###### University of the Punjab

###### Gujranwala Campus



**Subject:** Computer Vision

**Submitted By:**  Alishba Shahbaz

BIT21008

7th-Semester(Morning)

**Euclidian Distance**

A = zeros(100, 100); % Initialize a 100x100 matrix of zeros

Cx = 50; % X-coordinate of the circle's center

Cy = 50; % Y-coordinate of the circle's center

Radius = 20; % Radius of the circle

% Iterate through each pixel

for i = 1:1:100

for j = 1:1:100

% Check if the current pixel lies inside the circle

if sqrt((Cx - i)^2 + (Cy - j)^2) <= Radius

A(i, j) = 255; % Set pixel value to 255 (white)

end

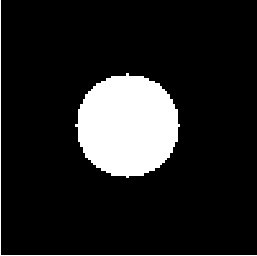
end

end

% Display the final result

imshow(A, []);

title('Circle in a Black Box');



**City Block Distance**

% Set higher resolution for smoother appearance

image\_size = 500; % Increase the resolution of the grid

Cx = 250; % X-coordinate of the center

Cy = 250; % Y-coordinate of the center

Radius = 100; % Increase radius to match the higher resolution

% Initialize the black image

A = zeros(image\_size, image\_size);

% Iterate through each pixel

for i = 1:1:image\_size

for j = 1:1:image\_size

% Check if the pixel lies within the city block distance

if abs(Cx - i) + abs(Cy - j) <= Radius

A(i, j) = 255; % Set pixel value to 255 (white)

end

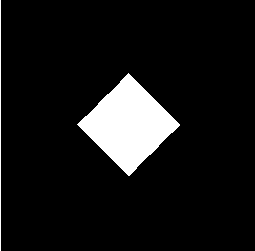
end

end

% Display the result

imshow(A, []);

title('City Block Distance (Improved Smoothness)');



**Chessboard Distance**

A = zeros(100, 100); % Initialize a 100x100 matrix of zeros

Cx = 50; % X-coordinate of the center

Cy = 50; % Y-coordinate of the center

Radius = 20; % Radius based on chessboard distance

% Iterate through each pixel

for i = 1:1:100

for j = 1:1:100

% Check if the current pixel is within the chessboard distance

if max(abs(Cx - i), abs(Cy - j)) <= Radius

A(i, j) = 255; % Set pixel value to 255 (white)

end

end

end

% Display the final result

imshow(A, []);

title('Chessboard Distance Square in a Black Box');

