Universidad de Santiago de Chile
Facultad de Ingeniería
Departamento de Ingeniería Mecánica.
Sistema de Control de Procesos.

Proyecto: Sistema de Control de Procesos
” Detector de objetos mediante una capa de profundidad con Kinect v1”

Profesor: Renato Salinas

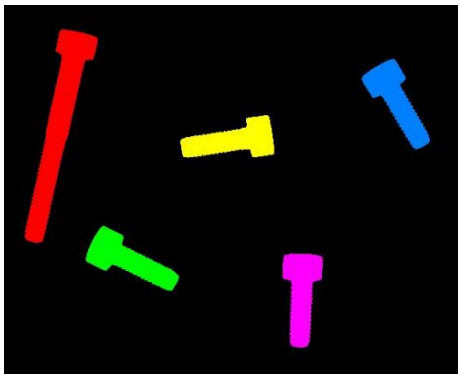
Alumnos: Iván Fernández

Claudio Canales D.

Índice

Resumen

Este proyecto aborda el problema de etiquetado de objetos que consiste en asignar una etiqueta única a todos los píxeles de cada objeto en una imagen binaria. El etiquetado es indispensable para distinguir diferentes objetos en una imagen binaria, y es un requisito previo para el análisis de la imagen y el reconocimiento de objetos en la imagen. Por lo tanto, el etiquetado de objetos es uno de los procesos más importantes para el análisis de imágenes, la comprensión de imágenes, el reconocimiento de patrones y la visión por computadora. En este proyecto, revisamos los algoritmos de etiquetado de objetos más modernos, explicamos las principales estrategias y algoritmos, presentamos sus pseudo-códigos y damos resultados experimentales para poner orden en los algoritmos.



Objetivos específicos

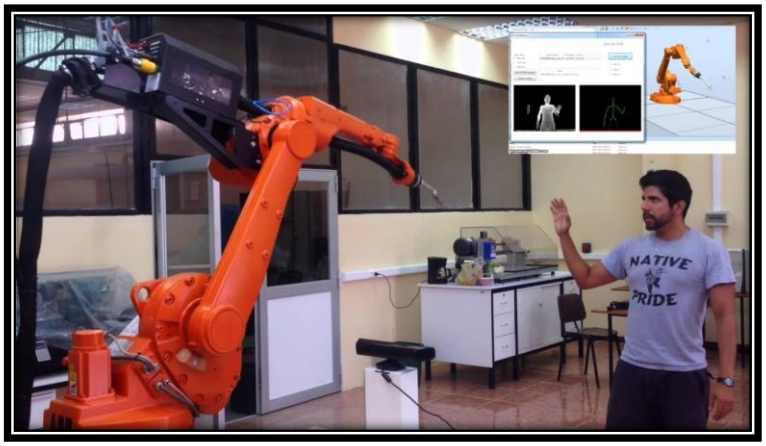
- Conectar el dispositivo Kinect v1 a la computadora con sistema operativo Windows
- Reconocimiento de Kinect v1 en el programa Matlab
- Obtención de la capa de profundidad de Kinect v1 para ser manipulada
- Algoritmos para la etiquetación de objetos a partir de una imagen binaria

Introducción.

Kinect v1 como desarrollo tecnológico

Para interiorizar y conocer el potencial de Kinect en el futuro para el crecimiento e innovación tecnología se buscan en la red proyectos ya comenzados a realizarse. Se encuentra tesis de algunos de universidades en España y algunas en chile (Universidad de concepción y Universidad de Chile). Algunos ejemplos recolectados de estas tesis son:

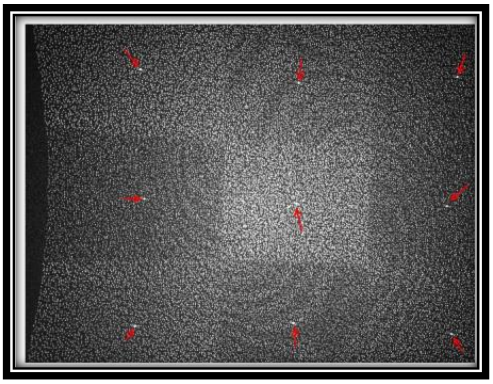
- Reconocimiento de esqueletos de distinta forma con métodos matemáticos vectoriales y de reconcomiendo en base a machine learning.
- Algoritmo de identificación de las posiciones para la poda de las parras mediante técnicas de procesamiento de imágenes.
- Implemento de algoritmos de procesamiento de imágenes para reconocer posición y color de objetos específicos en 3D (Esferas, cuadrados, triángulos)
- Movimiento de brazo robótico por medio de la posición de brazo humano reconocida por Kinect
- Sistema de control de televisión mediante Kinect y redes neuronales
- Generación de maniquís 3D a partir de imágenes del sensor Kinect
- Procesamiento de imágenes adquiridas por medio del sensor Kinect para determinar la posibilidad de una víctima en situaciones de peligro determinadas
- Moldeado 3D de cabeza y rasgos faciales mediante Kinect
- Variables cinemáticas de determinados puntos de control de un individuo mediante procesamiento de imágenes en Kinect
- Reconocimiento de gestos y actitudes corporales seminconscientes e inconscientes con visualizados con sensor Kinect



Descripción de los dispositivos interiores de Kinect v1.

Los componentes que utiliza Kinect para la interacción con el individuo, la adquisición de imágenes y de distancia a objetos son:

- Sensores de profundidad infrarrojo: Compuesto por un proyector de rayos infrarrojos donde se refleja la luz en los objetos y son capturados por un sensor monocromático. El sensor detecta los segmentos de puntos reflejados y estima la profundidad a partir de la intensidad y la distorsión de los mismos. Captura 30 cuadros por segundo con una resolución de 640x480 píxeles.



- Cámara RGB: La cámara RGB (rojo, verde, azul) es del tipo CMOS, trabaja por defecto de 640x480 a 30 cuadros por segundo. Para transmitir los datos de los pixeles utiliza comunicaciones serial.

- Micrófono Multi Matriz: Conjunto de cuatro micrófonos para el reconocimiento de órdenes y charla con el individuo.
- Inclinación motorizada: Permite ajustar la cámara hacia arriba o hacia abajo hasta 27º utilizada principalmente para la calibración de la cámara para utilizar un algoritmo que impone condiciones iniciales.

Drivers asociados a Windows de Kinect v1

Kinect para Windows SDK v1.8:

El Software de Desarrollo de Kinect para Windows (SDK) es un conjunto de herramientas que permiten al programador crear aplicaciones capaces de interactuar con el dispositivo. Destinada únicamente destinada a la investigación y al desarrollo de aplicaciones no comerciales. Soporta lenguaje de programación C+ y C++ con Visual estudio.

Tiene los siguientes elementos:

Controladores de Microsoft para Kinect:

- Captura y procesado de sonido a través de la API de audio estándar de Windows.
- Captura y transmisión de datos de imagen y profundidad.
- Funciones que permiten a una aplicación utilizar más de un sensor Kinect conectado a la computadora.

NUI API

- Se trata de un conjunto de APIs que obtienen datos de los sensores de imagen y controlan los dispositivos.

AUDIO API

- API que soporta y gestiona el sistema de micrófonos.

Requisitos del Sistema:

Sistema operativo compatible:

- Windows 7, Windows 8, Windows 8.1, Windows Embedded Standard 7

Requisitos de hardware

- Su computadora debe tener las siguientes capacidades mínimas:
- Procesador de 32 bits (x86) o 64 bits (x64)
- Procesador de doble núcleo a 2.66 GHz o más rápido
- Bus USB 2.0 dedicado
- 2 GB de RAM
- Un sensor de Microsoft Kinect para Windows

Requisitos de Software

- Visual Studio 2010 o Visual Studio 2012. Las ediciones Express gratuitas se pueden descargar de Microsoft Visual Studio 2010 Express o Microsoft Visual Studio 2012 Express.
- NET Framework 4 (instalado con Visual Studio 2010) o .NET Framwork 4.5 (instalado con Visual Studio 2012)
- Para desarrollar Kinect habilitado para voz para aplicaciones de Windows, debe instalar el Microsoft Tech Platform SDK v11

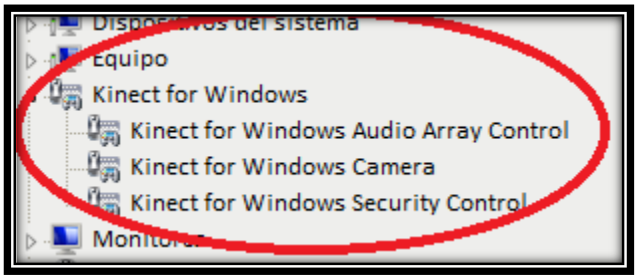
Para instalar el SDK:

- Asegúrese de que el sensor Kinect no esté enchufado a ninguno de los puertos USB de la computadora.
- Si tiene instalada una versión anterior de Kinect para Windows SDK, cierre las muestras abiertas, el Sample Browser, etc. y salte al paso 5. Kinect para Windows v1.8 actualizará la versión anterior.
- Retire cualquier otro controlador para el sensor Kinect.
- Si tiene instalado Microsoft Server Speech Platform 10.2, desinstale los componentes de Microsoft Server Speech Platform Runtime y SDK, incluidas las versiones de bit x86 y x64, más el lenguaje de reconocimiento de voz de Microsoft Server - Kinect Language Pack.
- Cierre Visual Studio. Debe cerrar Visual Studio antes de instalar el SDK y luego reiniciarlo después de la instalación para recoger las variables de entorno que requiere el SDK.
- Una vez que el SDK haya terminado de instalarse correctamente, asegúrese de que el sensor Kinect esté enchufado a una fuente de alimentación externa y luego enchufe el sensor Kinect en el puerto USB de la PC. Los controladores se cargarán automáticamente.
- El sensor Kinect ahora debería estar funcionando correctamente.

Reconocimiento del dispositivo:

Para verificar la instalación se conecta el kinect por el Puerto USB y en ventana de administración de dispositivos aparecerá de la siguiente forma:

- Microsoft Kinect Audio Array Control
- Microsoft Kinect Camera
- Microsoft Kinect Device



Soportes asociados a Matlab de Kinect v1

Image Acquisition Toolbox Support:

Image Acquisition Toolbox Support ofrece funciones y módulos que le permiten conectar cámaras industriales y científicos para MATLAB y Simulink. Incluye una aplicación MATLAB que le permite detectar y configurar de manera interactiva las propiedades del hardware. La caja de herramientas permite los modos de adquisición, como el procesamiento en el bucle, la activación del hardware, la adquisición en segundo plano y la adquisición sincronizada en múltiples dispositivos. Adquisición de imágenes de la caja de herramientas es compatible con todos los estándares y los proveedores de hardware, incluyendo USB3 Visión, GigE Visión , y GenICam GenTL. Puede conectarse a cámaras de profundidad 3D, cámaras de visión artificial y capturadores de cuadros, así como a dispositivos científicos e industriales de alta gama.

Es compatible con cámaras digitales que siguen la especificación de cámara digital (DCAM) basada en IIDC 1394 desarrollada por 1394 Trade Association. La especificación DCAM basada en IIDC 1394 describe una interfaz genérica para intercambiar datos con cámaras digitales IEEE 1394 (FireWire).

Soporte Microsoft Kinect para Windows de Image Acquisition Toolbox

De Image Acquisition Toolbox Support, se puede adquirir datos de Microsoft Kinect para Windows en MATLAB y Simulink. Microsoft Kinect para Windows es un dispositivo de interacción natural con una cámara RGB y un sensor de profundidad 3D. Microsoft Kinect para Windows se puede utilizar para realizar aplicaciones en campos como robóticos, kinesiología e ingeniería civil. Por ejemplo, puede dirigir un robot utilizando Kinect construyendo un modelo de su entorno con el sensor de profundidad 3D.

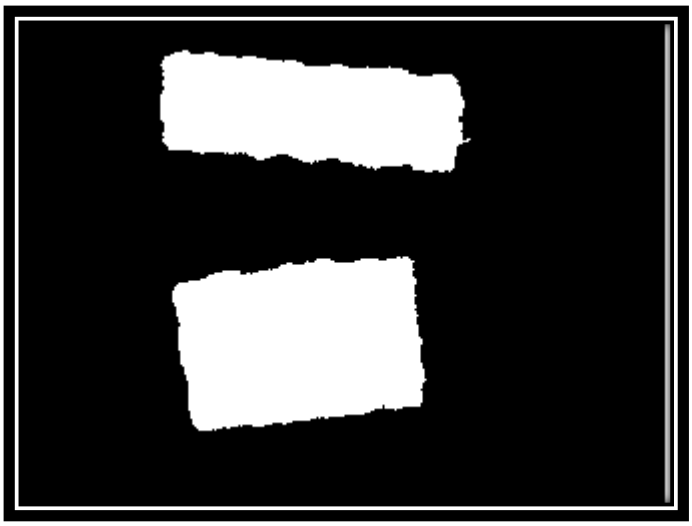
Segmentación de Imágenes con Kinect.

Para poder catalogar los diferentes elementos en la imagen es importante segmentarlos por algún tipo de atributo. En Visión por computadora, existen métodos de detección de regiones en una imagen digital, las cuales difieren en propiedades como brillo o color comparado con el resto de la imagen. Un método conocido ampliamente es “Blob Detection”, el cual permite obtener una región “blob”, donde en una parte de la imagen estos atributos o propiedades son constante.

El enfoque de segmentación de imágenes tratado en este trabajo es distinto. En este trabajo se aprovecha la información de profundidad de las cámaras del Kinect, con el fin, de obtener una segmentación en función de la profundidad de los objetos. Por lo tanto, nuestro método consiste en aplicar limites inferiores y superiores de profundidad, para así poder segmentar la presencia de un objeto, en una imagen digital.

Los resultados de la segmentación de los objetos realizadas en el Kinect se presentan a continuación, esta segmentación da origen a una matriz binaria de 0 y 1, donde hay presencia de 1, implica la presencia de una anomalía.

A continuación, se presenta un ejemplo de segmentación realizada por distancia con Kinect.



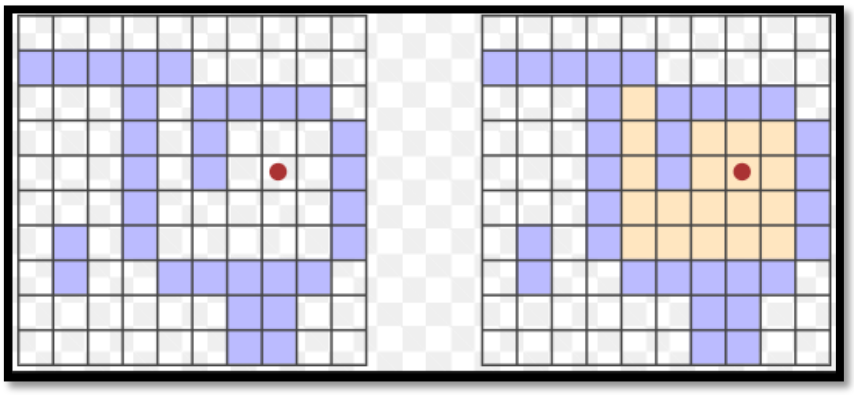
Se puede apreciar de la imagen binaria, dos rectángulos, los que representan 2 objetos capturados en tiempo real utilizando la Kinect para x360. Cabe mencionar, que como estamos midiendo distancia desde la cámara, existirá una superficie esférica equidistante de la cámara, donde los objetos se podrán observar y no un plano de estudio. En este trabajo se descarta esa problemática, debido que la zona estudiada tiende a ser plana.

Otra problemática que se presenta al momento de utilizar el Kinect, es la obtención de una doble imagen debido al efecto del proyector infrarrojo, esta doble imagen se segmente de forma similar, para así poder distinguir la proyección real de la sombra.

Algoritmo Basado En Flood Fill.

Para poder contar los elementos, se utiliza un algoritmo de pintado denominado “Flood Fill”, este algoritmo permite pintar de forma recursiva toda la extensión de área delimitada por otro color. La implementación de este algoritmo es la siguiente:

1. Inicialmente se busca en toda la imagen donde hay una anomalía (valor 1).
2. Si pilla una anomalía, inmediatamente inicia el pintado de la extensión de área cerrada.
3. Se inicia el proceso de pintado de la imagen global con un identificador de contador, como se puede apreciar en la siguiente figura, se inicia el proceso de pintado en un pixel y finalmente con la recursividad se pinta toda la superficie.
4. El algoritmo de relleno, inicia en un punto, este punto pinta los 4 puntos vecinos a este, siempre y cuando cumplan con las condiciones de pintado y cada uno de los 4 puntos pintados ejecuta la misma secuencia. En ultima instancia, cuando no hay donde mas por pintar, la recursividad se detiene.



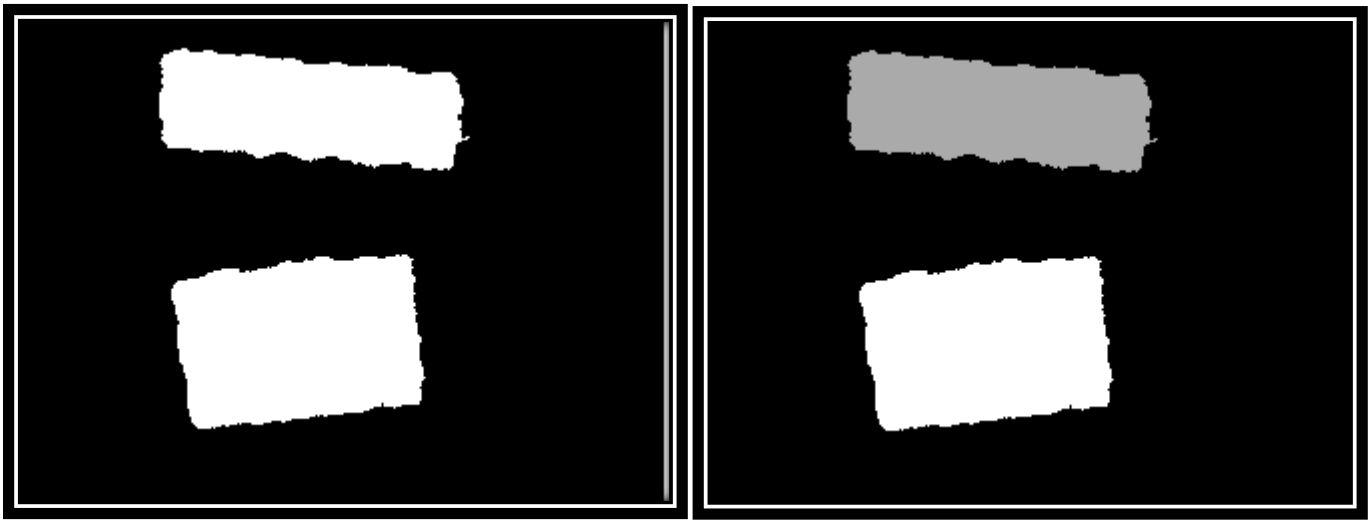
5. Una vez finalizado el proceso de pintado de la imagen, se procede a buscar otra sección de la imagen con anomalía, para reiterar el proceso de pintado.
6. Cada vez que se llama a la función de pintado, se suma 1 al contador, siempre y cuando el área de pintado sea superior a cierta cantidad de pixeles.
7. Finalmente, se obtiene una imagen con identificadores y se procede a transformarla en una imagen a color.

Este algoritmo da buenos resultados, aun así, es bastante costoso computacionalmente. A continuación se presentaran los resultados obtenidos.

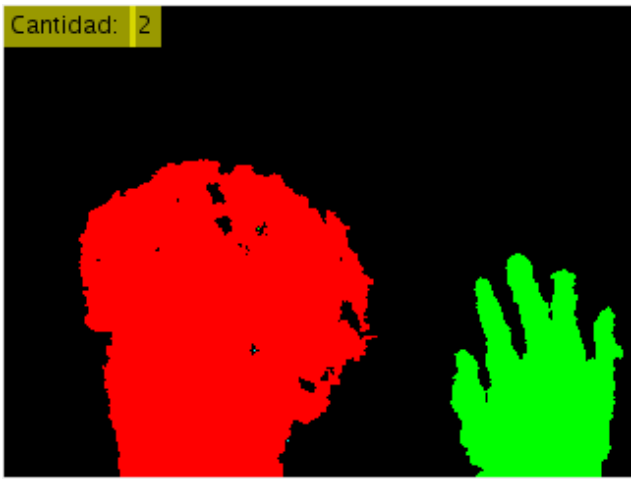
Resultados experimentales algoritmo Flood Fill.

Posterior a la segmentación de imágenes, se puede apreciar cada una de las imágenes con identificadores a continuación, cabe mencionar que los valores enteros son ploteados en una escala de grises:

Antes	Después

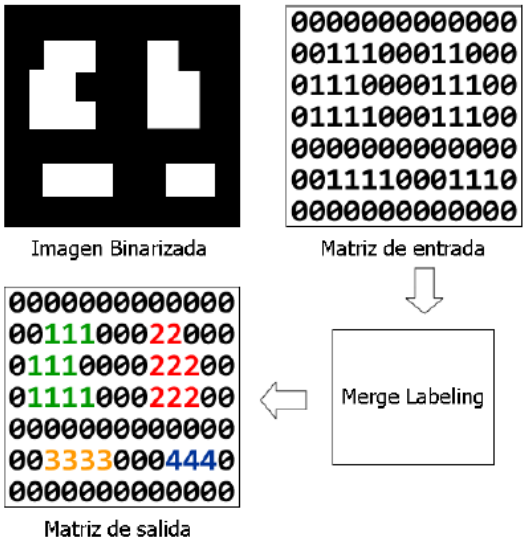


Como se puede apreciar de la imagen anterior, se puede observar como la imagen binaria es transformada a otra en escala de grises. Finalmente, estas imágenes son convertidas a color y son procesadas de forma continua, con el fin de obtener las imágenes en tiempo real. A continuación, se presenta una imagen en tiempo real, procesada de forma continua con este algoritmo.



Algoritmo de Etiquetación Merge Labeling

El algoritmo etiqueta como único aquellos pixeles activos próximos (detectados por la imagen de profundidad de Kinect) pertenecientes al mismo objeto. El proceso del algoritmo junta objetos etiquetados por separado cuando estos están unidos por al menos un pixel. El algoritmo de etiquetación recibe como entrada la imagen binarizada que contiene los pixeles activos correspondientes a los objetos a identificar en la escena, cuya salida es una matiz de etiquetas para cada uno de los pixeles de la imagen, en donde cada etiqueta identifica el objeto al cual dicho pixel pertenece.

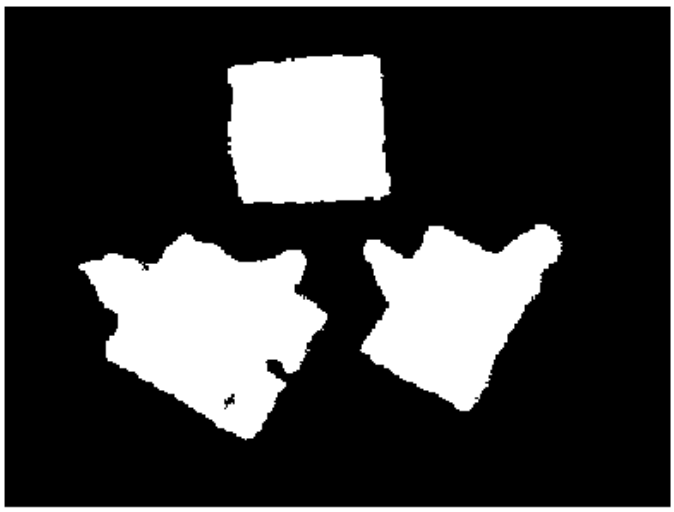


El resumen del algoritmo de etiquedato Merge Labeling es el siguiente:

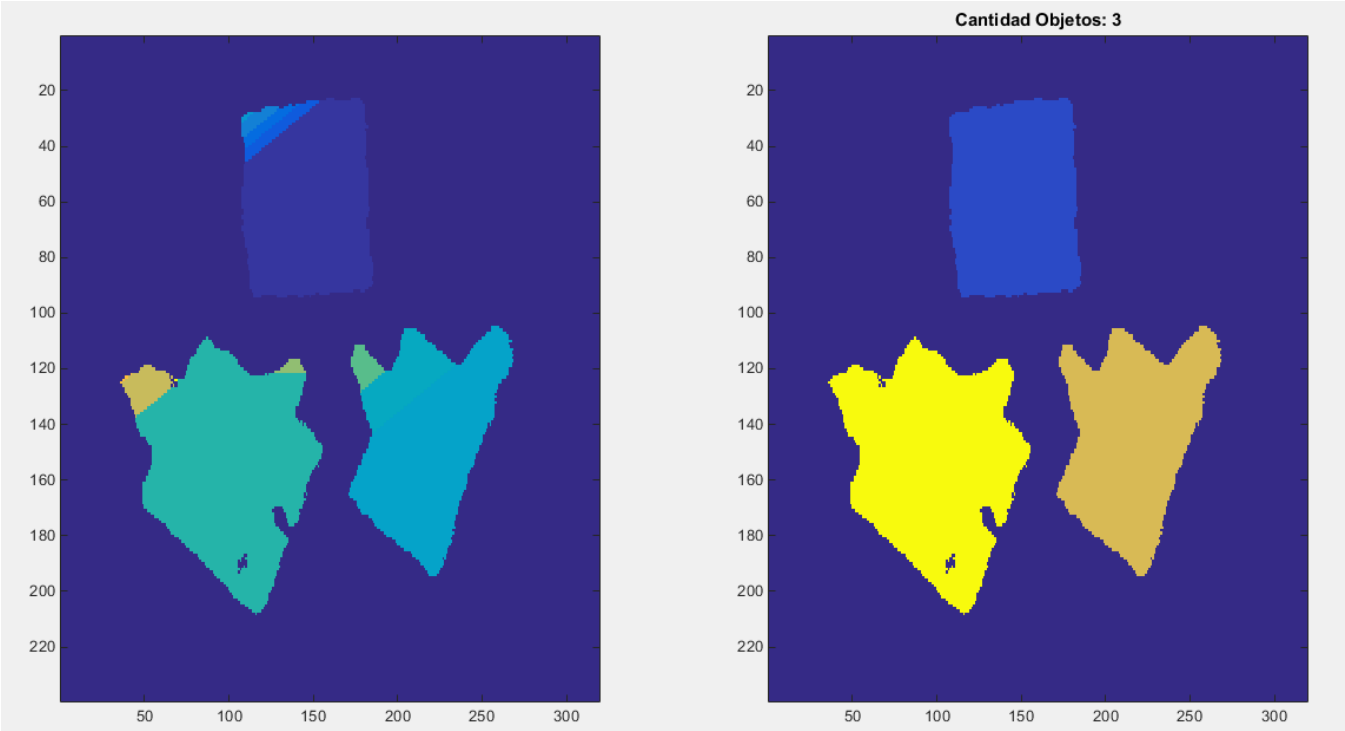
```
1. Por cada píxel P de la matriz de entrada, por cada fila y columna
2.   Si P tiene etiqueta
3.     Si todos los vecinos etiquetados tienen la misma etiqueta
4.        Se marcan los vecinos no etiquetados con dicha etiqueta
5.     Si no.
6.        Se obtiene la lista de vecinos presentes en la vecindad
7.        Se procede a hacer el cambio de etiquetas (merge)
8.   Si P no tiene etiqueta
9.     Si todos sus vecinos no están etiquetados o no tiene vecinos
10.       Se marca el pixel con la etiqueta siguiente
11.       Se marcan todos los vecinos no etiquetados con la misma etiqueta
12.       Incrementamos el valor de la etiqueta
13.    Y Si todos los vecinos etiquetados son diferentes
14.       Se obtiene la lista ordenada de vecinos presentes en la vecindad
15.       Se procede a hacer el cambio de etiquetas (merge)
16.    Si no.
17.       Se marca el pixel con la etiqueta de sus vecinos
18.       Se marcan todos los vecinos no etiquetados con la misma etiqueta
```

Resultados experimentales algoritmo de Etiquetación Merge Labeling

Imagen binaria de objetos:



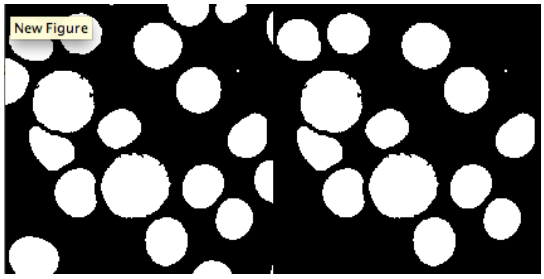
Resultados del algoritmo de Merge Labeling no optimizado:



Funciones y Algoritmos de procesamiento de imágenes en Matlab

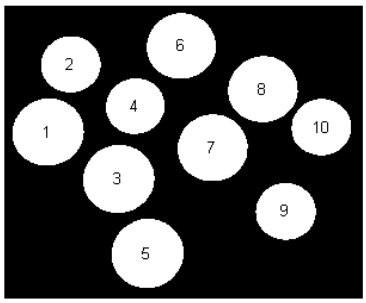
Imclearborder:

Suprime las estructuras de la imagen que son más claras que sus alrededores y que están conectadas al borde de la imagen. Se utiliza esta función para borrar el borde de la imagen. Para las imágenes en escala de grises, imclearborder tiende a reducir el nivel de intensidad además de suprimir las estructuras de borde. La imagen de salida es escala de grises o binaria, dependiendo de la entrada.



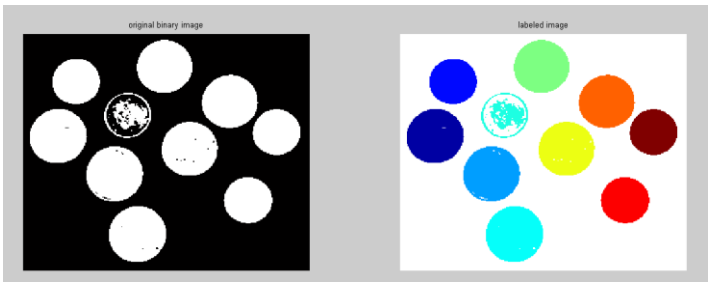
Bwlabel:

Etiquetar componentes conectados en 2-D imagen binaria. Devuelve también una matriz que contiene etiquetas para los objetos conectados que se encuentran en la en la imagen de blanco y negro.



Label2rgb

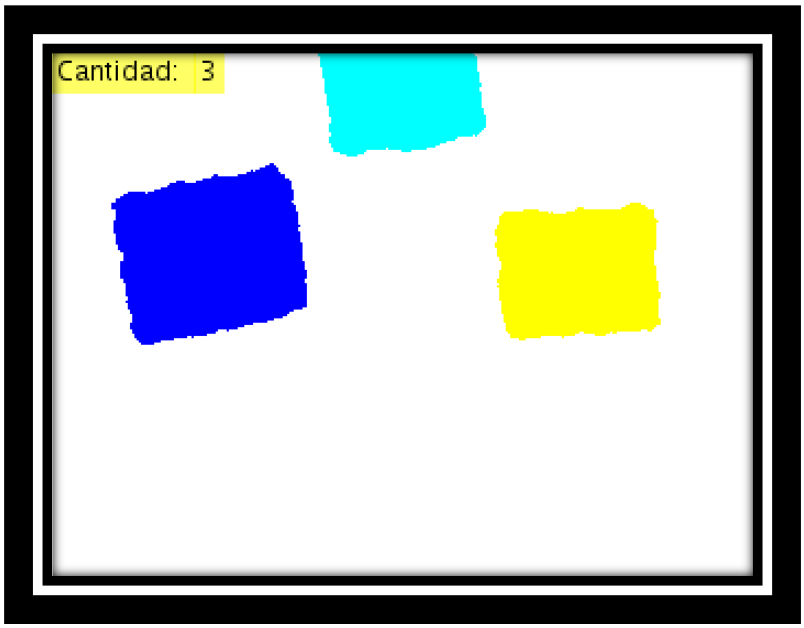
Convierte una matriz de etiqueta en una imagen de color RGB para el propósito de visualizar las regiones marcadas. La label2rgb función determina el color que se asignará a cada objeto en función del número de objetos en la matriz de etiquetas. La función selecciona colores de toda la gama del mapa de colores.



Resultados experimentales de funciones y algoritmos de procesamiento de imágenes en Matlab

Imagen binaria de objetos:

Resultado de funciones de etiquetado por Matlab:



Conclusión

Segmentación de Imagen Con Kinect.

Lo más importante para poder contar los objetos, es la apropiada segmentación. El enfoque trabajado se basa en el factor dimensional de los objetos y de su proyección en la cámara. Con esta información y una cámara de profundidad se pueden segmentar los diferentes objetos, aun así esta herramienta presenta algunos contras y desventajas. Uno de los contras es que existe dificultad para segmentar objetos que sean demasiado planos en profundidad y se encuentren cercanos a una pared, ya que no será posible distinguir entre la pared y el objeto. Aun así, en estos casos se podría realizar una segmentación con una imagen y corroborar con la imagen de profundidad. Una de las grandes ventajas de utilizar la cámara de profundidad, es la posibilidad de cuantificar las dimensiones de los objetos que están siendo observados.

Algoritmo basado en Flood Fill

Con el algoritmo desarrollado y la implementación de “Flood Fill”, se obtiene un resultado bastante confiable y robusto. Este algoritmo implementado permite identificar un objeto que presente severas discontinuidades, sin mayor problema. Aun así, el algoritmo es bastante costoso computacionalmente para ser utilizado con grandes resoluciones de manera continua y presenta limitantes cuando dos objetos están tocándose. Un enfoque que daría mejores resultados es la utilización de tanto un análisis de una imagen RGB y un análisis con imagen de profundidad. El análisis de segmentación RGB escapa de los objetivos de este trabajo y es por aquello que no se profundiza en aquello.

Mergen Labeling

Acerca del algoritmo de Mergen para la etiquetación de objetos se puede concluir que cumple con su objetivo pero solo para ciertas circunstancias. Los tiempos de procesamiento son aproximadamente de 8 segundos, por lo que para una detección de objetos en tiempo real en movimiento el algoritmo no detecta rápidamente la cantidad de objetos. Otro problema se produce si dos objetos están en contacto, estos se etiquetaran solo como un objeto. Si la imagen tiene objetos en las orillas, el algoritmo entrar a analizar pixeles no existentes, por lo que el programa muestra error. Finalmente cabe hacer notar que el algoritmo puede ser modificado a futuro para menores tiempos de detección de objetos y el presentado en este proyecto sirve para casos de tiempo estacionario.

Funciones Matlab

Sobre el paquete de funciones de procesamiento de imagines de Matlab se puede destacar la gran optimización de sus algoritmos ya que funcionan relativamente rápido en comparación con los programas básicos. Por ende el conteo de objetos en tiempo real es muy eficiente y tiene poca inestabilidad. Sin embargo para movimientos muy bruscos de objetos, la imagen de los objetos se distorsiona y dificulta el procesamiento para el conteo de objetos correcto

Proyecto

Finalmente, se concluye que el algoritmo que da mejores resultados es el que se basa en “Flood Fill”, ya que cuantifica los objetos de forma precisa, con o sin discontinuidades y exacta. Cabe destacar que el gran problema de este algoritmo es el costo computacional, por lo tanto, se recomendaría realizar un enfoque para la detección de objetos combinando este algoritmo con Deep Learning y Convolutional Neural Networks.

Bibliografía

- *Introduccion a la Vision Artificial* – DptGo. Electronica, Automatica e informatica Industrial - Carlos Platero
- *Vision artificial y Procesamiento Digital de Imagenes usando Matlab* – Ivan Danilo Garcia Santillan

Anexo Codigo Flood Fill

```
clc; clear all ; close all
imaqreset;
% --------------------------------------------------------------------
% DETECCION DE OBJETOS Y CONTEO EN IMAGEN.
%---------------------------------------------------------------------
% INICIALIZACION DE KINECT,SENSOR DE DISTANCIA
%---------------------------------------------------------------------
% PARAMETROS DE FUNCIONAMIENTO
%---------------------------------------------------------------------
th_min=800;                         %Ingresar en mm, 800 ES EL MINIMO
th_max=1000;
min_pix=5000;
%---------------------------------------------------------------------
%Flood Fill
%---------------------------------------------------------------------
depthVid= videoinput('kinect',2); % CREA EL OBJETO DE VIDEO.
triggerconfig (depthVid, 'manual');
depthVid.FramesPerTrigger=1;
depthVid.TriggerRepeat=inf;
%viewer=vision.DeployableVideoPlayer();
start(depthVid);
himg=figure;
global contador;
global depthMap;
global l;
global pix_contador;
c=zeros(240,320,3);
contador=0;
pix_contador=0;
ii=0;
%---------------------------------------------------------------------
%Captura de Imagen.
%---------------------------------------------------------------------
while ishandle(himg)
    contador=0;
trigger(depthVid);
[cap,~,depthMetaData]=getdata(depthVid);
depthMap=cap(121:360,161:480);
for i=1:240
 for j=1:320
  if((depthMap(i,j)<=th_min))
   depthMap(i,j)=0;
  elseif((depthMap(i,j)>=th_max))
   depthMap(i,j)=0;
  else
   depthMap(i,j)=1;
```

```
  end
 end
end
%-------------------------------------------------------------------------
%LLAMADA A Flood Fill START
%-------------------------------------------------------------------------
sz=size(depthMap);
l=logical(depthMap);
%%if(ii==5)
for i=1:sz(1)
    for j=1:sz(2)
        if(l(i,j)==1)
            pix_contador=0;
       sq(i,j,1,contador+2);
       if(pix_contador>=min_pix)
       contador=contador+1;
       end
        end
    end
end
for i=1:sz(1)
    for j=1:sz(2)
         if(depthMap(i,j)==0)
       c(i,j,:)=[0,0,0];
        end
        if(depthMap(i,j)==1)
       c(i,j,:)=[0,0,255];
        end
          if(depthMap(i,j)==2)
       c(i,j,:)=[255,0,0];
           end
         if(depthMap(i,j)==3)
       c(i,j,:)=[0,255,0];
          end
         if(depthMap(i,j)==4)
       c(i,j,:)=[0,255,255];
        end
    end
end
%end
%-------------------------------------------------------------------------
%LLAMADA A Flood Fill END
%-------------------------------------------------------------------------
pix_contador=0;
c = insertText(c,[0 0],'Cantidad: ');
c = insertText(c,[65,0],contador);
imshow(c)
```

```
contador
end
%stop(depthVid);


%--------------------------------------------------------------------------
%Flood Fill- ESTA FUNCION RECURSIVA HACE LA MAGIA
%--------------------------------------------------------------------------

function sq(x,y,t,r)
global depthMap;
global pix_contador;
global l;
sz=size(depthMap);
if(t==r)
    return
end
if(l(x,y)~=t)
    return
end
l(x,y)=0;
depthMap(x,y)=r;
pix_contador=1+pix_contador;
if(x<sz(1))
sq(x+1,y,t,r);
end
if(x>1)
sq(x-1,y,t,r);
end
if(y<sz(2))
sq(x,y+1,t,r);
end
if(y>1)
sq(x,y-1,t,r);
end
end
```

Anexo Codigo Merge Labeling

```
clc; clear all ; close all
%Programa cuenta los objetos en una imagen a partir de el metodo de Merge
%Laberning incompleto (se utiliza doble iteracion por derecha y por izquierda)

%load('imagen.mat')

% Captura imagen binaria con objetos
load('cap5.mat')
figure (1)
%cap=cap5;
cap1=cap;
capcortado=cap(120+1:360-1,160+1:480-1);
[x y]=size(capcortado);
etiquetas=zeros(x,y);
imshow(cap1,[0 1])
tic


% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%DE DERECHA A IZQUIERDA

contetiqueta=0;
for i=2:x-1
    for j=2:y-1
        if (capcortado(i,j)==1)
            [eti1,xxx,yyy]=pixelvecinos(etiquetas,i,j); % pixel vecinos etiquetas 0 1 2 3 ....
            [acti1,xxxx,yyyy]=vecactivos(capcortado,i,j);%pixel vecinos activos (1 o 0)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                % ¿Los vecinos etiquetados son todos iguales ?   si =0 no =1
                cont=1;
                diferentes=0;
                poseticigual=0;
                for jj=1:8
                    if eti1(1,jj)~=0 ;
                        if cont==1;
                            eti2viejo=eti1(1,jj);
                            poseticigual=jj;
                            cont=2;
                        else cont==2;
                            if eti1(1,jj)~=eti2viejo;
                                diferentes=diferentes+1;
                        end
                    end
                    end
                end
```

```
                %SALIDAS: POSETICIGUAL , DIFERENTES

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5
% ¿ TODOS LOS VECINOS ETIQUETADOS SON DIFERENTES?
                cont=1;
                cont3=0;
                cont4=0;
                diferentes1=0;
                poseticigual1=0;
                for jj=1:8
                    if eti1(1,jj)~=0 ;
                        cont3=cont3+1;
                        if cont==1;
                            eti2viejo=eti1(1,jj);
                            poseticigual1=jj;
                            cont=2;
                        else cont==2;
                            if eti1(1,jj)-(eti2viejo)~=0 & cont==2;
                               cont4=cont4+1;
                            end
                        end
                    end
                end
                numeropixelesestiquetados=cont3;
                numeropixelesdiferentes=cont4+1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
% Menor etiqueta de los vecinos
                menorviejo=1000;
                for jj=1:8
                    if eti1(1,jj)~=0 & acti1(1,jj)==1;
                        menornuevo=eti1(1,jj);
                        if menorviejo-menornuevo>0;
                           menorviejo=menornuevo;
                        end
                    end
                end
                % salida menor viejo
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
% Si el pixel (i,j) tiene etiqueta entonces:

            if etiquetas(i,j)~=0; %% si tiene etiqueca
                % Si todos los vecinos etiquetados tiene la misma etiqueta:
                if diferentes==0;
                    for jj=1:8
                        if eti1(1,jj)==0 & acti1==1;

etiquetas(xxx(1,jj),yyy(1,jj))=etiquetas(xxx(1,poseticigual),yyy(1,poseticigual));
```

```
                end
            end
        % Si no
        else
            %%MERGE: pintar pixeles alrededor con el menor valor
            for jj=1:8
                if acti1(1,jj)==1
                    etiquetas(xxxx(1,jj),yyyy(1,jj))=menorviejo; %etiquetas(i,j);
                    etiquetas(i,j)=menorviejo;
                end
            end
            %disp('hola1')
         end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Si el pixel(i,j) no tiene etiqueta entonces:

       else
           % Si todos los vecinos no estan etiquetados o no tienen vec
           if sum(eti1)==0 || sum(acti1)==0;
               % se marca el pixel con la etiqueta siguiente
               etiquetas(i,j)=contetiqueta+1;
               % se marcan lso vecinos no etiquetados con la misma
               % etiqueta
               for jj=1:8
                   if eti1(1,jj)==0 & acti1(1,jj)==1;
                       etiquetas(xxxx(1,jj),yyyy(1,jj))=etiquetas(i,j);
                   end
               end
               % se incrementa la etiqueta
               contetiqueta=contetiqueta+1;
           % si todos los vecinos etiquetados son diferentes
           elseif  numeropixelesestiquetados==numeropixelesdiferentes;
              %%MERGE
               for jj=1:8
                   if acti1(1,jj)==1
                       etiquetas(xxxx(1,jj),yyyy(1,jj))=menorviejo;
                       etiquetas(i,j)=menorviejo;
                   end
               end
              %disp('hola2')
           % SI NO
           else
               % se marca el pixel con la etiqueta de los vecinos
               etiquetas(i,j)=etiquetas(xxx(1,poseticigual),yyy(1,poseticigual));
               % se marcan todos los vecinos no etiquetados con ma
               % misma etiqueta
               for jj=1:8
                   if eti1(1,jj)==0 & acti1(1,jj)==1
```

```
etiquetas(xxxx(1,jj),yyyy(1,jj))=etiquetas(xxx(1,poseticigual),yyy(1,poseticigual));
                        end
                    end
                end
            end
        end
    end
    %disp( contetiqueta)
    %figure (2)
    %imshow(etiquetas, [0 1])
    %imagesc(etiquetas);
    %pause(0);
end

%Plotear
contetiqueta;
figure (3)
%imshow(etiquetas, [0 1])
etiquetavieja=etiquetas;
contetiqueta=0;


% DE IZQUIERDA A DERECHA
for i=x-1:-1:2
    for j=y-1:-1:2
        if (capcortado(i,j)==1)
            [eti1,xxx,yyy]=pixelvecinos(etiquetas,i,j); % pixel vecinos etiquetas 0 1 2 3 ....
            [acti1,xxxx,yyyy]=vecactivos(capcortado,i,j);%pixel vecinos activos (1 o 0)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                % ¿Los vecinos etiquetados son todos iguales ?   si =0 no =1
                cont=1;
                diferentes=0;
                poseticigual=0;
                for jj=1:8
                    if eti1(1,jj)~=0 ;
                        if cont==1;
                            eti2viejo=eti1(1,jj);
                            poseticigual=jj;
                            cont=2;
                        else cont==2;
                            if eti1(1,jj)~=eti2viejo;
                                diferentes=diferentes+1;
                        end
                    end
                    end
                end

                %SALIDAS: POSETICIGUAL , DIFERENTES

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5
```

```
% SI TODOS LOS VECINOS ETIQUETADOS SON DIFERENTES
                cont=1;
                cont3=0;
                cont4=0;
                diferentes1=0;
                poseticigual1=0;
                for jj=1:8
                    if eti1(1,jj)~=0 ;
                        cont3=cont3+1;
                        if cont==1;
                            eti2viejo=eti1(1,jj);
                            poseticigual1=jj;
                            cont=2;
                        else cont==2;
                            if eti1(1,jj)-(eti2viejo)~=0 & cont==2;
                               cont4=cont4+1;
                            end
                        end
                    end
                end
                numeropixelesestiquetados=cont3;
                numeropixelesdiferentes=cont4+1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
%Menor de los vecinos
                menorviejo=1000;
                for jj=1:8
                    if eti1(1,jj)~=0 & acti1(1,jj)==1;
                        menornuevo=eti1(1,jj);
                        if menorviejo-menornuevo>0;
                           menorviejo=menornuevo;
                        end
                    end
                end
                % salida menor viejo
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%

            if etiquetas(i,j)~=0; %% si tiene etiqueca
                % Si todos los vecinos etiquetados tiene la misma etiqueta:
                if diferentes==0;
                    for jj=1:8
                        if eti1(1,jj)==0 & acti1==1;

etiquetas(xxx(1,jj),yyy(1,jj))=etiquetas(xxx(1,poseticigual),yyy(1,poseticigual));
                        end
                    end
               % Si no
               else
```

```
                    %%MERGE
                    for jj=1:8
                        if acti1(1,jj)==1
                            etiquetas(xxxx(1,jj),yyyy(1,jj))=menorviejo; %etiquetas(i,j);
                            etiquetas(i,j)=menorviejo;
                        end
                    end
                    %disp('hola1')
                end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            else
                % Si todos los vecinos no estan etiquetados o no tienen vec
                if sum(eti1)==0 || sum(acti1)==0;
                    % se marca el pixel con la etiqueta siguiente
                    etiquetas(i,j)=contetiqueta+1;
                    % se marcan lso vecinos no etiquetados con la misma
                    % etiqueta
                    for jj=1:8
                        if eti1(1,jj)==0 & acti1(1,jj)==1;
                            etiquetas(xxxx(1,jj),yyyy(1,jj))=etiquetas(i,j);
                        end
                    end
                    % se incrementa la etiqueta
                    contetiqueta=contetiqueta+1;
                % si todos los vecinos etiquetados son diferentes
                elseif  numeropixelesestiquetados==numeropixelesdiferentes;
                   %%MERGE: etiqueta vecinos con el nuemro menor de ellos
                    for jj=1:8
                        if acti1(1,jj)==1
                            etiquetas(xxxx(1,jj),yyyy(1,jj))=menorviejo;
                            etiquetas(i,j)=menorviejo;
                        end
                    end
                   %disp('hola2')
                % SI NO
                else
                    % se marca el pixel con la etiqueta de los vecinos
                    etiquetas(i,j)=etiquetas(xxx(1,poseticigual),yyy(1,poseticigual));
                    % se marcan todos los vecinos no etiquetados con ma
                    % misma etiqueta
                    for jj=1:8
                        if eti1(1,jj)==0 & acti1(1,jj)==1

etiquetas(xxxx(1,jj),yyyy(1,jj))=etiquetas(xxx(1,poseticigual),yyy(1,poseticigual));
                        end
                    end
                end
            end
        end
```

```
    end
    %disp( contetiqueta)
    %figure (2)
    %imshow(etiquetas, [0 1])
    %imagesc(etiquetas);
    %pause(0);
end

%PLOTEAR RESULTADOS FINALES
contetiqueta=unique(etiquetas);
[x,y]=size(contetiqueta);
cantidad= x-1;
figure (3)
%imshow(etiquetas, [0 1])
subplot(1,2,2) ;imagesc(etiquetas)
title(['Cantidad Objetos: ', num2str(cantidad)])
subplot(1,2,1);imagesc(etiquetavieja)
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
toc
```

Anexo Codigo con funciones de procesamiento de imagenes en Matlab

```
clc; clear all ; close all
% deteccion de objetos tiempo real y calcula cuantos
% objetos hay

% INICIALIZACION DE KINECT, CAMARA RGB Y SENSOR DE DISTANCIA
imaqreset; %Borra los objetos de adquicicion de todos los adaptadores que entran a la toolbox

% Limites entre los que el Kinect captara objetos
th_min=800;         %Ingresar en mm
th_max=1000;         %mm

depthVid= videoinput('kinect',2); %Creo un objeto de video.
triggerconfig (depthVid, 'manual'); %configura la toma de fotogramas
depthVid.FramesPerTrigger=1; %cantidad de fotogramas
depthVid.TriggerRepeat=inf;

%set(getselectedsource(depthVid),'TrackingMode','Skeleton');
viewer=vision.DeployableVideoPlayer(); % Visualizacion por medio de un reproductor de video
start(depthVid); % Inicia el objeto Kinect
himg=figure;
pause(1)

ii=1;
tic
while ishandle(himg)
trigger(depthVid);
[depthMap,~,depthMetaData]=getdata(depthVid); % Extraccion capa de profundidad
a=depthMap(320,240);


% Seleccionar region de interes (Region central de la imagen)
for i=1:480
 for j=1:640
    if(i<=120|j<=160|i>=360|j>=480)
        depthMap(i,j)=1;
    end
 end
end


% Funcion que activa (les da valor 1) a los objetos que esten entre los
% limites antes seleccionados y a los que esten fuera de los limites les da
% valor 0 (Imagen binaria)
for i=120:360
 for j=160:480
  if((depthMap(i,j)<=th_min))
```

```
    depthMap(i,j)=0;
   elseif((depthMap(i,j)>=th_max))
    depthMap(i,j)=0;
   else
    depthMap(i,j)=1;
   end
  end
end

% Plotea la imagen de profundidad Binaria
imshow(depthMap,[0 1]);
b=depthMap(320,240);
[a b];

cap=depthMap;
ii=1+ii;
if(ii==10)
 cap=depthMap;
 toc
 disp(toc)
 break
end
ii;

%figure (1)
%cap1=cap;
%imshow(cap1,[0 1])
[etiquetas num]=bwlabel(cap(120+1:360-1,160+1:480-1)); %% detecta objetos y los etiqueta en la region de
interes
figure (1)
color=label2rgb(etiquetas);   %% da color a los objetos etiquetados
color = insertText(color,[0 0],'Cantidad: ');
color = insertText(color,[65,0],num);
foto=ii-1;
color = insertText(color,[80 0],'Foto: ');
color = insertText(color,[140,0],foto);
%subplot(1,2,1);imshow(cap1(120+1:360-1,160+1:480-1),[0 1]);subplot(1,2,2);
imshow(color);
%disp('Numero de objetos: ')
%Numero=num


end
stop(depthVid);
```