Hough Transform Experiment

Garrett London and Jaydin Andrews

University of North Carolina at Asheville

CSCI 412: Computer Vision

Dr. Mec

24 October 2022

Background

The Hough Transform is an algorithm designed to detect complex lines in photographs (Hough, 1962). Generally, in the x-y space, a line is defined in slope intercept form $y=mx+b$ where the slope m and the y-intercept b are constants and the points x and y are variables. Detecting vertical lines in this image space is impossible since the slope of a vertical line would approach infinity/be undefined which is mathematically and programictly impossible to solve (Leavers, 1992). To solve this issue, a straight line can be represented by another line, often called the normal line, that passes through the origin and is perpendicular to the target line. To do this, the x-y space parameters of the line must be converted to their corresponding polar representation in the Hough Space. In the Hough Space a line is represented by $\rho = x \cos(\theta) + y \sin(\theta)$ where ρ is the length of the normal line and θ is the angle between the normal line and the x axis. Note: when transforming our equations into a ρ-θ polar space, curved lines such as circles and ellipses can also be represented (Smith, 2018). With this in mind, when mapping edge points in the Hough Space a cosine curve is generated rather than a straight line. In this normal representation, there is no unbounded value of m in the x-y space that comes with vertical lines that cannot be represented in the Hough Space. Since one edge point produces one cosine curve when transformed into the Hough Space, a lot of curves will be generated. Cosine curves are periodic, meaning that two points that lay one the same line, their cosine curves will intersect. All intersections in the ρ-θ parameter space are counted and if a (ρ, θ) pair has a number of intersections above a specified threshold, a line is likely to exist with these parameters. To find those lines the algorithm loops over all points in the edge image, calculates all possibilities of θ and its corresponding ρ, and counts the number of times each (ρ, θ) occurs. Finally, it will loop over all those (ρ, θ) pairs and if its count is larger than a specified threshold.

Parameters of the Experiment

The designed experiment measures "performance" of images when processed by the Hough Transform function. One control group will be compared to one experimental group, both containing the same ~1000 300px by **X**px images. The control group will consist of images processed through a Canny edge detector and have their features extracted through a Probabilistic Hough Transform algorithm. The experimental group will consist of images processed through a number of preprocessing methods, namely histogram equalization, gaussian blurring, dilation, and erosion. Those pre-processed images will then be processed by the same Canny edge detector and Probabilistic Hough Transform algorithm as the control group. Once both groups have been run through the Canny edge detection and Hough Transform algorithm, a mean square error is calculated from the resulting images and the images will be ranked in two separate lists. Between these two ranked lists of images, the difference between their ranks determines which category they belong to: the top, middle, and bottom performers. The top performers are images that did well raw but didn't do well after the preprocessing. The bottom performers are images that did well raw but failed to detect more edges after preprocessing. The middle performers are images that didn't experience much change between the control and preprocessing experimental groups.

Results

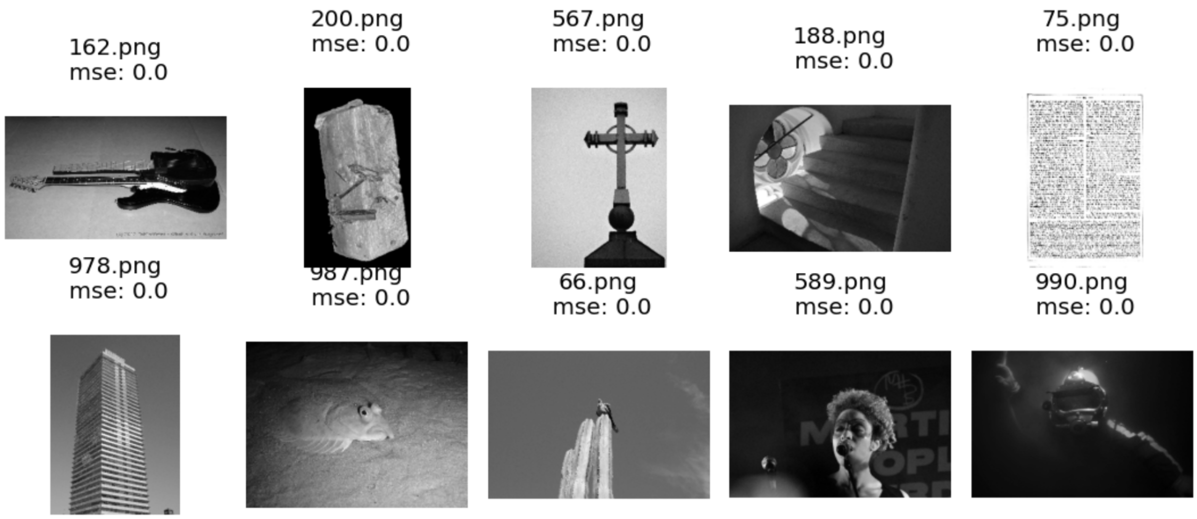

Figure 1.1
Images classified as Top Performers

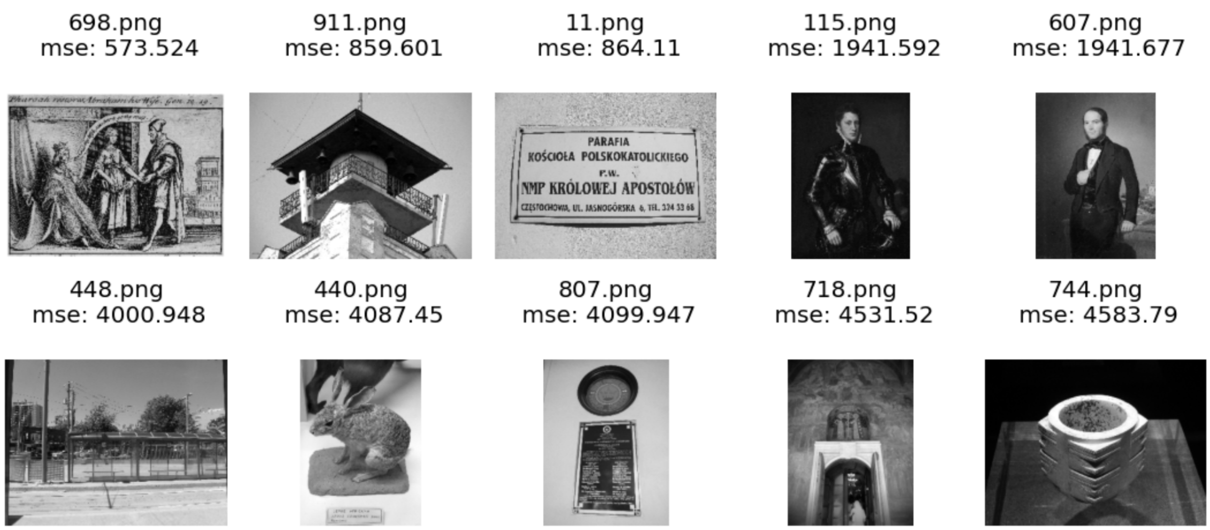

Figure 1.2
Images classified as Mid Performers

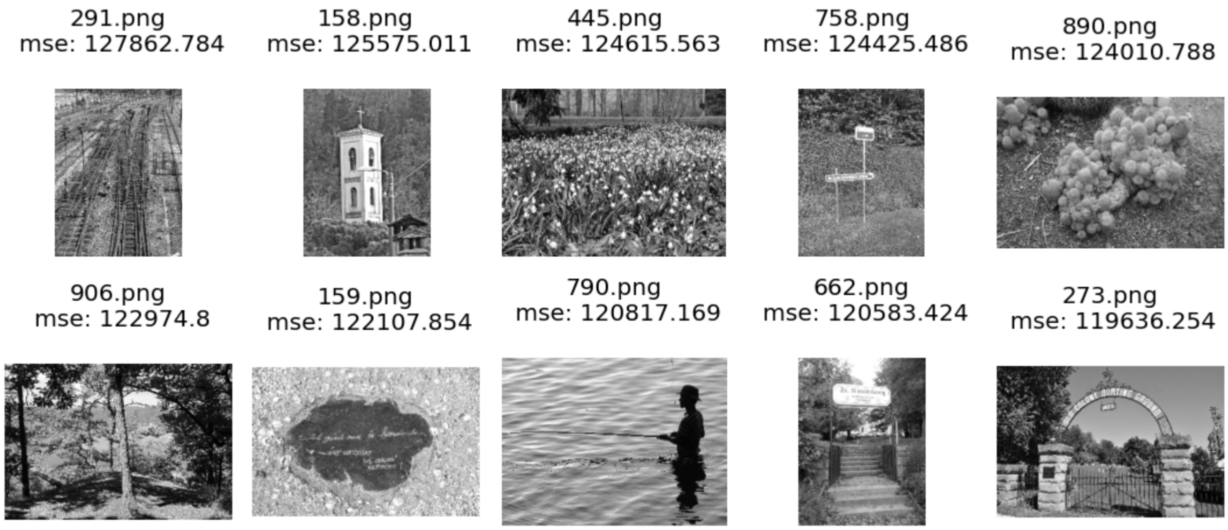

Figure 1.3
Images classified as Poor Performers

Each image belonging to the group of top performers remained ranked comparatively high in both categories. The images belonging to the bottom performers group, ranked low in both categories. Finally, the images in the bottom group were ranked higher in the raw run than in the preprocessing experimental group. As seen in Figures 1.1 through 1.3, the images in each category pass the “eyeball verification test” however the Mean Square Error results for some images are summed to 0.0 while others are in the 100,000’s. This is believed to be related to a scalar overflow bug found late in the development. The interesting thing with this bug is that the images do seem to be sorted and categorized correctly but this is hard to mathematically prove with inconsistent and possibly inaccurate error calculations. From the “eyeball verification test” it is clear to see that the images in the top category tend to be clear images with very little noise. The inverse can be seen in the images categorized in the bottom tier: image backgrounds tend to have more random noise that leads to high error results from the Canny and Probabilistic Hough Transform algorithms.

Reflection

When developing a solution to objectively compare the results of edge detection and edge localization algorithms, ensuring the data stays in a consistent and uniform format is crucial. During the development of this project, several problems faced were rooted in the fact that the outputs of these algorithms are fundamentally different and therefore attempting to compare them in their raw forms was very difficult. While it was relatively easy to decide which images did well in the both categories from an “eyeball verification test”, it is difficult to translate those parameters into an algorithm that a machine can objectively implement.

Conclusion

While the Canny edge detector does a very good job at finding edges and determining contrast, it ends up generating a lot of noise when converting the edge pixels into edges that can be extracted. To solve this problem, a Probabilistic Hough Transform can help localize those edge pixels and filter out a lot of the noise generated by the Canny edge detector. This noise can be further cleaned up with careful preprocessing techniques. The preprocessing pipeline implemented in this project is just one of millions of preprocessing algorithm combinations. In this implementation, the pipeline of repeatedly blurring, dilating and eroding the image data in conjunction with normalizing the color histogram of the original image resulted in a significant gain of the clarity in the detected lines for most images. Some images of course, those in the poor performers category, didn’t show any improvement or in some cases netted a decline in performance.

References

Hough, P. V. C. (1962). Method and means for recognizing complex patterns (U.S. Patent 3069654). Retrieved from https://patentimages.storage.googleapis.com/9f/9f/f3/87610ddec32390/US3069654.pdf

Leavers, V. F. (1992). Preprocessing, Shape Detection in Computer Vision Using Hough Transform (pp. 39–64). doi:10.1007/978–1–4471–1940–1

Smith, B. (2018, September 21). *Hough Circle Transform*. ImageJ. Retrieved from https://imagej.net/Hough_Circle_Transform