

Detecciones de Esquina

Tomas Lopez Pérez

June 2024

Índice general

1. Introducción	2
1.1. Introduction	2
1.2. Fundamento teórico	2
2. Metodología	4
3. Resultados	7
4. Conclusiones	10

1. Introducción

1.1. Introduction

El procesamiento de imágenes es un campo fundamental en la visión por computadora y en numerosas aplicaciones tecnológicas modernas. Uno de los aspectos más críticos de este campo es la detección de bordes y contornos, ya que estos elementos proporcionan información esencial sobre la forma, estructura y características de los objetos presentes en una imagen. Los bordes representan transiciones bruscas en la intensidad de la imagen, mientras que los contornos delinean los límites de los objetos, facilitando su identificación y análisis.

1.2. Fundamento teórico

El Procesamiento de imágenes es un campo de investigación muy extenso que involucra diversas áreas del conocimiento, ya que involucra diversos procesos, tales como: la adquisición, transmisión, representación y procesamiento. En términos generales el procesamiento de imágenes se utiliza para modificar una imagen con el fin de mejorar la apariencia visual para un observador y para resaltar convenientemente el contenido de la imagen de cara a la percepción por parte de máquinas, es decir, se hace una manipulación de imágenes con objeto de producir nuevas imágenes que son mejores, en algún sentido. El procesamiento de imágenes comprende distintas técnicas que se utilizan para mejorar, restaurar e inclusive comprimir imágenes ya existentes; la implementación y desarrollo de dichas técnicas agregadas a sistemas ya existentes nos permiten editar imágenes de forma más sencilla y precisa [?]. La Figura 1.1 muestra el flujo de las etapas del procesamiento digital de imágenes propuestas por Gonzalez en donde se establece como la etapa inicial la adquisición de la imagen digital, lo que se puede hacer mediante algún sistema de sensores que permitan digitalizar la imagen u obtener las imágenes de algún repositorio o colección. El conjunto de imágenes adquiridas conforma inicialmente la base de conocimiento, ya que para algunas aplicaciones basta con tener la información pura, sin embargo en la mayoría de las aplicaciones del procesamiento de imágenes, es necesario enriquecer la base del conocimiento con al menos las etapas de procesado y segmentación, lo cual requiere aplicar una serie de operadores que transformen las imágenes para resaltar características relevantes, en el caso del procesado, o separar los diferentes elementos que componen la imagen, en el caso de la segmentación, de modo que los resultados de estas dos etapas permiten tener una base del conocimiento más amplia y robusta. Respecto a la etapa de representación y descripción. las imágenes

procesadas y/o segmentadas son usadas para extraer características más específicas sobre la imagen, tales como cantidad, dimensión, forma, posición, etc. de los objetos de interés en las imágenes, esta información también puede formar parte de la base del conocimiento, de modo que en la ultima etapa, reconocimiento e interpretación, se puedan obtener respuestas a partir de las imágenes y la información generada por estas.

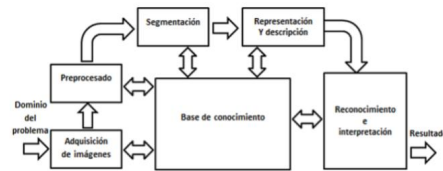


Figura 1.1: Etapas fundamentales del procesamiento digital de imágenes.

2. Metodología

La metodología propuesta en este trabajo constituye en la implementación de el algoritmo detector esquinas de Harris El detector de esquinas de Harris se basa en la premisa: existe un punto de esquina donde el gradiente de la imagen es especialmente fuerte en más de una dirección al mismo tiempo.

Como primer paso sera aplicar un prefiltro a mi imagen en escala de grises 2.1 utilizando el filtro smooth 2.2

$$I' \leftarrow I * H_p$$

Figura 2.1: Prefilter

$$H_{px} = \frac{1}{9} \begin{bmatrix} 2 & 5 & 2 \end{bmatrix} \quad \text{and} \quad H_{py} = H_{px}^T = \frac{1}{9} \begin{bmatrix} 2 \\ 5 \\ 2 \end{bmatrix}.$$

Figura 2.2: Smooth

Calculamos la derivada horizontal y vertical de imagen utilizando Hdx 2.3 y Hdy 2.4

$$H_{dx} = \begin{bmatrix} -0.453014 & 0 & 0.453014 \end{bmatrix}$$

Figura 2.3: Hdx

Calcular la matriz de estructura local M(u,v)

$$\begin{aligned} A &= I_x^2, \\ B &= I_y^2, \\ C &= I_x \cdot I_y. \end{aligned}$$

Aplicar filtro de suavizado a componentes de M se aplica la siguiente convolucion

$$H_{dy} = H_{dx}^T = \begin{bmatrix} -0.453014 \\ \mathbf{0} \\ 0.453014 \end{bmatrix}.$$

Figura 2.4: Hdy

de la matriz Hb

$$Hbx = [1 \quad 6 \quad 15 \quad 20 \quad 15 \quad 6 \quad 1], \quad Hby = \begin{bmatrix} 1 \\ 6 \\ 15 \\ 20 \\ 15 \\ 6 \\ 1 \end{bmatrix}$$

$$Hb = Hbx \cdot Hby$$

$$A' = \text{conv2}(\text{double}(A), Hb, 'same')$$

$$B' = \text{conv2}(\text{double}(B), Hb, 'same')$$

$$C' = \text{conv2}(\text{double}(C), Hb, 'same')$$

Encontrar esquinas locales máximas

```
Q = padarray(Q, [1, 1], 0);
[M,N] = size(Q);
Corners = [];
for u = 2:M-1
    for v = 2:N-1
        if Q(u,v) > th && IsLocalMax(Q, u , v)
            newCorner = [v,u, Q(u, v)];
            Corners = [Corners; newCorner];
        end
    end
end
Corners = sortrows(Corners, -3);
GoodCorners = uint16(CLEANUPNEIGHBORS(Corners,dmin));
```

Figura 2.5: localfunction

limpieza de esquinas cercanas

```

%Determine si Q(u,v) is a local maximum
function result = IsLocalMax(Q,u,v)
    qc = Q(u,v);
    N = neighbors(Q, u, v);
    result = all(qc >= N);
end

function N = neighbors(Q, u, v)
    neighbors = [
        u-1, v-1; % Vecino superior izquierdo
        u-1, v;   % Vecino superior
        u-1, v+1; % Vecino superior derecho
        u,   v-1; % Vecino izquierdo
        u,   v+1; % Vecino derecho
        u+1, v-1; % Vecino inferior izquierdo
        u+1, v;   % Vecino inferior
        u+1, v+1 % Vecino inferior derecho
    ];
    N = zeros(size(neighbors, 1), 1);
    for i = 1:size(neighbors, 1)
        u = neighbors(i, 1);
        v = neighbors(i, 2);
        N(i) = Q(u, v);
        fprintf('u = %d, v = %d\n', u, v); % Imprimir u y v
    end
end

```

Figura 2.6: localfunction

```

function GoodCorners = CLEANUPNEIGHBORS(Corners, dmin)
% Inicializan la lista de buenas esquinas
GoodCorners = [];

% Mientras la lista de Corners no esté vacía
while ~isempty(Corners)
    % Remover la primera esquina de Corners y añadirla a GoodCorners
    Ci = Corners(1, :);
    GoodCorners = [GoodCorners; Ci];

    % Eliminar la primera esquina de Corners
    Corners(1, :) = [];
    % Extraer el valor que representa la distancia (el último valor en este caso)
    dist_Ci = Ci(end);
    CjAll = Corners(:,end);
    distances = double(dist_Ci) - double(CjAll);
    %distances = distances/max(distances);
    % Eliminar todas las esquinas que estén dentro de dmin de Ci
    Corners(distances < dmin, :) = [];
end
end

```

Figura 2.7: Limpieza de esquinas

3. Resultados

En esta sección se presentan los resultados obtenidos en la implementación del algoritmo Harris corner detector. Para los resultados se realizaron 2 pruebas una imagen es un tablero de ajedrez el cual la otra es una imagen que se generon con ayuda de paint.

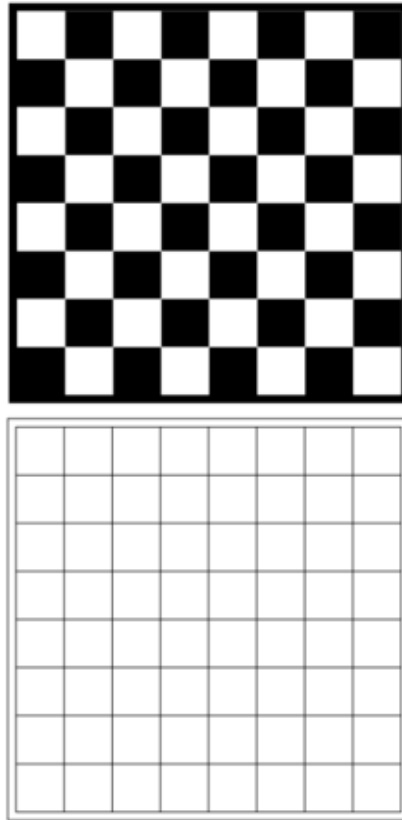


Figura 3.1: Imagen original

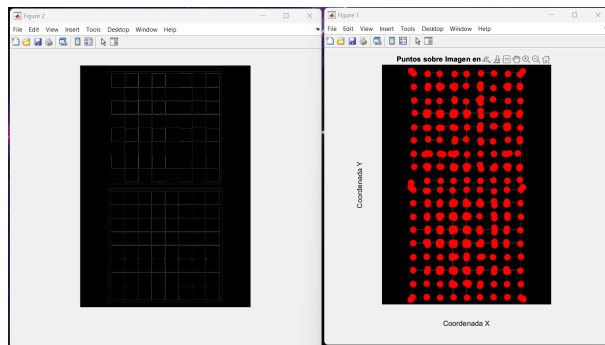


Figura 3.2: Imagen procesada

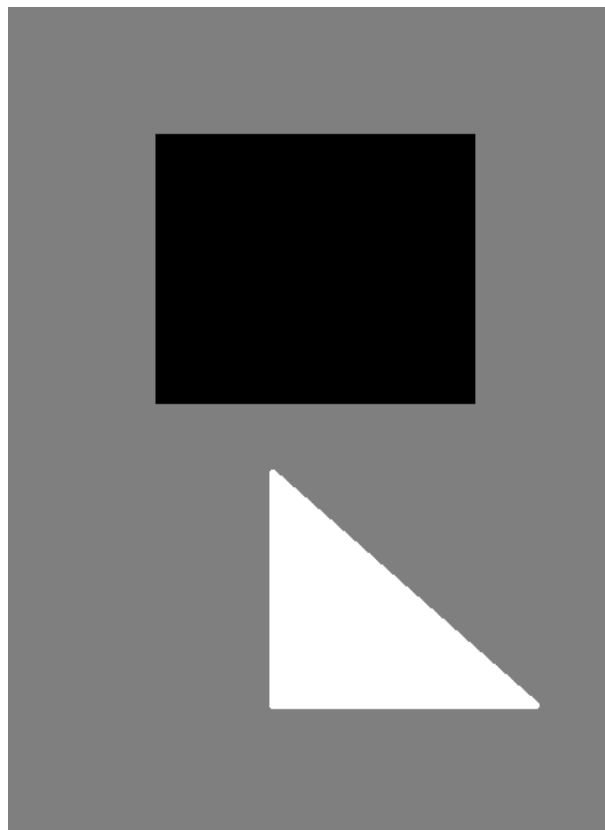


Figura 3.3: Imagen generada manualmente

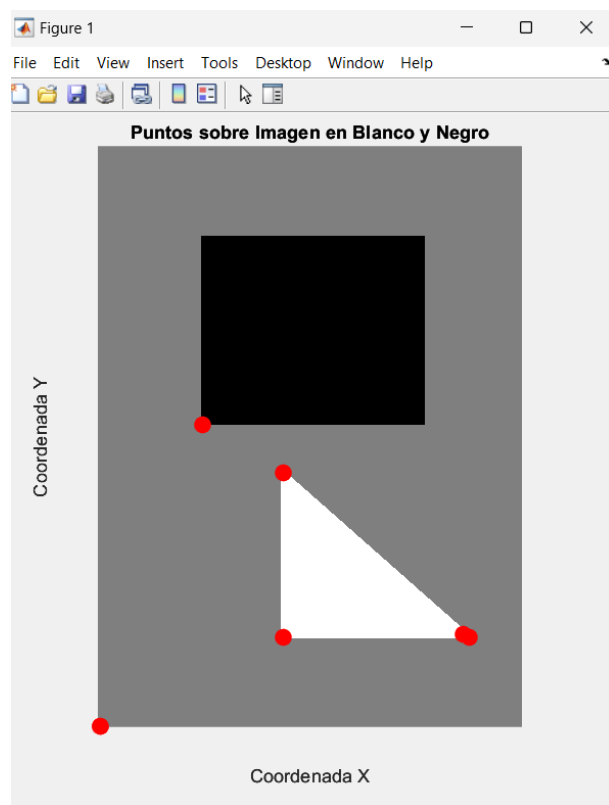


Figura 3.4: Imagen generada manualmente procesada

4. Conclusiones

El método de Harris proporciona una forma efectiva de detectar esquinas en imágenes, destacándose por su capacidad para identificar puntos de interés robustos que son invariantes a cambios en la escala, rotación y luminosidad. Sin embargo, la efectividad del método puede depender significativamente de la selección adecuada de parámetros y de la calidad del pre-procesamiento de la imagen.