```
// first, initialize CUDA environment
Initialize CUDA and allocate memory for input image and accumulator array.




// then, define a kernel function outside of the main function
Define a kernel function:
    Calculate x, y coords of the pixel this thread is responsible for
    blockId * blockDim + threadId
    For each pixel (x, y) in the image:
        If the pixel is an edge (Image[x, y] > 0):
            For each theta in range of angles:
                Calculate rho = x * cos(theta) + y * sin(theta)
                 Increment the accumulator at (theta, rho)


// Launch and call the kernel
Set grid and block dimensions
Call the kernel:
    houghTransform<<gridDimensions, blockDimensions>>(d_img, d_accumulator, width, height, max_rho, num_thetas)
Synchronize CUDA device to make sure all threads are done

// Copy results back to host and free space
Copy accumulator from Device to Host
Free device memory
Process the accumulator in MATLAB (call this C function in MATLAB)
```