello:

- A cada objeto se le asignará una etiqueta con el objeto comparado y se hará el seguimiento del objeto mediante segmentación y tracking.
- Las plantillas se calcularán previamente y tan solo será necesario cargar los datos una vez en el workspace de Matlab.

La función SIFT (Scale Invariant Feature Transform) detectará los puntos característicos de la imagen de entrada que serán invariantes ante cambios de escala, traslación y rotación. Comparando dichos puntos con la función Match con las plantillas, se darán una serie de correspondencias. La plantilla del objeto con mayor número de correspondencias será el candidato para identificar el objeto de la imagen.

Para asegurar el reconocimiento correcto sin llamar a SIFT de forma constante, se ha elaborado un cuantificador que tras varias llamadas a SIFT calificará la calidad del reconocimiento basándose en los reconocimientos anteriores. Dicha cualidad es la ya explicada variable interna del objeto 'Precision'. Cuando la precisión supera cierto umbral se recuadrará el objeto de distinto color:

- Amarillo : el objeto recuadrado es altamente probable que corresponda con la plantilla con el que el algoritmo lo ha relacionado.
- Verde: el objeto recuadrado coincide con seguridad con la plantilla que se indica.

Ilustración 5: Funcionamiento de SIFT y MATCH

4. Seguimiento del objeto

Para poder conocer que el objeto tiene permanencia en el tiempo y darle una identidad, es necesario elaborar un modelo de seguimiento del objeto. El algoritmo implementado en el programa es muy sencillo.

Para determinar si un objeto que existe en el frame anterior sigue existiendo en el frame actual, se ha empleado la propiedad del objeto de EquivDiameter. Dado que la lista de objetos existentes se almacena en memoria, los nuevos objetos son comparados con los antiguos. En el caso de que el nuevo objeto cumpla los requisitos de la distancia mínima de EquivDiameter/5, se

asignará la identidad del antiguo al nuevo, y se eliminará de la lista el antiguo para evitar duplicaciones de objetos. A cada nueva copia que se realiza del objeto, se amplía el vector Tracking que va almacenando todos los centroides de todos los objetos que han tenido dicha identidad. De este modo se puede conocer la trayectoria que ha realizado el objeto en la escena.

En la imagen siguiente se puede ver cómo funciona el algoritmo, siendo la trayectoria verde el recorrido en la escena que ha realizado el LOGOTIPO. En el caso del TELEFONO1 que se ve al fondo, no se puede visualizar apenas la trayectoria establecida dado que el objeto acaba de entrar en la escena y su recorrido aún es muy bajo.

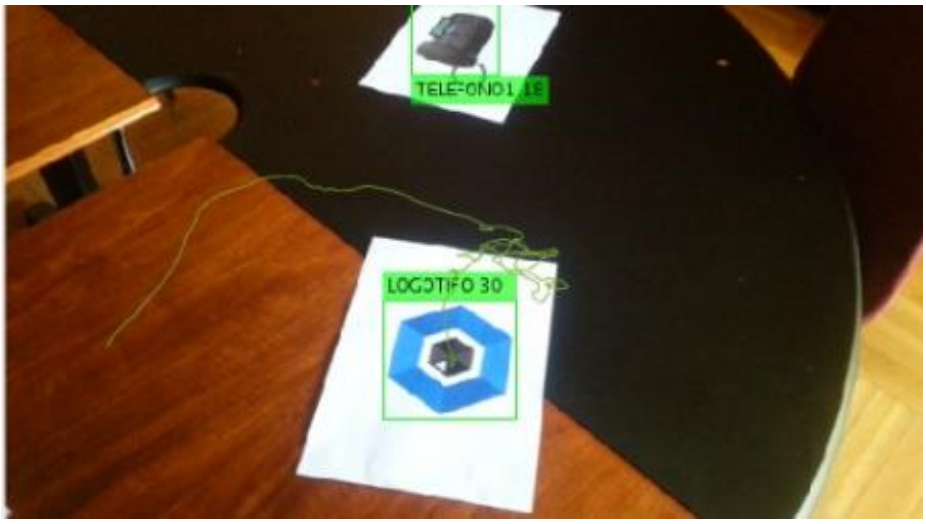

Ilustración 6: Ejemplo de seguimiento del objeto reconocido

4.1. Filtro de Kalman

El algoritmo propuesto tan solo es capaz de conocer las trayectorias anteriores y actual. Pero gracias al filtro de Kalman es posible realizar una predicción de la posición gracias a las posiciones anteriores.

Esta aplicación es particularmente útil para casos en los que los movimientos de los objetos en la escena llevan una velocidad o aceleración constantes como en seguimiento de peatones o vehículos. En esta ocasión, la cámara es transportada por un usuario y los movimientos son muy poco predecibles y erráticos, por no hablar de que las velocidades y aceleraciones del movimiento son completamente aleatorias.

El filtro de Kalman en este caso se puede emplear para intervalos de tiempo cortos en los que el objeto se haya perdido y no se desplace demasiado dado que las predicciones del filtro no serán válidas para intervalos muy grandes.

5. Odometría Visual

La Odometría consiste en la localización en el espacio de un elemento en un entorno. Para poder orientarse en el espacio es necesario el uso de visión binocular, en este caso las imágenes solo son tomadas con una cámara monocular, pero es posible hacer uso de los frames anteriores y siguientes para tener la información que se necesita para una reconstrucción tridimensional del entorno.

Los primeros pasos son como la detección de objetos, mediante el uso de SIFT y MATCH es posible detectar los puntos comunes entre dos imágenes. Esta vez no es necesario detectar el objeto únicamente y se puede aplicar el algoritmo a toda la imagen.

Es posible que algunas de estas características no se unan correctamente, por lo que es altamente recomendable hacer uso de algoritmos de limpieza como es el de RANSAC.

Finalmente, con los puntos obtenidos es posible obtener la matriz fundamental y la geometría epipolar de la escena. Con esos datos se pueden convertir los puntos de la escena en distancias en el espacio tridimensional. Si se repite este proceso durante toda la escena, unido con una construcción homográfica de los distintos entornos, es posible reconstruir en 3D el entorno de la habitación grabado.

6. Conclusiones

El algoritmo de identificación de objetos es el principal elemento del programa, ya que sobre él se estructuran el resto de procesos y su buena depuración de elementos no deseados es vital. Por lo tanto, se puede afirmar que es un éxito dada su robustez ante ruidos y brillos, siendo estos últimos muy destacables en la escena.

El seguimiento del objeto ha sido facilitado por la buena calidad del reconocimiento, haciendo posible localizar y mantener la trayectoria durante la estancia del objeto en la escena.