~/Downloads/lab12.m

```
% Parameters
a = 0.1;   % Motion rate in x-direction
b = 0.1;   % Motion rate in y-direction
T = 1;     % Time constant
image = imread('rgb.jpeg');  % Replace with your image file

% Convert the image to grayscale if it is RGB
if size(image, 3) == 3
    image = rgb2gray(image);
end

% Normalize the image to double precision for processing
image = double(image);

% Get the size of the image
[M, N] = size(image);

% Step 1: Create a motion blur kernel based on the equations of motion
t = 1;  % Assume 1 unit time for simplicity

% Compute the displacement in the x and y directions
dx = (a * t) / T;   % Displacement in x-direction
dy = (b * t) / T;   % Displacement in y-direction

% Define the size of the blur kernel (kernel size based on the displacement)
kernel_size_x = max(round(abs(dx)), 1);   % Ensure the kernel size is at least 1
kernel_size_y = max(round(abs(dy)), 1);   % Ensure the kernel size is at least 1

% Step 2: Create a motion blur kernel (simple rectangular motion blur)
motion_kernel = zeros(kernel_size_x, kernel_size_y);

% Apply a simple rectangular blur kernel (set the last element to 1)
motion_kernel(kernel_size_x, kernel_size_y) = 1;

% Normalize the kernel
motion_kernel = motion_kernel / sum(motion_kernel(:));

% Step 3: Apply the motion blur using convolution
blurred_image = conv2(image, motion_kernel, 'same');  % Convolve the image with
the kernel

% Step 4: Display the original and blurred images
figure;
subplot(1, 2, 1);
imshow(image, []);
title('Original Image');

subplot(1, 2, 2);
imshow(blurred_image, []);
```

```
title('Blurred Image');

```