



CENTRO DE ENSEÑANZA TECNICA INDUSTRIAL

PRACTICAS DE LABORATORIO

INGENIERÍA CARRERA	PLAN DE ESTUDIO	CLAVE ASIGNATURA	NOMBRE DE LA ASIGNATURA
INGENIERÍA DE DESARROLLO DE SOFTWARE	2025-B	19SDS32	Procesamiento de Imágenes
ALUMNO		FECHA	EVALUACIÓN

PRÁCTICA No.	LABORATORIO DE COMPUTACIÓN No	NOMBRE DE LA PRÁCTICA	DURACIÓN (HORAS)
10	LCS	Contador de objetos	2

INTRODUCCIÓN

En el ámbito del procesamiento de imágenes, una de las tareas más fundamentales es la identificación y análisis de componentes conectados en imágenes binarias. Este proceso, conocido como etiquetado de componentes, es ampliamente utilizado en aplicaciones como el conteo de objetos, la segmentación de regiones específicas y la detección de patrones. En una imagen binaria, los píxeles se representan como valores de 0 y 1, donde el fondo corresponde al valor 0 y los objetos tienen un valor de 1. La principal finalidad de este etiquetado es asignar un identificador único a cada región conectada de píxeles con valor 1, lo que permite distinguir de manera efectiva cada objeto presente en la imagen.

El etiquetado de componentes conectado con vecindad de 4 píxeles es un método eficiente y sencillo que considera únicamente las conexiones horizontales y verticales entre píxeles, dejando de lado las conexiones diagonales. Este enfoque es adecuado para escenarios donde las formas de los objetos no requieren una evaluación de conexiones más complejas. El algoritmo que se implementa en esta práctica se basa en recorrer la imagen en dos fases principales: la asignación inicial de etiquetas y la resolución de colisiones entre etiquetas asignadas a un mismo objeto. Finalmente, se reorganizan las etiquetas para facilitar su interpretación y se presenta una visualización que utiliza colores para diferenciar cada objeto etiquetado.

El estudio y la implementación de este algoritmo no solo permiten entender la lógica detrás del etiquetado de componentes conectados, sino también fortalecen las habilidades en la manipulación de matrices, estructuras de control y visualización de datos utilizando MATLAB, un entorno ampliamente utilizado en investigación y desarrollo.

OBJETIVO (COMPETENCIA)

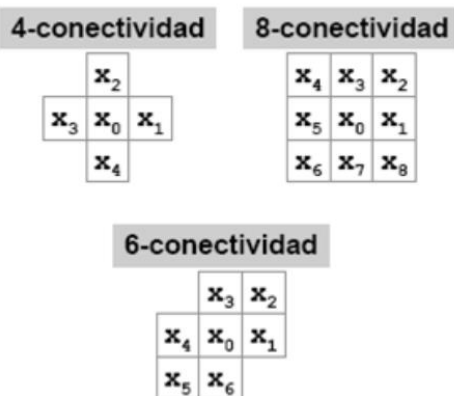
El objetivo principal de esta práctica es desarrollar e implementar un algoritmo eficiente para identificar y etiquetar componentes conectados en una imagen binaria mediante el uso de vecindad de 4 píxeles. Se busca comprender y aplicar cada una de las fases del algoritmo, que incluyen la asignación inicial de etiquetas, la resolución de colisiones y el re-etiquetado de los objetos. Además, se pretende optimizar el manejo de etiquetas para garantizar una identificación precisa de los objetos conectados y generar una visualización clara y diferenciada.

Con este enfoque, se busca fortalecer el entendimiento de los procesos básicos del procesamiento de imágenes, así como las habilidades prácticas en programación utilizando MATLAB. Este conocimiento tiene aplicaciones directas en áreas como visión por computadora, inteligencia artificial y análisis de datos visuales.

FUNDAMENTO

El etiquetado de componente conexas es una operación sobre imágenes binarias que se emplea para distinguir objetos (previamente binarizados) presentes en una imagen.

En primer lugar se debe definir la conectividad entre píxeles, es decir, en qué manera se considerará que un grupo de píxeles pertenecen o forman un objeto. Un píxel, de la proximidad espacial entre píxeles de una imagen, se puede decir que tiene 8 posibles "vecinos": dos horizontales, dos verticales, y cuatro diagonales. Se puede definir la conectividad de un píxel en tres maneras distintas: cuatro-vecinos, ocho-vecinos y seis-vecinos:

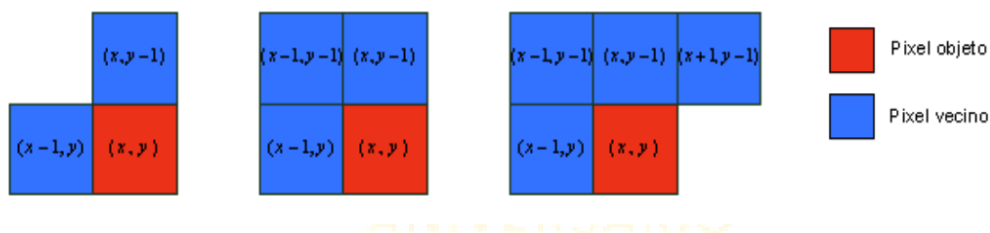


La idea es encontrar todos los píxeles conectados en un mismo objeto dentro de la imagen, y asignar a cada uno de ellos una sola etiqueta.

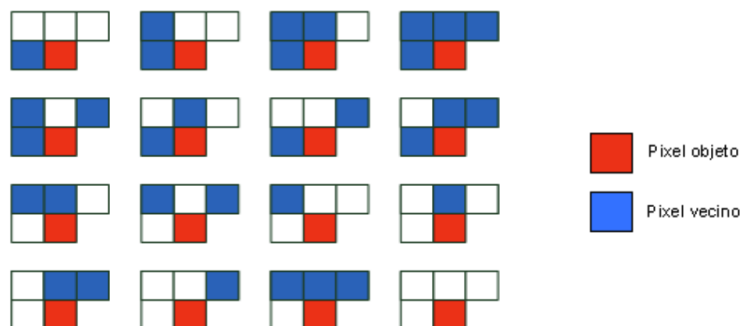
Una forma muy simple de realizar el etiquetado es:

1. Recorrer la imagen para encontrar un píxel correspondiente a un objeto no etiquetado y asignarle una nueva etiqueta, supongamos, de valor L.
2. Asignar recursivamente la etiqueta L a todos sus vecinos (que se comenta más abajo) que sean además píxeles objeto y no estén etiquetados con anterioridad.
3. Si aún quedan píxeles objeto no etiquetados volver al paso 1.

El algoritmo que se propone es iterativo recorriéndose dos veces la imagen a etiquetar, en una primera pasada se extiende las etiquetas necesarias par identificar a todos los píxeles objetos, y una segunda pasada donde se le asigna a cada píxel perteneciente a un objeto, la etiqueta identificativa correspondiente. A partir de las definiciones de vecindad, para el algoritmo de etiquetado definiremos los siguientes tipos de variantes para la vecindad de un píxel objeto de coordenadas (x,y):



Para resolver el problema del etiquetado se propone elegir una conectividad basada en 8-vecinos. Siendo así, si de partida se nos presenta una imagen binaria, los distintos casos de píxeles conectados y que forman parte de un objeto son:





CENTRO DE ENSEÑANZA TECNICA INDUSTRIAL

PRACTICAS DE LABORATORIO

El algoritmo de etiquetado necesario para diferenciar los objetos de la imagen será un proceso que nos requerirá escanear la imagen y etiquetar, identificar mediante un número (etiqueta), los objetos conexos.

METODOLOGÍA (DESARROLLO DE LA PRACTICA)

El código implementa un algoritmo de etiquetado de componentes conectadas en una imagen binaria utilizando la vecindad de 4 píxeles. Su objetivo principal es identificar y asignar etiquetas únicas a las regiones conectadas de píxeles con valor 1 (objetos) mientras se diferencian del fondo (valor 0).

Descripción general del funcionamiento

1. Preparación de la imagen:

La imagen binaria es cargada y preparada para el procesamiento, asegurándose de que los valores sean 0 (fondo) y 1 (objetos). Se convierte a tipo double para permitir asignar etiquetas numéricas mayores a 1.

2. Asignación inicial de etiquetas:

La imagen se recorre píxel por píxel, evaluando únicamente los píxeles de valor. Cada píxel objeto analiza sus vecinos izquierdo y superior:

- Si ambos vecinos son 0, se asigna una nueva etiqueta.
- Si uno de los vecinos ya tiene una etiqueta, el píxel hereda esa etiqueta.
- Si ambos vecinos tienen etiquetas distintas, el píxel recibe la menor etiqueta y se registra una colisión entre las etiquetas.

3. Resolución de colisiones:

Las colisiones registradas en la etapa anterior son resueltas reemplazando etiquetas mayores por menores en toda la imagen, unificando así las etiquetas de un mismo componente conectado.

4. Re-etiquetado:

Las etiquetas resultantes se reorganizan para que sean consecutivas y más fáciles de interpretar, comenzando desde 1.

5. Visualización del resultado:

La imagen etiquetada es representada gráficamente, asignando colores únicos a cada etiqueta para diferenciar visualmente los objetos.

Código:

```
% Programa para etiquetar los objetos contenidos en una imagen binaria,  
% utilizando vecindad de 4 píxeles.  
% Lectura de la imagen binaria (donde 0 es fondo y 1 son objetos)  
Im = imread('figuras.png');  
figure;  
imshow(Im);  
title('Imagen original');  
if size(Im, 3) > 1  
    Im = rgb2gray(Im); % Convertir a escala de grises si no es binaria  
end  
% Asegurarse de que la imagen sea binaria  
Im = imbinarize(Im);  
[m, n] = size(Im);  
% Convertir la imagen binaria a tipo double para etiquetado  
Ibd = double(Im);  
% Paso 1: Asignación inicial de etiquetas  
% Inicialización de etiquetas y detección de colisiones  
e = 2; % Primera etiqueta  
collisions = []; % Lista para registrar colisiones  
% Etiquetado inicial con vecindad 4  
for r = 2:m  
    for c = 2:n  
        if Ibd(r, c) == 1  
            left = Ibd(r, c - 1); % Vecino izquierdo  
            top = Ibd(r - 1, c); % Vecino superior  
  
            if left == 0 && top == 0  
                % Nueva etiqueta  
                Ibd(r, c) = e;  
                e = e + 1;  
            elseif left > 1 && top == 0  
                % Asignar etiqueta del vecino izquierdo  
                Ibd(r, c) = left;  
            elseif top > 1 && left == 0  
                % Asignar etiqueta del vecino superior  
                Ibd(r, c) = top;  
            elseif left > 1 && top > 1  
                % Resolver conflicto de etiquetas  
                Ibd(r, c) = min(left, top);  
                if left ~= top  
                    % Registrar colisión entre etiquetas  
                    collisions = [collisions; min(left, top), max(left, top)];  
                end  
            end  
        end  
    end  
end
```

```
end
% Paso 2: Resolución de colisiones
if ~isempty(collisions)
    collisions = unique(collisions, 'rows'); % Eliminar duplicados
    for i = 1:size(collisions, 1)
        min_label = collisions(i, 1);
        max_label = collisions(i, 2);
        % Reemplazar todas las etiquetas max_label por min_label
        Ibd(Ibd == max_label) = min_label;
    end
end
% Paso 3: Re-etiquetado para garantizar etiquetas consecutivas
unique_labels = unique(Ibd(:)); % Etiquetas únicas
new_labels = 0:length(unique_labels) - 1;
for i = 2:length(unique_labels)
    Ibd(Ibd == unique_labels(i)) = new_labels(i);
end
% Paso 4: Mostrar la imagen etiquetada
figure;
imshow(label2rgb(Ibd, 'jet', 'k')); % Visualización con colores
title('Imagen Etiquetada');
```

RESULTADOS Y CONCLUSIONES



CENTRO DE ENSEÑANZA TECNICA INDUSTRIAL

PRACTICAS DE LABORATORIO

REFERENCIAS

1. Calvo, E., Brox, P., & Sánchez-Solano, S. (2012). Un algoritmo en tiempo real para etiquetado de componentes conectados en imágenes. Recuperado de <https://digital.csic.es/bitstream/10261/84143/1/algoritmo%20en%20tiempo%20real.pdf>
2. Greyrat, R. (2022). Python OpenCV: etiquetado y análisis de componentes conectados. Barcelona Geeks. Recuperado de <https://barcelongeeks.com/python-opencv-etiquetado-y-analisis-de-componentes-conectados/>
3. Procesamiento de imágenes con Python: componentes conectados y etiquetado de regiones. (s.f.). ICHI.PRO. Recuperado de <https://ichi.pro/es/procesamiento-de-imagenes-con-python-componentes-conectados-y-etiquetado-de-regiones-69196627360081>
4. Etiquetado de componentes conectados - Documentación de ImageMagick. (s.f.). Herramientas en línea. Recuperado de <https://toolonline.net/es/CodeDocs/ImageMagick/connected-components>

Elavoro	Observaciones	Evaluacion
DR. Gerardo García Gil		