Project 2

Part 1

Homography example: ‘Hard to see rug pattern made clear’

Original

Rectified

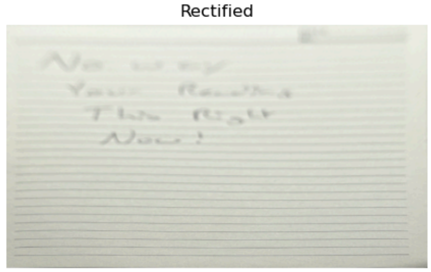
Homography allows us to find the projective transformation between two views of the same planar surface. In this part, I manually selected four rectangular point correspondences on the perspective image of the rug and estimated H using OpenCV’s homography method which utilizes the Direct Linear Transform (DLT) algorithm. Also used openCV’s RANSAC function to reduce the effect of point inaccuracy. The mapping was then applied using OpenCV’s warpPerspective, which performs inverse mapping to resample the source image and produce the fronto-parallel, rectified view. This image is on the right, labeled rectified.

Part 2

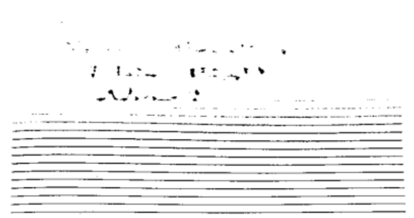
Creative Example: ‘Important step before scanning a paper as demonstrated below’

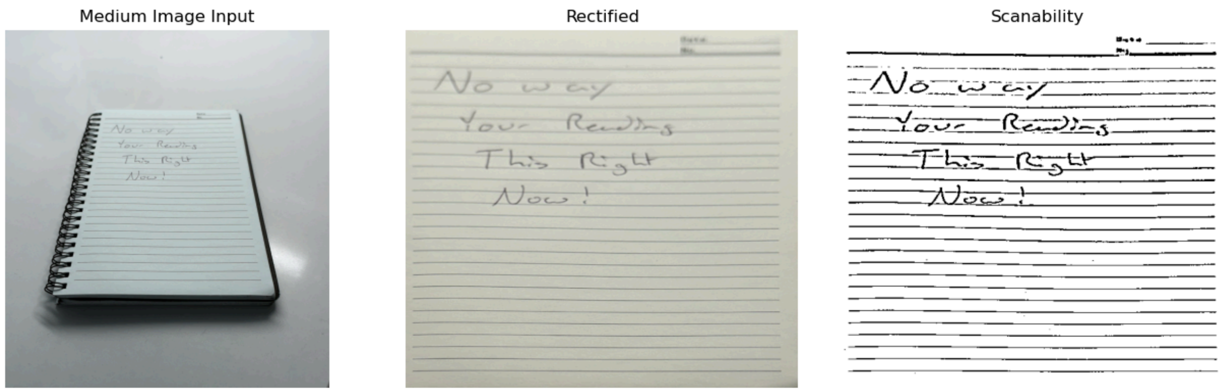
Hard Image Input

Rectified



Scanability



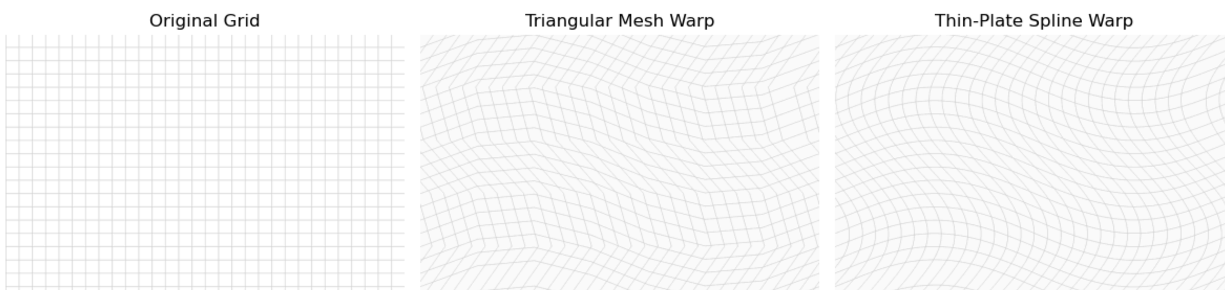


For my creative application, I used the homography implementation to pre-process documents before scanning them in. The goal was to test how much perspective distortion my homography could handle before the results became unusable. To do this, I captured two photos—a “medium” test with moderate tilt and a “hard” test taken from a sharper angle. After applying the same four-point correspondence and warping process, the rectified medium image produced a clean, nearly front-facing result that would be easy to scan or classify text. The hard image was more challenging because of the stronger perspective and small errors in point selection, but it still improved noticeably compared to the raw input. Overall, this showed that the implementation can effectively handle typical document photos and correct them well enough for scanning or digital archiving.

Part 3

Triangular Mesh Warping vs TPS Warping

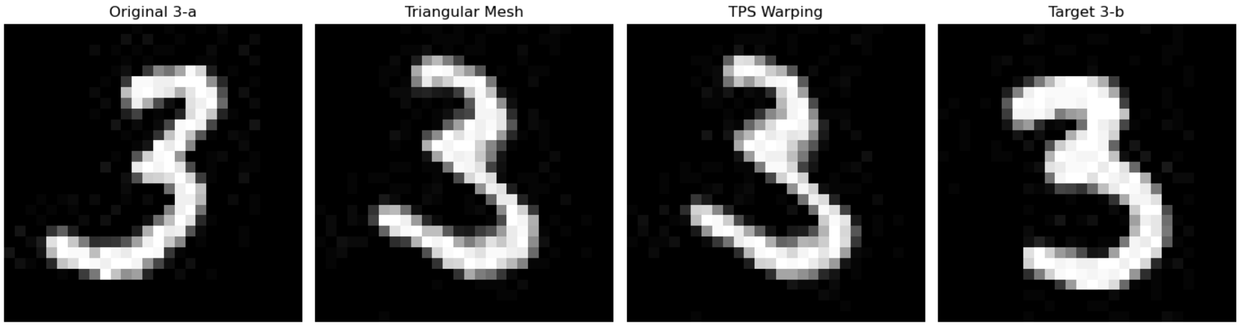
Warp strength (0.1) Low



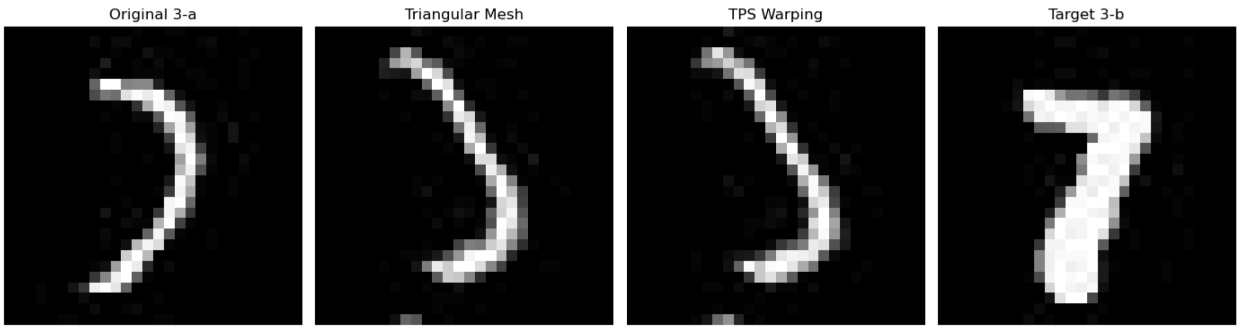
Warp Strength (0.25) High

'Triangle Mesh much more jagged compared to smoother TPS’

Warp Strength (0.1)



Warp Strength (0.1) ‘neither warping technique were very helpful for the 7’



The results show that the two warping methods behave very differently. The **piecewise-affine (triangular mesh)** approach bends the grid in straight segments, so the image is continuous but you can sometimes see faint creases where the triangles meet. Each triangle moves independently, which makes this method fast and gives you good local control—changing one point mostly affects the area around it. In contrast, the **thin-plate spline (TPS)** warp produces a much smoother and more natural-looking deformation because it spreads the motion evenly across the whole image. However, it also means that moving one control point slightly influences everything else, and it’s a bit slower to compute since it solves a larger system. In short, piecewise-affine warping is better for quick, local adjustments, while TPS is better for smooth, continuous distortions that look more organic.