CHUN Hiu Sang `hschun`
TSE Ho Nam `hntse`

Homework 1 Write-up

1 Theory questions

Q1.1

1. Given that $\rho = x\cos\theta + y\sin\theta$, let $z = \sqrt{x^2+y^2}$, then $\rho = z\left(\frac{x}{z}\cos\theta + \frac{y}{z}\sin\theta\right)$. Note that in Figure 1, where $x = a$ and $y = b$ so the sloped line has length z, we have $\sin\theta = \frac{x}{z}$ and $\cos\theta = \frac{y}{z}$.

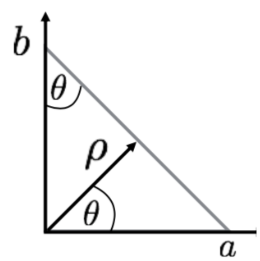


Figure 1: Plot of $\rho = x\cos\theta + y\sin\theta$

$$\rho = z\left(\frac{x}{z}\cos\theta + \frac{y}{z}\sin\theta\right) = z(\sin\theta\cos\theta + \cos\theta\sin\theta) = z\sin 2\theta$$

So when (x, y) are fixed as an image point and we plot ρ against θ, this equation represents a sinusoid in the (ρ, θ) Hough space.

The amplitude is $z = \sqrt{x^2+y^2}$, and the phase is 0.

2. The parameters (m, c) both have the domain of $\mathbb{R}$, which is infinitely large. Since the algorithm runs with a finite accumulator array, it is better to have parameters that have finitely large domains. In particular, to capture vertical lines, we need $m = \pm\infty$. (ρ, θ) do not have these problems so they are chosen as a better parametrization.

$$\begin{aligned} &\rho = x\cos\theta + y\sin\theta \\ \iff\ & y = -\frac{\cos\theta}{\sin\theta}x + \frac{\rho}{\sin\theta} \\ \iff\ & (m, c) = \left(-\frac{\cos\theta}{\sin\theta}, \frac{\rho}{\sin\theta}\right) = \left(-\cot\theta, \frac{\rho}{\sin\theta}\right) \end{aligned}$$

3. From part 1 above, $\rho = z\sin 2\theta$.

$$\max|\rho| = |z\sin 2\theta| = (\max|z|)(1) = \sqrt{W^2+H^2}$$
$$\theta \in [0, 2\pi]$$

4. From Figure 2, $(m, c) = \left(-\cot\frac{3\pi}{4}, \frac{0}{\sin\frac{3\pi}{4}}\right) = (1, 0)$

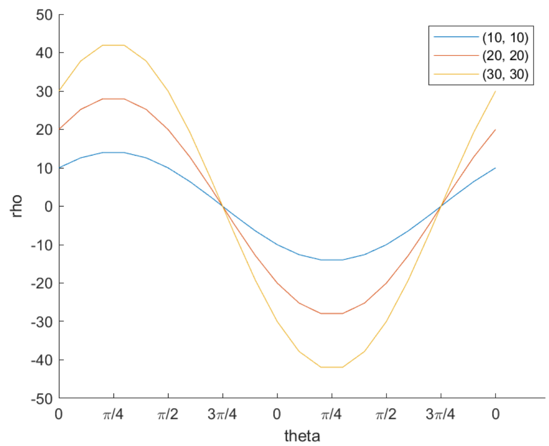


Figure 2: Plots of Sinusoid waves

2 Implementation

Q2.3

Figure 3: Hough Transform result on img03

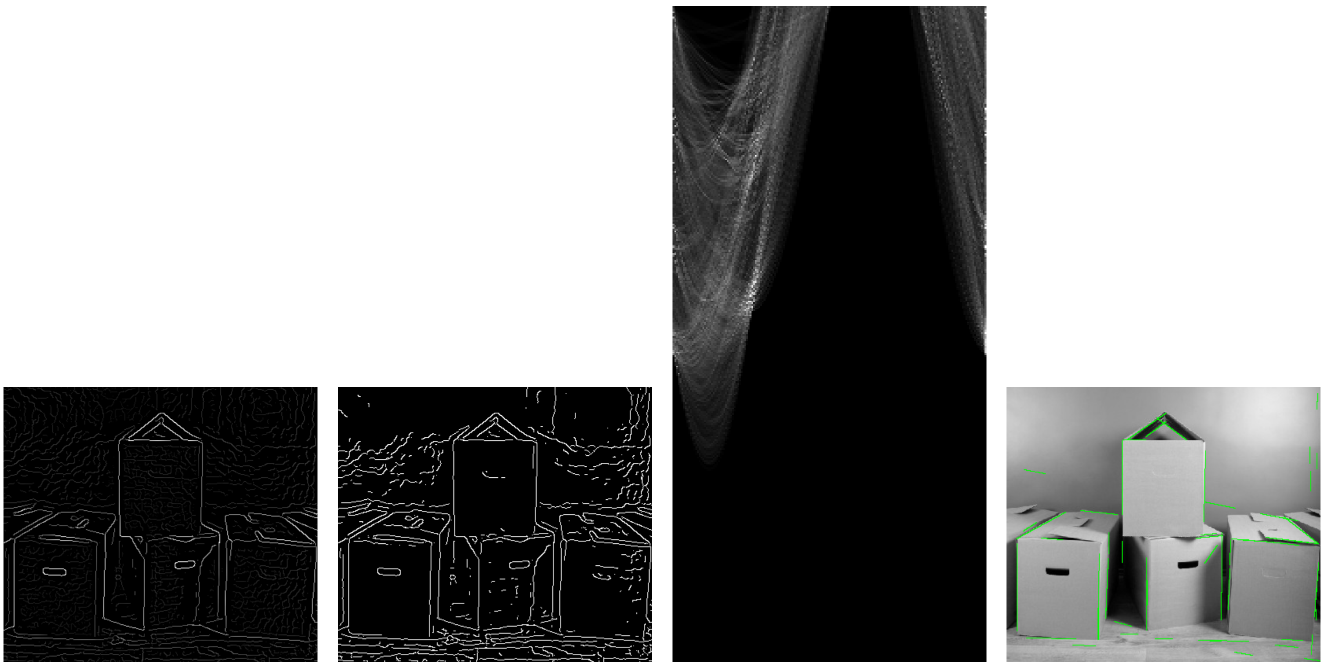
Q2.5x

As shown in Figure 4, the results of our own houghlines function and that of Matlab's built-in function are mostly similarly, but our approach generates more broken lines under the same choice of parameters, and there are also sometimes false positive or false negative lines.

Figure 4: Hough Lines result on img09

3 Experiments

Q3.1



(a) Edges (b) Thresholded edges (c) Hough transform (d) Line segments

Figure 5: Set of intermediate outputs of Hough detector on img05

Our code with default parameters works well in most cases on different images, as the prominent edges are returned accordingly. For particular images, tuning the parameters to adapt to an image – not using a single set of parameters – would give a more satisfying edge detection result, having less noise and more detail.

The optimal set of parameters for images without too much noisy details like small texts and icons would be to decrease `sigma`, and increase `rhoRes`, `thetaRes` and `nLines`. For images that do have small details (e.g. the provided `img06`), higher `threshold` and lower `sigma`, `rhoRes`, `thetaRes` would be better.

Effects of the parameters:

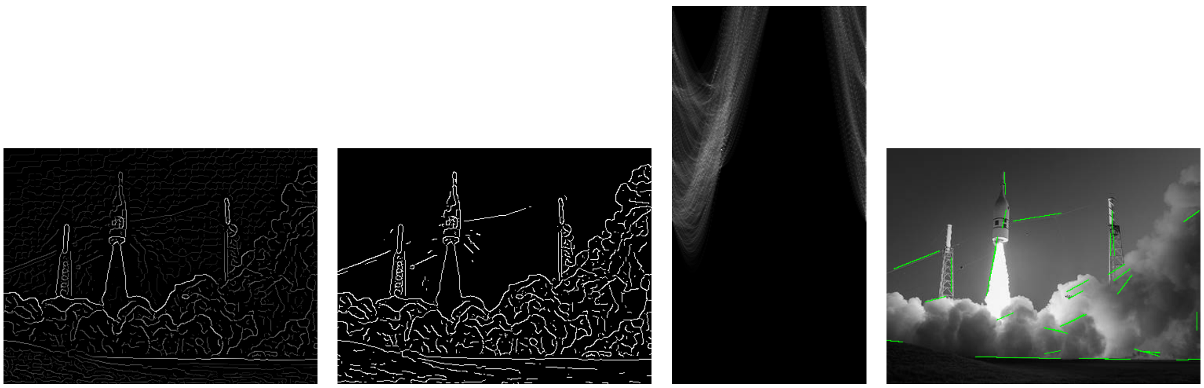
- `sigma`: A smaller `sigma` (=1) results in producing more noisy edge responses and in general more edges are attempted by the Hough detector. A larger `sigma` (=6) results in a more precise and concise set of edges which tend to be valid edges, because in the edges and threshold steps, only prominent edges remain and detailed edge outlines are lost.

- `threshold`: This directly controls the amount of noise in the edge responses. A low `threshold` (=0.01) makes even less likely edges count as edges, a slightly higher one (=0.05) would keep the prominent edges but rule out the others.
- `rhoRes`, `thetaRes`: A lower resolution in `rhoRes` (=1) and `thetaRes` (=$\pi/120$) gives fewer edge responses and tend to miss out many edges too, a higher resolution (`rhoRes`=4 and `thetaRes`=$\pi/60$) gives rise to more discovery of edges with relatively little noise added.
- `nLines`: Having fewer lines (=30) forces the detector to select the edges of the strongest responses only, but would miss details easily. Having more lines (=70) actually do not increase noise a great deal, but reveals more likely edges.

The feature detection step of Hough transform in the algorithm causes the most problems. By increasing the resolution of ρ and θ, better discrete approximations can be reached, thus giving more precise edge detection results.

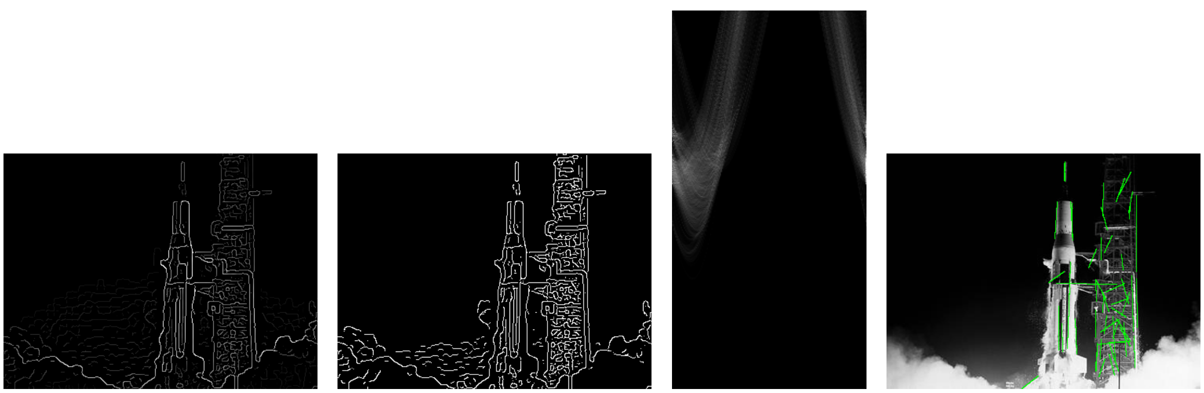
4 Try your own images!

Q4.1x



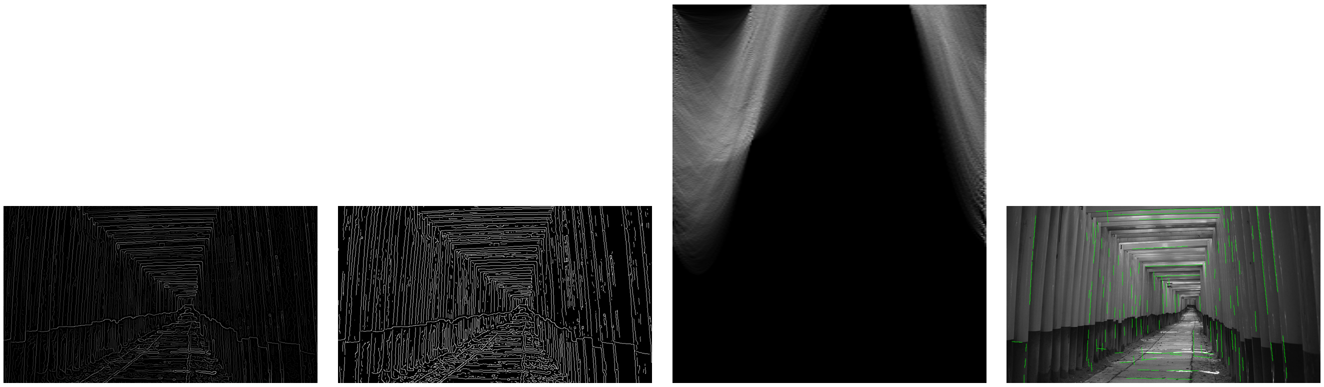
(a) Edges (b) Thresholded edges (c) Hough transform (d) Line segments

Figure 6: Set of intermediate outputs of Hough detector on myimg01



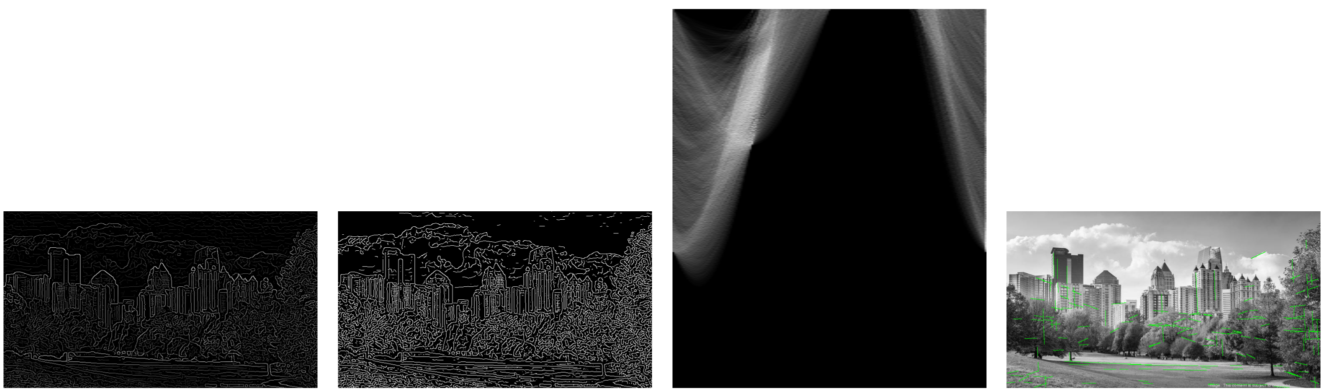
(a) Edges (b) Thresholded edges (c) Hough transform (d) Line segments

Figure 7: Set of intermediate outputs of Hough detector on myimg02



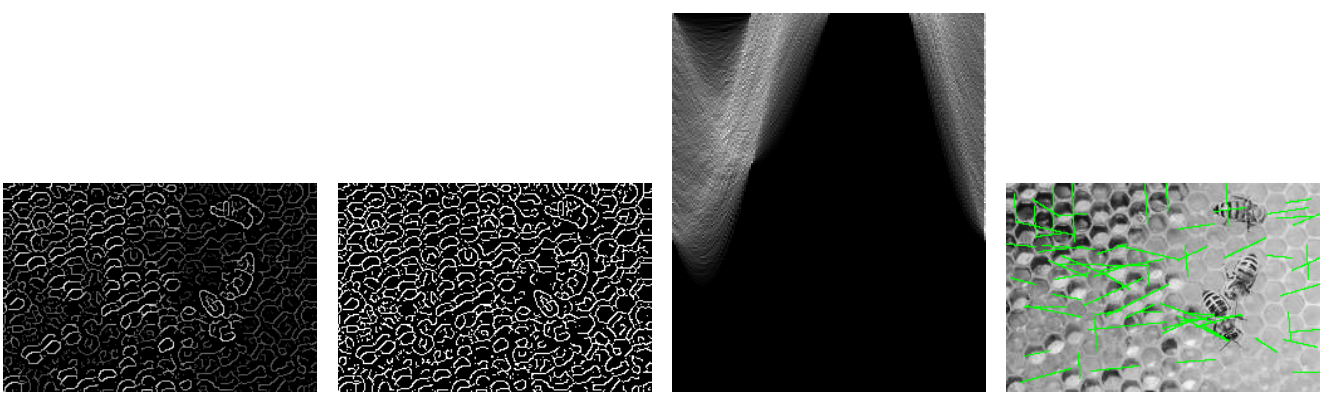
(a) Edges (b) Thresholded edges (c) Hough transform (d) Line segments

Figure 8: Set of intermediate outputs of Hough detector on myimg03



(a) Edges (b) Thresholded edges (c) Hough transform (d) Line segments

Figure 9: Set of intermediate outputs of Hough detector on myimg04



(a) Edges (b) Thresholded edges (c) Hough transform (d) Line segments

Figure 10: Set of intermediate outputs of Hough detector on myimg05