# COMP 478/6771 Image Processing Assignment

Qingbo Kang, Concordia University 40058122

## 1 Eq.(10.3.15)

The Eq.(10.3.15) has two lines. I will derive two lines separately.

### 1.1 $\sigma_B^2 = P_1P_2(m_1 - m_2)^2$ derivation

From Eq.(10.3.14), Eq.(10.3.10), Eq.(10.3.11):

$$
\begin{aligned}
\sigma_B^2 &= P_1(m_1 - m_G)^2 + P_2(m_2 - m_G)^2 \\
&= P_1[m_1 - (P_1m_1 + P_2m_2)]^2 + P_2[m_2 - (P_1m_1 + P_2m_2)]^2 \\
&= P_1(m_1 - P_1m_1 - P_2m_2)^2 + P_2(m_2 - P_1m_1 - P_2m_2)^2 \\
&= P_1[m_1 - (1 - P_2)m_1 - P_2m_2]^2 + P_2[m_2 - P_1m_1 - (1 - P_1)m_2]^2 \\
&= P_1(m_1 - m_1 + P_2m_1 - P_2m_2)^2 + P_2(m_2 - P_1m_1 - m_2 + P_1m_2)^2 \\
&= P_1(P_2m_1 - P_2m_2)^2 + P_2(P_1m_2 - P_1m_1)^2 \\
&= P_1[P_2(m_1 - m_2)]^2 + P_2[P_1(m_2 - m_1)]^2 \\
&= P_1P_2^2(m_1 - m_2)^2 + P_2P_1^2(m_1 - m_2)^2 \\
&= (P_1P_2^2 + P_2P_1^2)(m_1 - m_2)^2 \\
&= P_1P_2(P_2 + P_1)(m_1 - m_2)^2 \\
&= P_1P_2(m_1 - m_2)^2
\end{aligned}
$$

### 1.2 $\sigma_B^2 = \frac{(m_GP_1 - m)^2}{P_1(1-P_1)}$ derivation

From Eq.(10.3.5) - Eq.(10.3.9):

$$
\begin{aligned}
m_2(k) &= \frac{1}{P_2(k)} \sum_{i=k+1}^{L-1} ip_i \\
&= \frac{1}{1 - P_1(k)} \sum_{i=k+1}^{L-1} ip_i \\
&= \frac{1}{1 - P_1(k)} [\sum_{i=0}^{L-1} ip_i - \sum_{i=0}^{k} ip_i] \\
&= \frac{1}{1 - P_1(k)} (m_G - m(k))
\end{aligned}
$$

Then we have:

$$
\begin{aligned}
\sigma_B^2 &= P_1P_2(m_1 - m_2)^2 \\
&= P_1P_2(\frac{m}{P_1} - \frac{m_G - m}{1 - P_1})^2 \\
&= P_1(1 - P_1)[\frac{m(1 - P_1) - P_1(m_G - m)}{P_1(1 - P_1)}]^2 \\
&= P_1(1 - P_1)[\frac{m - P_1m_G}{P_1(1 - P_1)}]^2 \\
&= \frac{(m_GP_1 - m)^2}{P_1(1 - P_1)}
\end{aligned}
$$

# 2 Hough Transform

## 2.1

Because in the Cartesian $(x, y)$ coordinate system, the general equation of a straight line in slope-intercept form is:

$$y = ax + b$$

And actually in principle the $ab$-plane (or $parameter\ space$) lines corresponding to all points $(x_k, y_k)$ in the Cartesian $(x, y)$ coordinate system. But a practical difficulty with this approach is that $a$ (the slope of a line) approaches infinity as the line approaches the vertical direction. So this is the reason that Hough transform for lines uses the normal representation of a line and the $\rho\theta$ parameter space, and cannot be carried out in the Cartesian $(x, y)$ coordinate system.

## 2.2

1. For a set of $n$ points $(x_1, y_1)$, $(x_2, y_2)$, ..., $(x_n, y_n)$. Let $(\rho_{\min}, \rho_{\max})$ and $(\theta_{\min}, \theta_{\max})$ are the expected ranges of the parameter values. $-D \leq \rho \leq D$ and $-90° \leq \theta \leq 90°$ where $D$ is the maximum distance between opposite corners in the image.
2. Every cell at $(i, j)$ has its accumulator value $A(i, j)$ which is corresponds to the square associated with parameter-space coordinates $(\rho_i, \theta_j)$. At beginning, set these values to zero.
3. Then, for every point $(x_k, y_k)$ in the $xy$-plane, let $\theta$ equal each of the allowed subdivision values on the $\theta$-axis and solve for the corresponding $\rho$ using the equation $\rho = x_k \cos\theta + y_k \sin\theta$.
4. The resulting $\rho$ values are then rounded off to the nearest allowed cell value along the $\rho$ axis. If a choice of $\theta_p$ results in solution $\rho_q$, then let $A(p, q) = A(p, q) + 1$.
5. At last, a value of $P$ in $A(i, j)$ means that $P$ points in the $xy$-plane lie on the line $x \cos\theta_j + y \sin\theta_j = \rho_i$.

# 3 Textbook Problem 10.24

For every non-background point $(x_k, y_k)$ in the $xy$-plane, where $k$ is the number of subdividing the $\theta$-axis, in other words, $\theta$ has $k$ possible values. Hough transform for lines in the parameter space has $nk$ operations. Therefore, the number of operations $nk$ required to implement the accumulator cell approach is linear in $n$.

# 4 Programming - Otsu

## 4.1 a

The code is implemented in Solution_4.m.
Figure 1 shows the result images.

## 4.2 b

The code is implemented in Solution_4.m.
Figure 2 shows the result images.

# 5 Programming - Wavelet

## 5.1 a

The code is implemented in Solution_5.m.

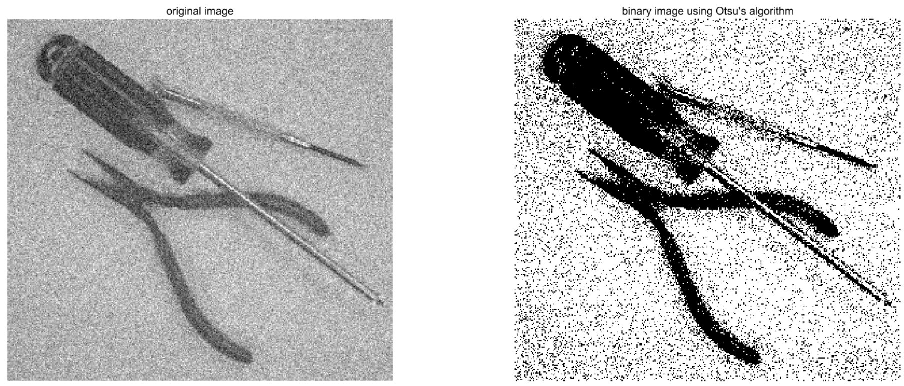



Figure 1: Result images of 4.a

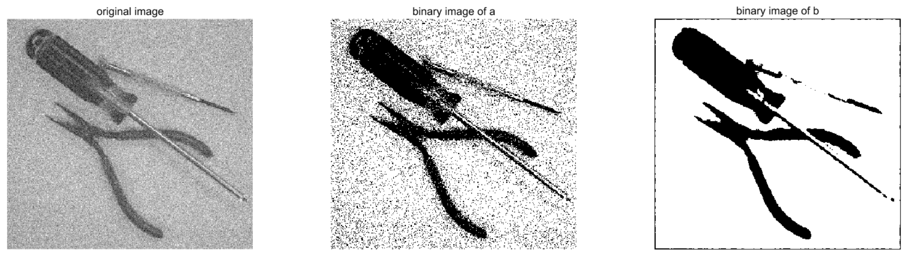


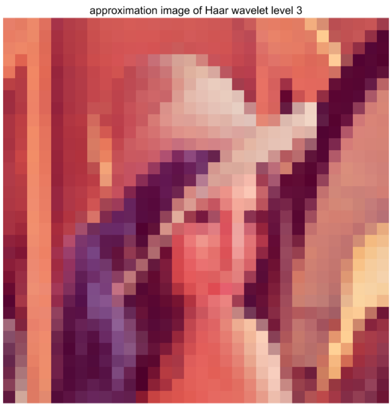

Figure 2: Result images of 4.b

### 5.2 b

The code is implemented in Solution_5.m.

### 5.3 c

The code is implemented in Solution_5.m.
Figure 3 shows the result images.



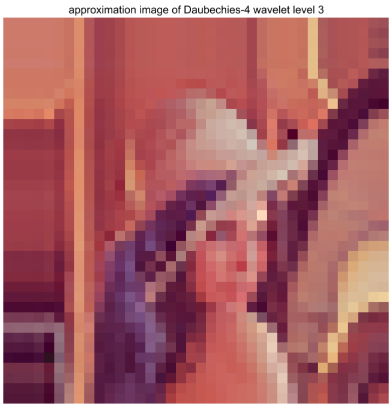




Figure 3: Result images of 5.c