ECE 276 Project 1

Xin Li

A91068482

Hand-Written:

Problem 1

(a) $p(u) = \begin{cases} 1 & 0 < u < 1 \\ 0 & , \text{else} \end{cases}$ $\quad p(v) = \begin{cases} 1 & , 0 < v < 1 \\ 0 & , \text{else} \end{cases}$

$P(R \le r) = P(\sqrt{-2\log(1-u)} \le r)$

$= P(u \le 1 - e^{-\frac{1}{2}r^2})$

$D(R) = \int_{-\infty}^{1-e^{-\frac{1}{2}r^2}} 1 \; du$

In this case since $\sqrt{-2\log(1-u)} \ge 0$

$-2\log(1-u) \ge 0$

$\log(1-u) \le 0$

$0 < 1-u \le 1$

$0 \le u < 1$

$1 - e^{-\frac{1}{2}r^2} = \begin{cases} 0 & \text{when } r = 0 \\ \uparrow & \text{when } |r| \uparrow \end{cases}$

as $0 \le 1 - e^{-\frac{1}{2}r^2} < 1$

so $D(R \le r) = \int_0^{1-e^{-\frac{1}{2}r^2}} 1 \, du = \begin{cases} 0 & r = 0 \\ 1 - e^{-\frac{1}{2}r^2} & r \ne 0 \end{cases}$

$P(\theta \le \theta) = P(2\pi V \le \theta)$

$= P(V \le \frac{\theta}{2\pi})$

$D(\theta) = \int_{-\infty}^{\frac{\theta}{2\pi}} 1 \; dv$

$= \int_0^{\frac{\theta}{2\pi}} 1 \, dv$

$D(\theta \le \theta) = \begin{cases} 0 & , \theta < 0 \\ \frac{\theta}{2\pi} & , 0 \le \theta \le 2\pi \\ 1 & , \theta > 2\pi \end{cases}$

(b) $(Y, X) = f(\theta, R)$ $\quad$ for joint distribution

$f\begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} R\cos\theta \\ R\sin\theta \end{pmatrix}$ $\quad \frac{df}{dx}(f^{-1}y) = \begin{pmatrix} \cos\theta & -R\sin\theta \\ \sin\theta & R\cos\theta \end{pmatrix}$

$\det(f^{-1}y) = R\cos^2\theta + R\sin^2\theta = R = \sqrt{X^2+Y^2}$

$P_{X,Y} = \frac{1}{R} P_{R,\theta}(f^{-1}(x, y)) =$

$P(\theta, R)$ has joint pdf $\quad P(\theta, R) = \begin{cases} (1-e^{-\frac{1}{2}r^2}) \cdot \frac{\theta}{2\pi} & 0 \le \theta < 2\pi, \; r \ne 0 \\ 1 - e^{-\frac{1}{2}r^2} & \theta > 2\pi; \; r \ne 0 \\ 0 & \text{else} \end{cases}$

$\theta = \tan^{-1}(\frac{Y}{X})$

$R = \sqrt{X^2+Y^2}$

$$P(x, y) = \begin{cases} \frac{1}{\sqrt{X^2+Y^2}} (1 - e^{-\frac{X^2+Y^2}{2}}) \cdot \frac{\tan^{-1}(\frac{Y}{X})}{2\pi} & , 0 \le \tan^{-1}(\frac{Y}{X}) \le 2\pi, \; \sqrt{X^2+Y^2} \ne 0 \\ \frac{1}{\sqrt{X^2+Y^2}} (1 - e^{-\frac{X^2+Y^2}{2}}) & , \tan^{-1}(\frac{Y}{X}) > 2\pi, \; \sqrt{X^2+Y^2} \ne 0 \\ 0 & , \text{else} \end{cases}$$

Problem 2.

(a) $E[(x-x')^T(x-x') \mid (Y=1, Y'=1)]$

$= E[xx] - 2E[xx'] + E[x'x']$

$= E[x^2] - 2E[xx'] + E[x'^2]$

$Var[X] = E[(X-\mu_1)^2]$

$= E[X^2] - 2E[X]E[\mu_1] + E[\mu_1^2]$

$E[X^2] = Var[X] - 2E[X]\mu_1 + \mu_1^2$ $\quad E[X] = \mu_1$ for Gaussian distribution

$= \sigma^2 - 2\mu_1^2 + \mu_1^2$

$= \sigma^2 - \mu_1^2$

similarly $E[X'^2] = \sigma^2 - \mu_1^2$

$E[XX'] = E[X]E[X'] = \mu_1^2$

So $E[(x-x')^T(x-x') \mid (Y=1, Y'=1)] = (\sigma^2 - \mu_1^2) - 2\mu_1^2 + (\sigma^2 - \mu_1^2)$

$= 2\sigma^2 - 4\mu_1^2$

(b) $E[(x-x')^T(x-x') \mid (Y=1, Y'=2)]$

$= E[xx] - 2E[xx'] + E[x'x']$

similar to above,

$= (\sigma^2 - \mu_1^2) - 2\mu_1\mu_2 + (\sigma^2 - \mu_2^2)$

$= 2\sigma^2 - \mu_1^2 - 2\mu_1\mu_2 - \mu_2^2$

c) $R = \frac{E[\text{intra}]}{E[\text{inter}]} = \frac{2\sigma^2 - 4\mu_1^2}{2\sigma^2 - (\mu_1+\mu_2)^2} \Rightarrow \frac{2\sigma^2 - 4\mu_{1j}^2}{2\sigma^2 - (\mu_{1j}+\mu_{2j})^2}$

as m becomes larger

$R \Rightarrow \frac{4\mu_{1j}^2}{(\mu_{1j}+\mu_{2j})^2} \Rightarrow \frac{4\mu_{1j}^2}{(2\mu_{1j})^2} \Rightarrow 1$

which accuracy approches 1.

3.

(a) $p(x;\lambda) = \frac{\lambda^x e^{-\lambda}}{x!}$ $\lambda$ event per interval $\lambda$ mean

$\hat{\lambda} = \frac{1}{n}\sum_{i=1}^n X_i$ is MLE of $\lambda$

$$E(X) = \sum_{x=0}^{\infty} x \frac{e^{-\lambda}\lambda^x}{x!}$$

$$= \sum_{x=1}^{\infty} x \frac{e^{-\lambda}\lambda^x}{x!}$$

$$= \sum_{x=1}^{\infty} \frac{e^{-\lambda}\lambda^x}{(x-1)!}$$

$$= \lambda e^{-\lambda} \sum_{x=1}^{\infty} \frac{\lambda^{x-1}}{(x-1)!}$$

$$= \lambda e^{-\lambda} \left(\frac{\lambda^0}{0!} + \frac{\lambda^1}{1!} + \frac{\lambda^2}{2!} + \cdots\right)$$

$$= \lambda e^{-\lambda} \sum_{x=0}^{\infty} \frac{\lambda^x}{x!}$$

$$= \lambda e^{-\lambda} e^{\lambda}$$

$= \lambda$ which expected value of $E[X]$

since $\lambda$ is average $\hat{\lambda} = \frac{1}{n}\sum_{i=1}^n X_i$

and hence $E[\hat{\lambda}] = \lambda$

(b) prove. $P(\lambda|\alpha,\beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda^{\alpha-1} e^{-\beta\lambda}$, $\lambda > 0$

from (a)

$f(x|\lambda) = P(X|\lambda) = \prod_{i=1}^{n} \frac{\lambda^{x_i} e^{-\lambda}}{x_i!}$ for posterior distribution of $X_1 = x_1 \cdots, X_n = x_n$

$$f_{\lambda|X}(\lambda | x_1 \cdots x_n) = \frac{f_{\lambda,X}(\lambda, x_1 \cdots x_n)}{f_X(X)} = \frac{f(x|\lambda) f_\lambda(\lambda)}{\sum_{\lambda_i} f(x|\lambda_i) f_\lambda(\lambda_i)}$$

since $\lambda$ has mean $\alpha/\beta$, mode $(\alpha-1)/\beta$

$$f_{\lambda|X}(\lambda|x_1 \cdots x_n) \propto \prod_{i=1}^{n} \frac{\lambda^{x_i} e^{-\lambda}}{x_i!} \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda^{\alpha-1} e^{-\beta\lambda} \propto \frac{\lambda^{\sum x_i + \alpha - 1} e^{-(n+\beta)\lambda} \beta^\alpha}{\prod_{i=1}^{n} x_i! \cdot \Gamma(\alpha)}$$

which is in form of gamma distribution

$\sum_{\lambda_i} f(x|\lambda_i) f_\lambda(\lambda_i)$ In this case equals to constant let it $= S$

$$\text{so } f_{\lambda|X}(\lambda|x_1 \cdots x_n) = \frac{\lambda^{\sum x_i + \alpha - 1} e^{-(n+\beta)\lambda} \beta^\alpha}{S \prod_{i=1}^{n} x_i! \; \Gamma(\alpha)}$$

(c) MAP. as mode has its highest probability

$$\Gamma(\alpha,\beta) = \frac{\alpha-1}{\beta} \quad \text{as } \alpha > 1$$

$$\text{mode} = \frac{\alpha-1}{\beta} = \frac{\sum_i x_i + \alpha - 1}{N + \beta}$$

d) Set $X$ distribution parameter to $\lambda(\eta)$.

$e^{-2\lambda} \to \eta \qquad \lambda(\eta) = -\frac{1}{2}\ln\eta$

$$P(X|\lambda(\eta)) = \frac{\lambda(\eta)^X e^{-\lambda(\eta)}}{X!} = \frac{1}{X!}\left(-\frac{1}{2}\ln\eta\right)^X e^{\frac{1}{2}\ln\eta}$$

$$\arg_{max} P(X|\lambda(\eta)) = \arg_{max} \log P(X|\lambda(\eta))$$

$$\frac{d \log P(x|\lambda(\eta))}{d\eta} = 0$$

$$X \frac{1}{-\frac{1}{2}\ln\hat\eta} \cdot \frac{-1}{2\hat\eta} + \frac{1}{2\hat\eta} = 0$$

$$\frac{X}{\eta \ln\hat\eta} = -\frac{1}{2\hat\eta}$$

$$-2X = \ln\hat\eta$$

$$\hat\eta = e^{-2X}$$

(e) $E[\hat{\eta}] = \sum_{x_i=0}^{\infty} \hat{\eta} P(X_i)$

$= \sum_{x_i=0}^{\infty} e^{-2x} \frac{\lambda^{x_i} e^{-\lambda}}{x_i!}$

$= e^{-\lambda} \sum_{i=0}^{\infty} \frac{(\lambda e^{-2})^{x_i}}{x_i!}$

$= e^{-\lambda} e^{\lambda e^{-2}}$

$= e^{(e^{-2}-1)\lambda}$

bias $= e^{(e^{-2}-1)\lambda} - e^{-2\lambda}$

(f) $E[\hat{\eta}] = \sum_{i=0}^{\infty} \hat{\eta} P(X_i)$

$= \sum_{i=0}^{\infty} (-1)^{x_i} \frac{\lambda^{x_i} e^{-\lambda}}{x_i!}$

$= e^{-\lambda} \sum_{i=0}^{\infty} \frac{(-\lambda)^{x_i}}{x_i!}$

$= e^{-2\lambda} = \eta$

which bias is 0

if $(-1)^x$ varies from negative to positive depending on $x$ value

Introduction:

Object detection is a good practice in the computer vision realm. After training some data, we could do color segmentation and distance estimation with a high accuracy, and such approach helps with solving some real-life problems such as automated driving and robots exploring paths.

In this project, we do color segmentation on images by constructing single gaussian distribution to measure the likelihood of each color; besides, we utilize linear regression to measure distance of barrel to the camera with provided measured width and height of barrels as well as distance of barrels from the camera.

Problem Formulation:

Firstly, we collect data samples of different colors (barrel red, non-barrel red, yellow, brown) using roipoly functions. After calculating the mean and covariance of collected 3-dimensional value for different classes, we could reconstruct single gaussian distribution which provides us the estimated 3-dimension value of a pixel given its class. Then, according to Bayes Formula, p(x|y) = p(y|x)*p(y), we could calculate the probability of a pixel belong to each class given its color value.

For the distance estimation, by measuring the width and length of barrel which we create a (n*2) matrix(n = 37, I extracted samples from 37 images). As the locations of barrels are different, I add a bias term which make X to (n*3) matrix. Meanwhile I create the distance of barrel to camera a (n*1) vector. We could derive the MLE of weight to connect these relate these two terms using linear regression, V is n*n identity matrix

$\omega_{MLE} = (X^T V^{-1} X)^{-1} X^T V^{-1} \mathbf{y}$ , which is a (3*1) vector

Technical Approach:

For the color segmentation, I've trained 4 different classes, barrel-red (red on the barrel), non-barrel red (red not from barrel), yellow (some light color, white& yellow), and brown (some dark color, brown& black). In this way, I can detect barrel red with a high accuracy. When I read a new image, and classify it into 4 classes, I do resample 4-class matrix into binary by setting barrel-red to 1 and others to 0. With such binary matrix, I can use minAreaRect to find the area of each contour. By setting a threshold to the area of each rectangle(1000), and the ratio between contour area and rectangle area(0.75) barrels can be classified from image with a high accuracy.

Results:

The barrel and bound detection works for most of the barrels in the images.

In terms of testing, when there exists barrel red connected with the detected barrel part, the model will overestimate the barrel red parts. Sometimes, such the irregular pattern connected with the barrel will cause a low barrel score which makes barrel undetected. For examples, in test img 003 barrels are not detected because of the red color of cola machine. Also, for barrels getting blocked, this leads to smaller bound (smaller height& width) detected for those barrels. Such deficiency will cause inaccurate distance estimation.

In terms of some barrels which are separated by objects like stairs, to maintain the accuracy for distance estimation. After detecting multiple parts with barrelness>0.75, my algorithm creates a combined bound for these multiply parts, which can maintain the height and width of barrels.

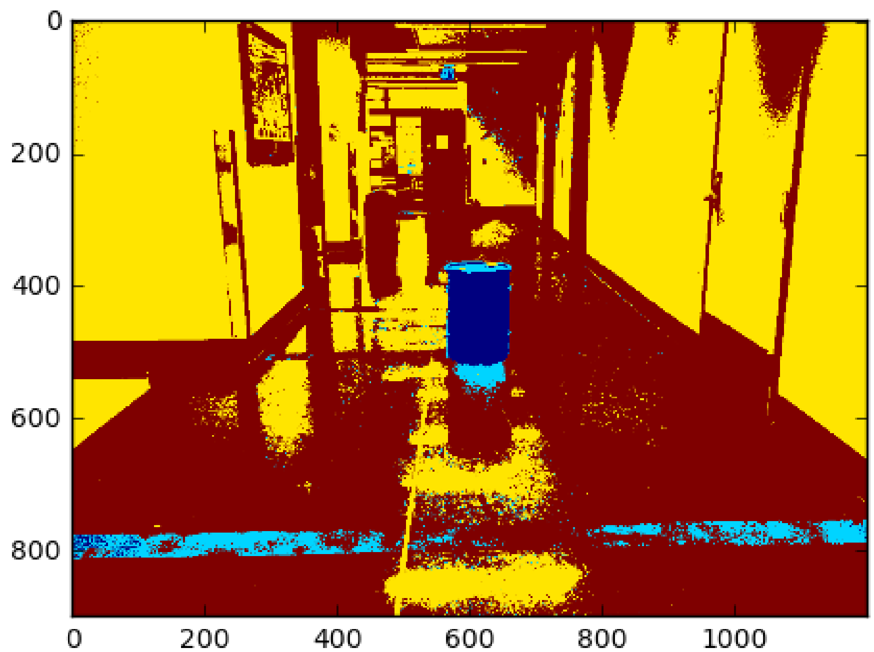
The output format:

barrel area.

Bound area

Barrelness

Image No, Coordinates, distance

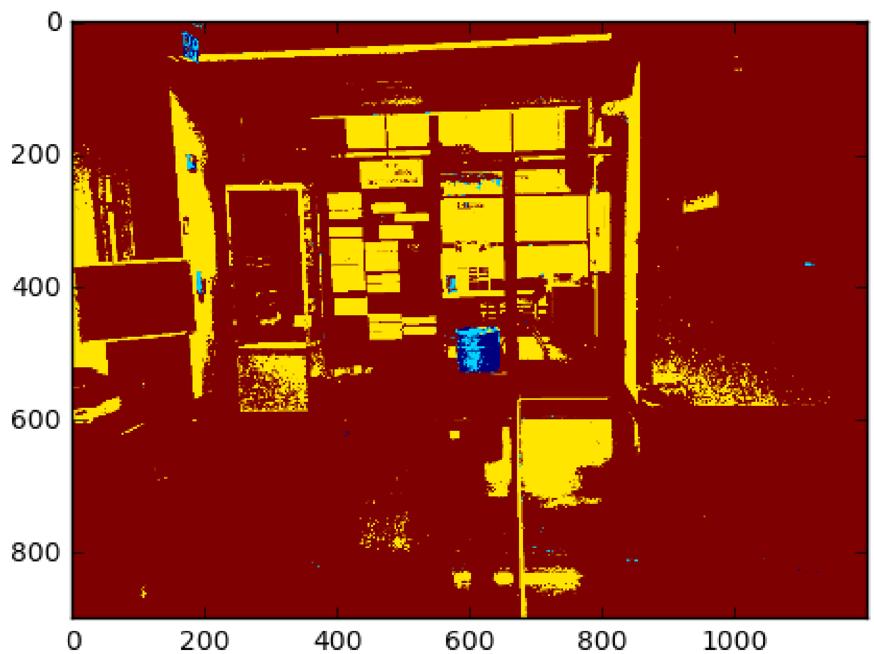



12589.5

14310.0873512

0.8797640217697729

ImageNo = 1 BottomLeftX = 563 BottomLeftY = 524 TopRightX = 659 TopRightY = 377 Distance = [[ 4.850 09113]]

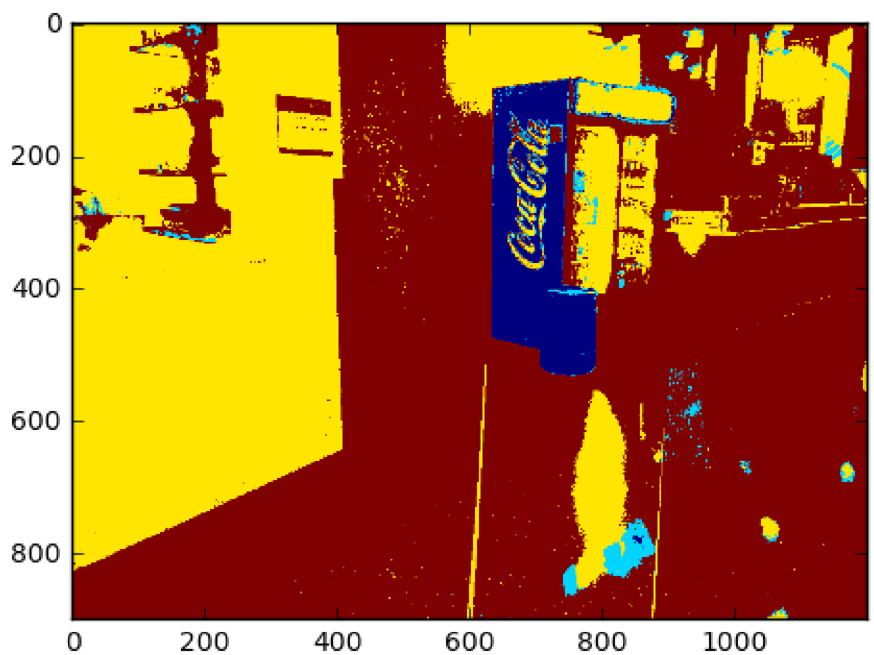



2792.0

3916.05107219

0.7129631224238994

ImageNo = 2 no barrels detected

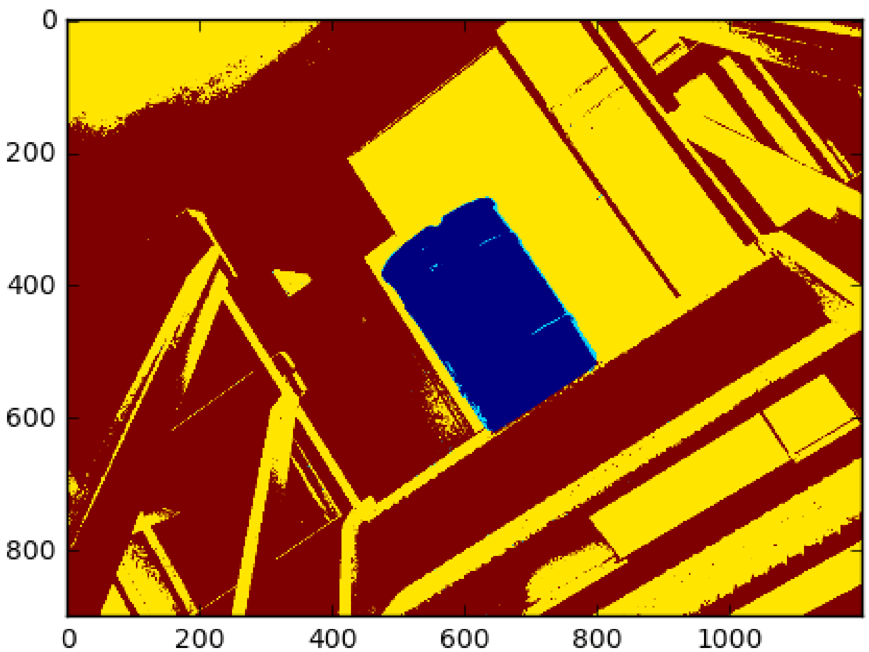



48983.5

69865.0

0.7011164388463466

ImageNo = 3 no barrels detected

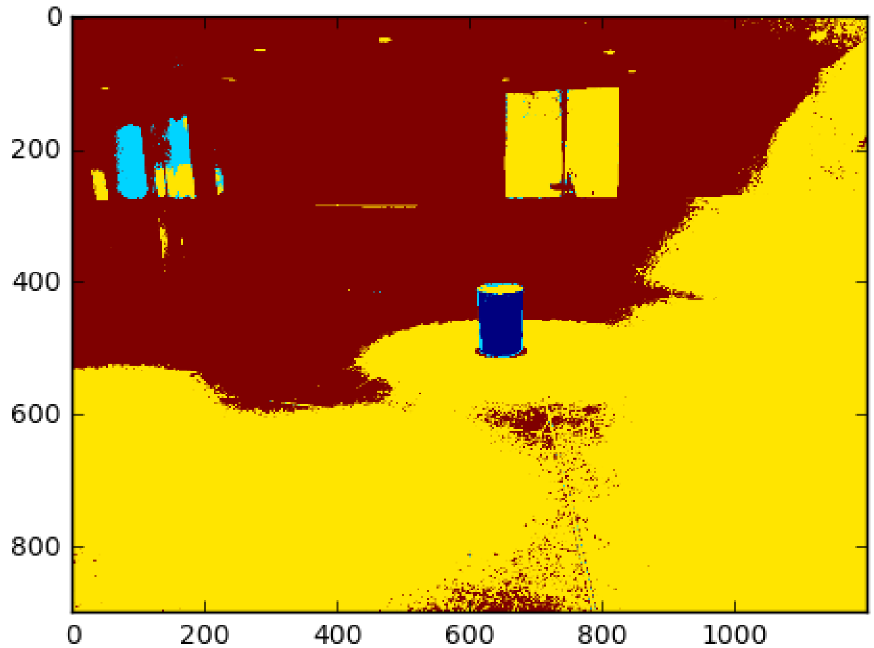



59862.0

64228.668311

0.9320137187054237

ImageNo = 4 BottomLeftX = 629 BottomLeftY = 631 TopRightX = 636 TopRightY = 256 Distance = [[ 3.35984151]]

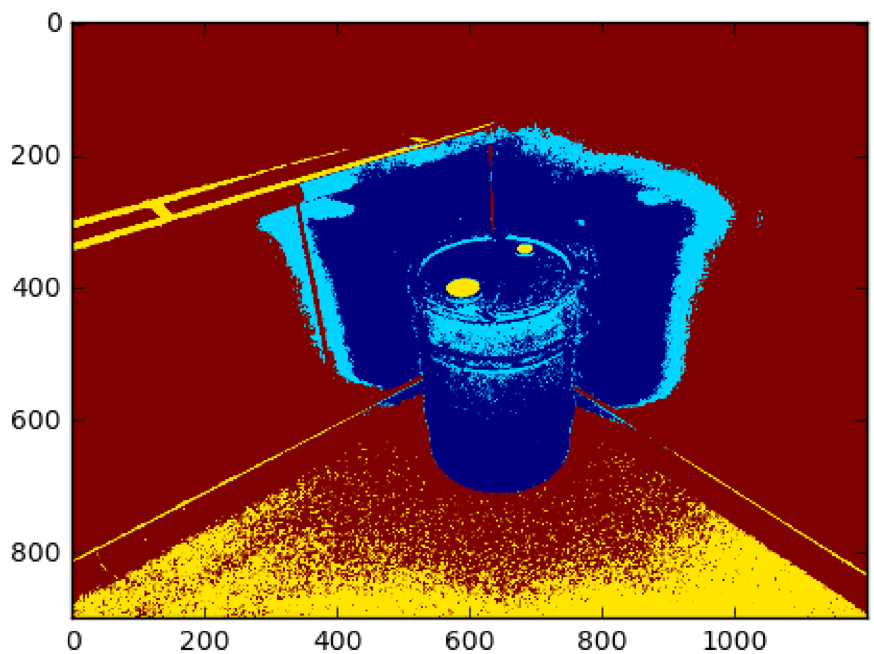



5767.0

7461.00060314

0.7729526248229763

ImageNo = 5 BottomLeftX = 609 BottomLeftY = 515 TopRightX = 677 TopRightY = 408 Distance = [[ 6.02149793]]

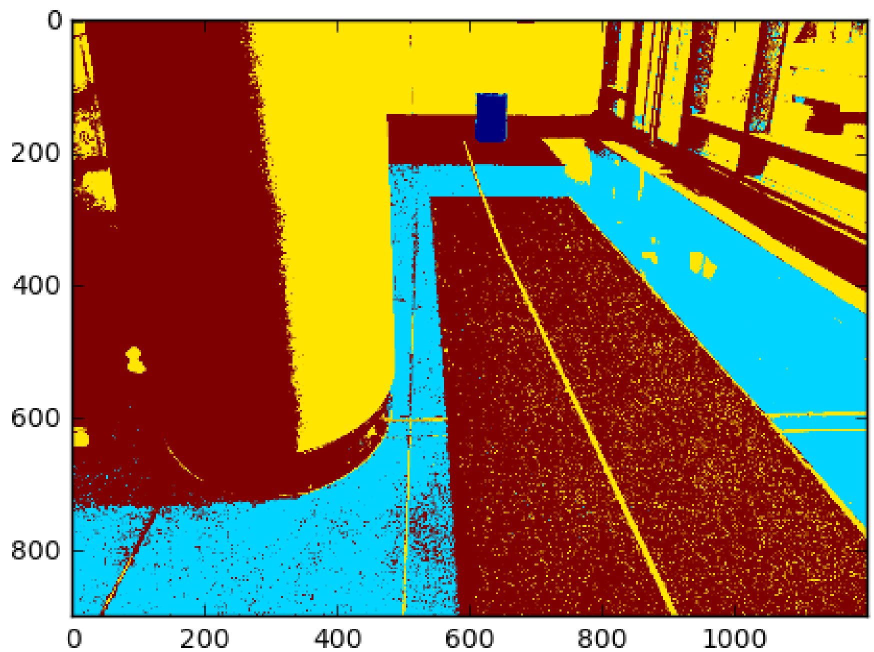


203537.0

304713.016409

0.6679629324625947

ImageNo = 6 no barrels detected

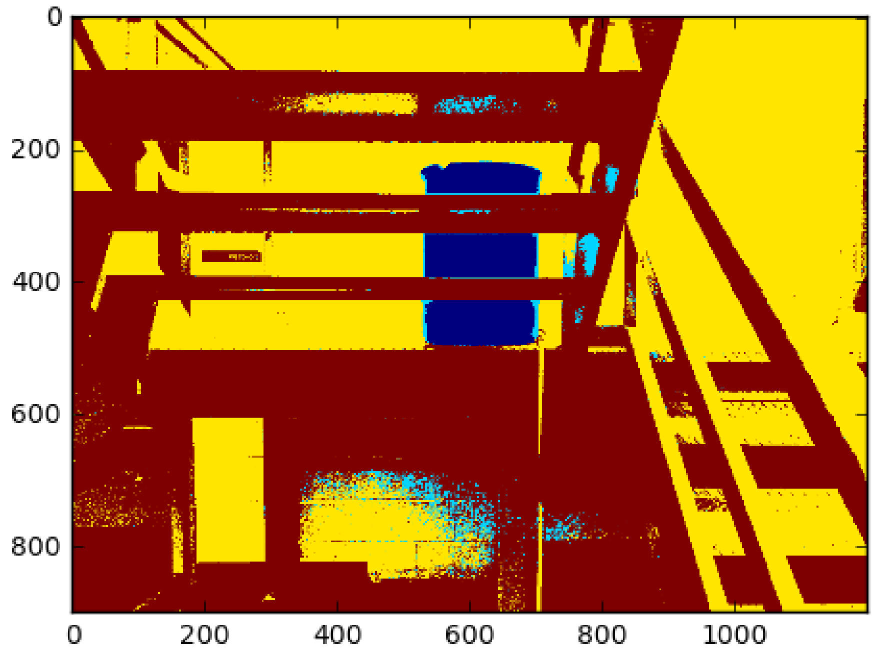



3243.5

3498.03659215

0.9272344398230402

ImageNo = 7 BottomLeftX = 609 BottomLeftY = 188 TopRightX = 653 TopRightY = 111 Distance = [[ 6.84571553]]




10504.5

10710.0

7330.5

11055.0

0.9502035278154681

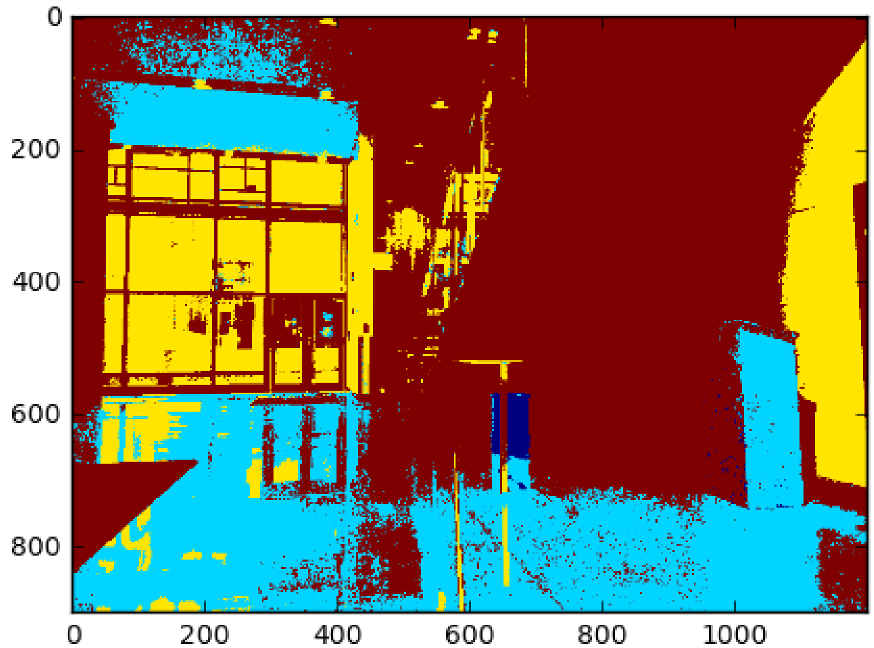
11088.0

0.9659090909090909

8460.0

0.8664893617021276

ImageNo = 8 BottomLeftX = 525 BottomLeftY = 499 TopRightX = 705 TopRightY = 222 Distance = [[ 4.0214648]]

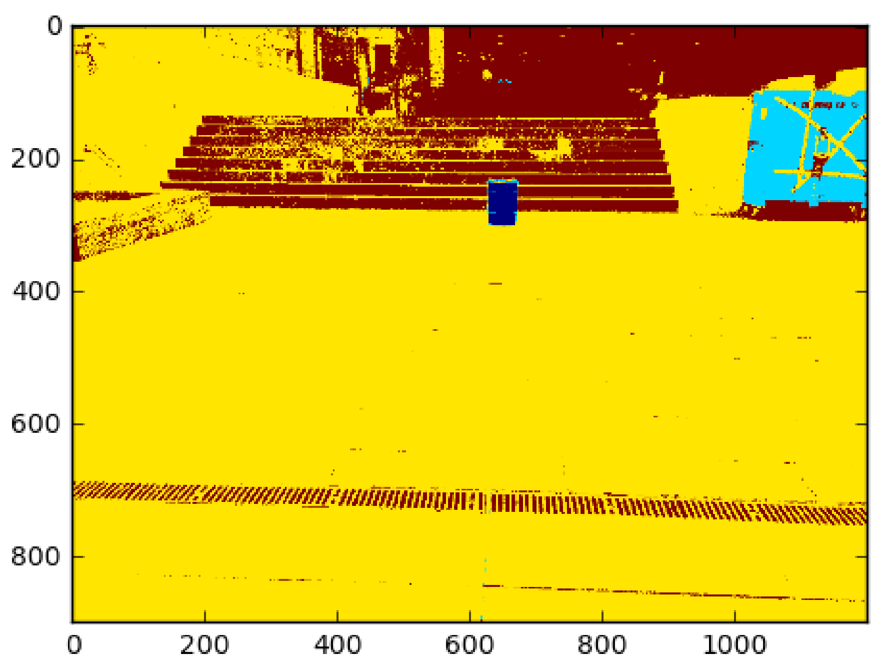



3058.0

3468.0

0.8817762399077278

ImageNo = 9 BottomLeftX = 656 BottomLeftY = 672 TopRightX = 690 TopRightY = 570 Distance = [[ 8.12373121]]



2612.5

2862.0924164

0.9127937256786253

ImageNo = 10 BottomLeftX = 627 BottomLeftY = 304 TopRightX = 669 TopRightY = 237 Distance = [[ 7.16487937]]