Lanczos Resizing and Rotation for Image Transformation

Kunovics Dávid Zoltán

December 16, 2024

1 Introduction

Image resizing and rotation are fundamental operations in image processing, commonly used in computer vision and graphics. Achieving high-quality visual results while performing these transformations simultaneously poses a significant challenge. This document describes a method implemented in MATLAB, combining Lanczos resampling for resizing and rotation using a 2D transformation matrix. The method ensures proper scaling and alignment by calculating the bounding box for the transformed image and interpolating pixel values with high accuracy.

2 Mathematical Model

2.1 Lanczos Resampling

Lanczos resampling is a high-quality interpolation method based on the sinc function, defined as:

$$\operatorname{sinc}(x) = \begin{cases} 1 & \text{if } x = 0 \\ \frac{\sin(\pi x)}{\pi x} & \text{if } x \neq 0 \end{cases}$$

The Lanczos kernel is constructed by multiplying two sinc functions:

$$w(x) = \operatorname{sinc}(x) \cdot \operatorname{sinc}\left(\frac{x}{A}\right)$$

where A is the kernel size parameter, controlling the range of influence of each pixel. Larger values of A yield smoother results but increase computational complexity. The kernel is truncated to A to make the computation practical:

$$w(x) = 0 \quad \text{for } |x| \geq A$$

2.2 Rotation Transformation

To rotate an image by an angle θ, we use the 2D rotation matrix:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x - c_x \\ y - c_y \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \end{bmatrix}$$

where (c_x, c_y) is the image center. To ensure the rotated image fits within the output frame, the bounding box of the scaled and rotated image is calculated by transforming the corners of the scaled input image. The dimensions of this bounding box determine the size of the destination image.

3 Methodology

The method integrates resizing and rotation into a single sampling process using Lanczos resampling. The steps are as follows:

1. **Scaling:** The source image dimensions are scaled by the given factor.
2. **Bounding Box Calculation:** The corners of the scaled image are rotated to determine the size of the output image, ensuring the entire rotated image fits within the frame.
3. **Inverse Mapping:** For each pixel in the output image, its coordinates are mapped back to the input image using the inverse rotation and scaling transformations.
4. **Lanczos Interpolation:** The surrounding pixels in the input image are weighted using the Lanczos kernel, summing their contributions to compute the final pixel value.

This approach ensures high-quality results by precisely mapping and interpolating pixel values.

4 Test Cases and Results

The function was tested with the following configurations:

1. **Test Case 1:** Rescale by a factor of 4 with no rotation (0 degrees).
2. **Test Case 2:** Rescale by a factor of 2 and rotate by 45 degrees.
3. **Test Case 3:** Downscale by a factor of 0.5 and rotate by 90 degrees.

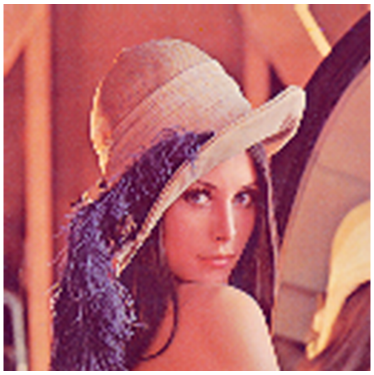

Figure 1: Test Case 1: Rescaled by a factor of 4 with no rotation.

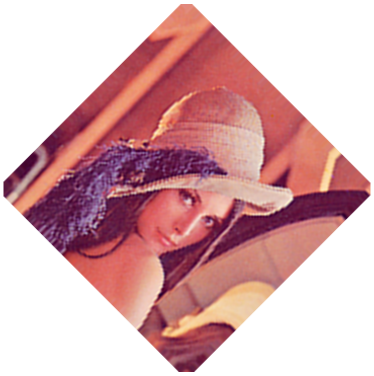

Figure 2: Test Case 2: Rescaled by a factor of 2 and rotated by 45 degrees.

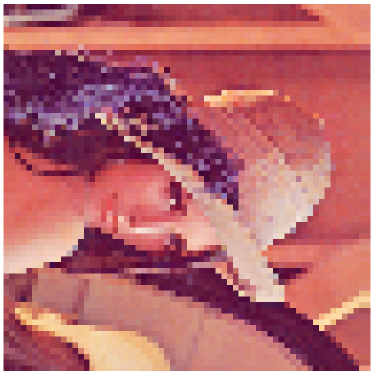

Figure 3: Test Case 3: Downscaled by a factor of 0.5 and rotated by 90 degrees.

5 Conclusion

The described MATLAB implementation combines Lanczos resampling with inverse mapping and bounding box adjustments to perform high-quality resizing and rotation. The method ensures that the entire image fits within the output frame, preserving image details and achieving visually accurate transformations. The test cases demonstrate the method's reliability across a range of scales and rotation angles.