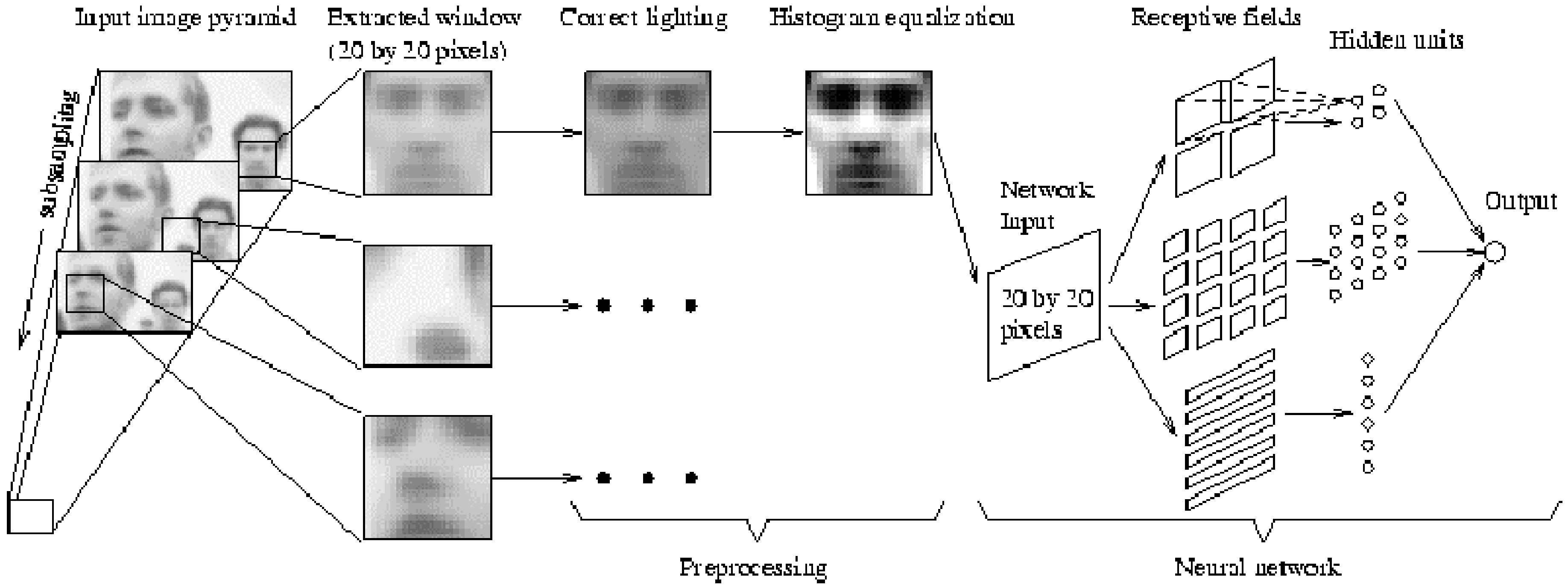
Example application: CMU face detector



From: http://www.ius.cs.cmu.edu/IUS/har2/har/www/CMU-CS-95-158R/

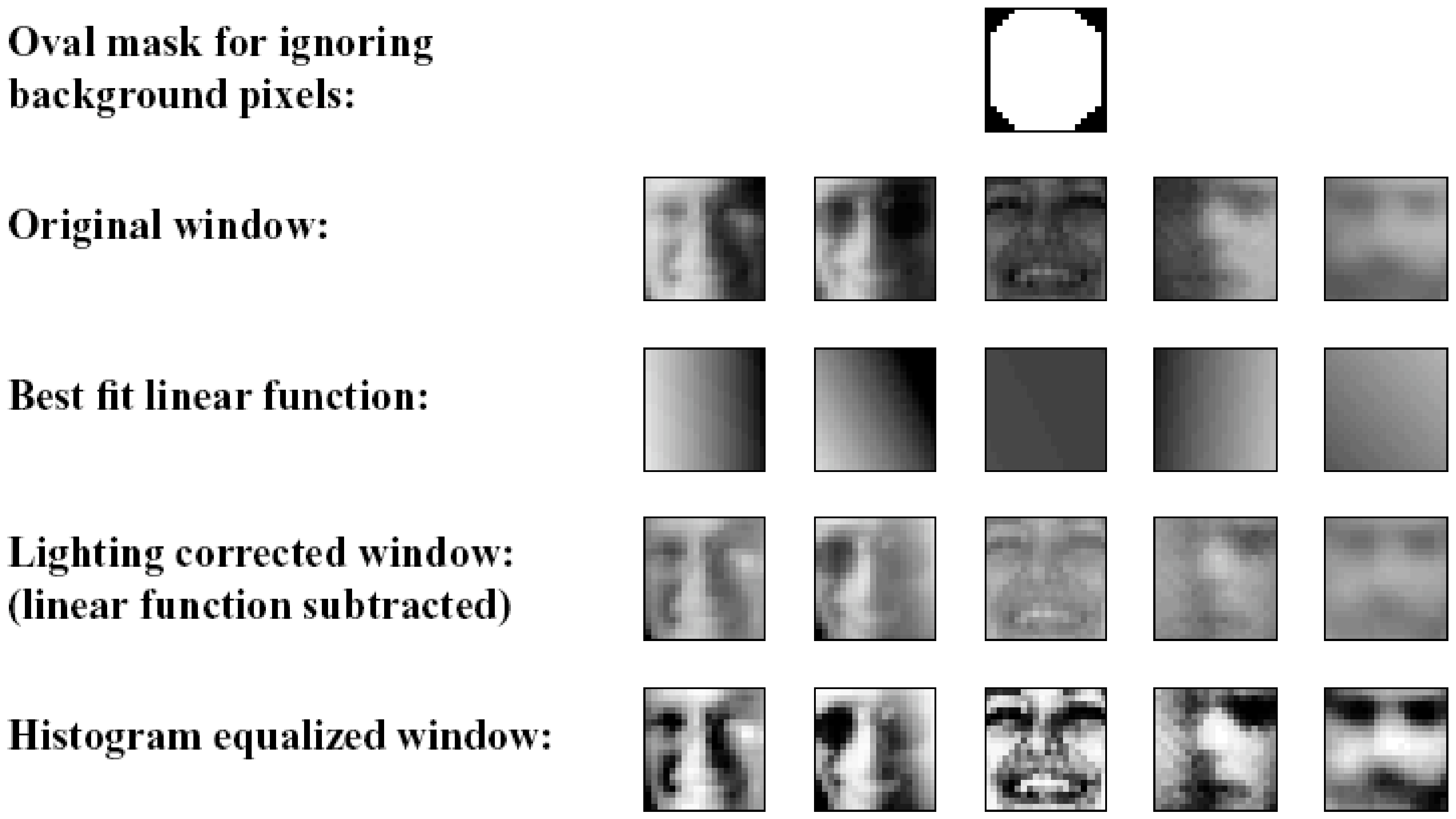

Oval mask for ignoring background pixels:
Original window:
Best fit linear function:
Lighting corrected window: (linear function subtracted)
Histogram equalized window:

Figure 2: The steps in preprocessing a window. First, a linear function is fit to the intensity values in the window, and then subtracted out, correcting for some extreme lighting conditions. Then, histogram equalization is applied, to correct for different camera gains and to improve contrast. For each of these steps, the mapping is computed based on pixels inside the oval mask, while the mapping is applied to the entire window.

Mr. Dupont is a professional wine taster. When given a French wine, he will identify it with probability 0.9 correctly as French, and will mistake it for a Californian wine with probability 0.1.

When given a Californian wine, he will identify it with probability 0.8 correctly as Californian, and will mistake it for a French wine with probability 0.2.

Suppose that Mr. Dupont is given ten unlabelled glasses of wine, three with French and seven with Californian wines. He randomly picks a glass, tries the wine, and solemnly says: "French". What is the probability that the wine he tasted was Californian?

Mr. Dupont is a professional wine taster. When given a French wine, he will identify it with probability 0.9 correctly as French, and will mistake it for a Californian wine with probability 0.1.

When given a Californian wine, he will identify it with probability 0.8 correctly as Californian, and will mistake it for a French wine with probability 0.2.

Suppose that Mr. Dupont is given ten unlabelled glasses of wine, three with French and seven with Californian wines. He randomly picks a glass, tries the wine, and solemnly says: "French". What is the probability that the wine he tasted was Californian?

	Rf	Rc
F	0.9	0.1
C	0.2	0.8

P(F) = 0.3; P(C) = 0.7;

$$
\begin{aligned}
P(C|Rf) &= P(Rf|C)\ p(\ C\ )/P(Rf) \\
&= 0.2*0.7/\square_{w}\ P(Rf\ |w)p(w) \\
&= 0.2*0.7/(0.9*0.3+0.2*0.7) = 0.34 \\
&= 0.2*0.7/0.41 = 0.34
\end{aligned}
$$

Bayes theorem

$$P(x, y) = P(x|y)\,P(y)$$

so

$$P(x|y)\,P(y) = P(y|x)\,P(x)$$

and

$$P(\underset{\underset{\text{The parameters you want to estimate}}{\uparrow}}{x}|\underset{\underset{\text{What you observe}}{\uparrow}}{y}) = \underset{\underset{\text{Likelihood function}}{\uparrow}}{P(y|x)}\;\underset{\underset{\text{Prior probability}}{\uparrow}}{P(x)}\;/\;\underset{\underset{\text{Constant w.r.t. parameters x.}}{\uparrow}}{P(y)}$$

"You must choose, but Choose Wisely"

- Given only probabilities, can we minimize the number of errors we make?
- *Given:*

  responses R_i, categories C_i, current category c, data x

- *To Minimize error:*
  - Decide R_i **if** $P(C_i \mid x) > P(C_k \mid x)$ **for all** $i \neq k$

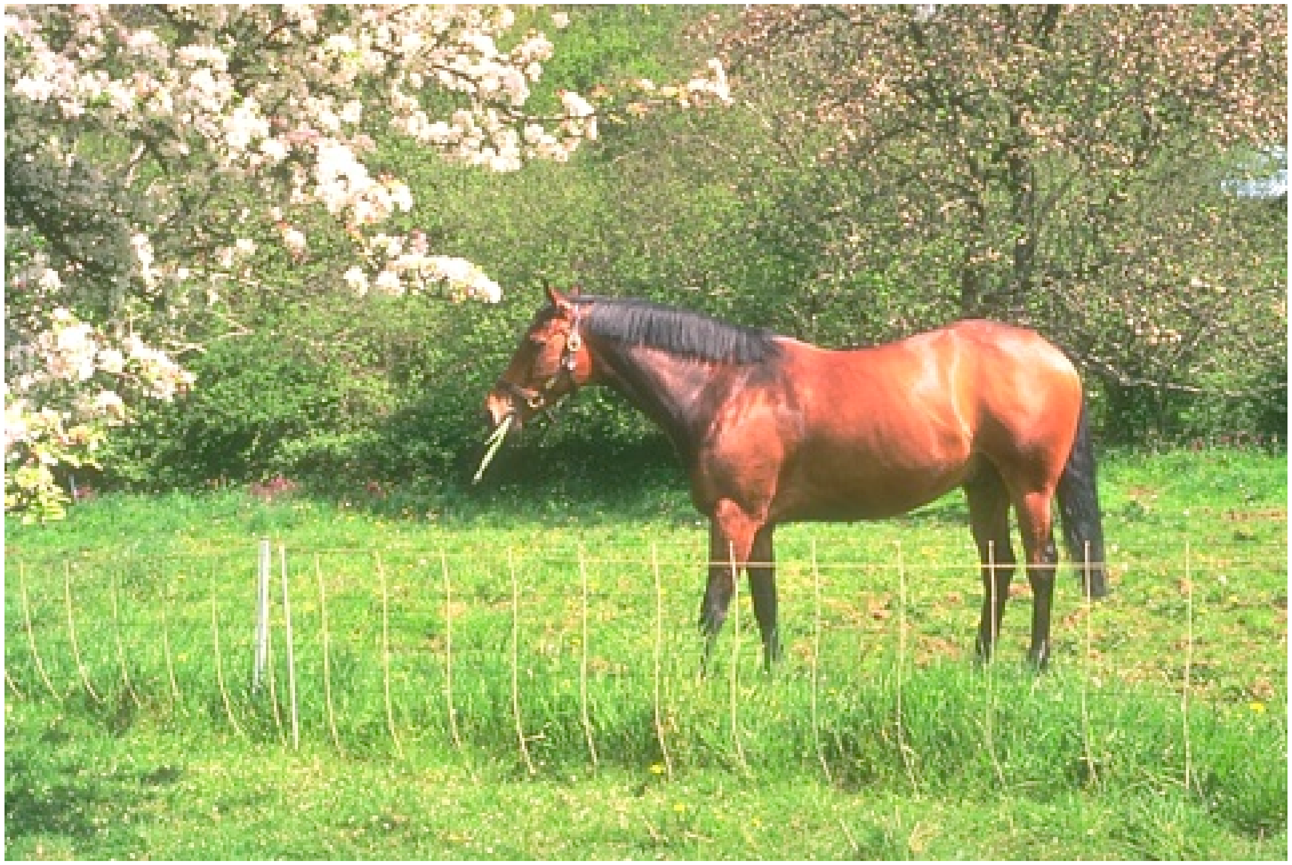
    $P(x \mid C_i)\, P(C_i) > P(x \mid C_k)\, P(C_k)$

    $P(x \mid C_i) / P(x \mid C_k) > P(C_k) / P(C_i)$

    $P(x \mid C_i) / P(x \mid C_k) > T$

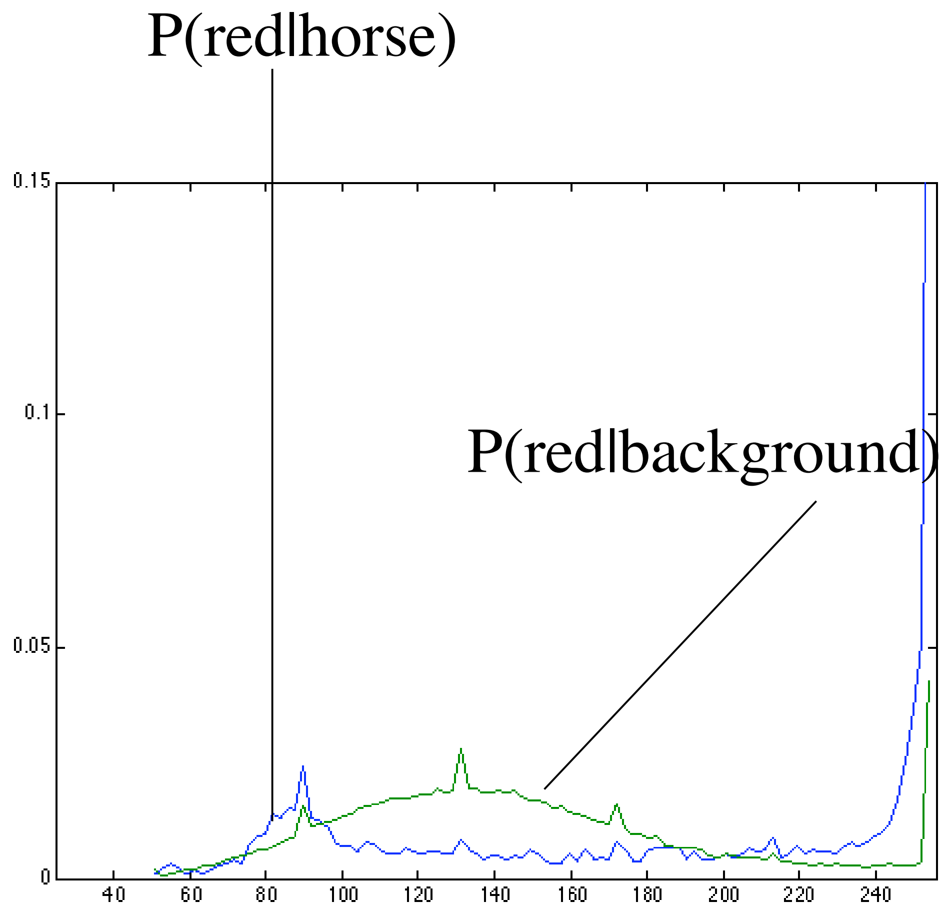
Optimal classifications always involve hard boundaries

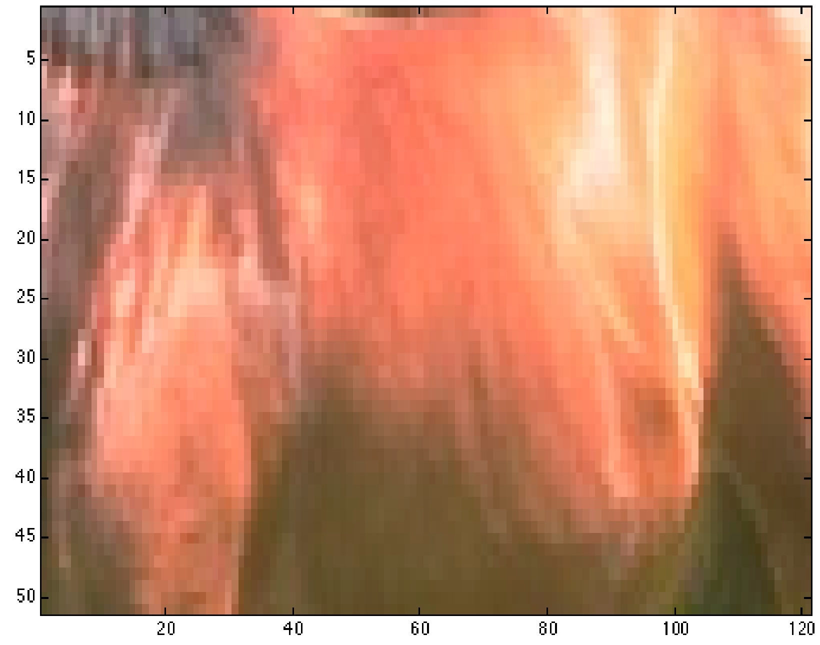
Horse Segmentation

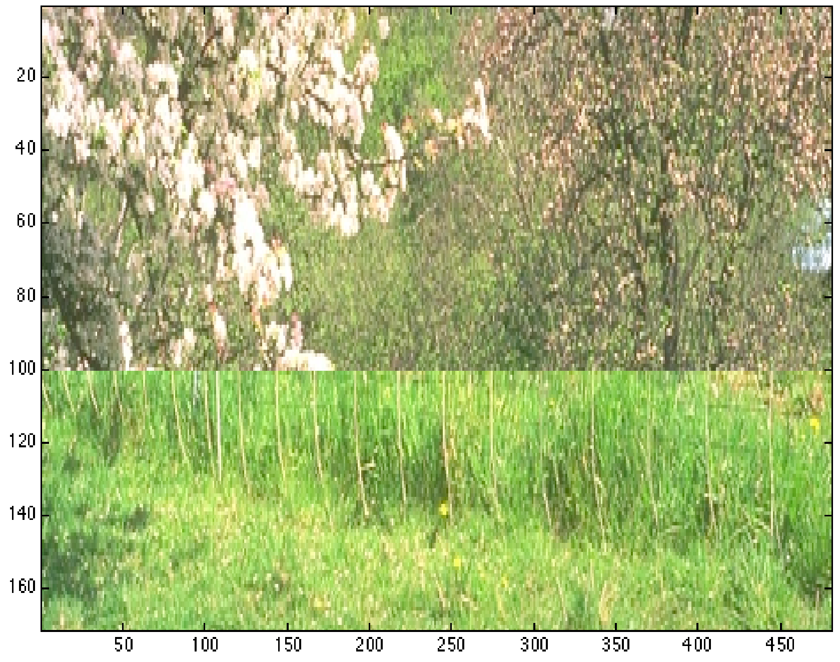


$P(\text{horse}) = 0.04$

$P(\text{background}) = 0.96$

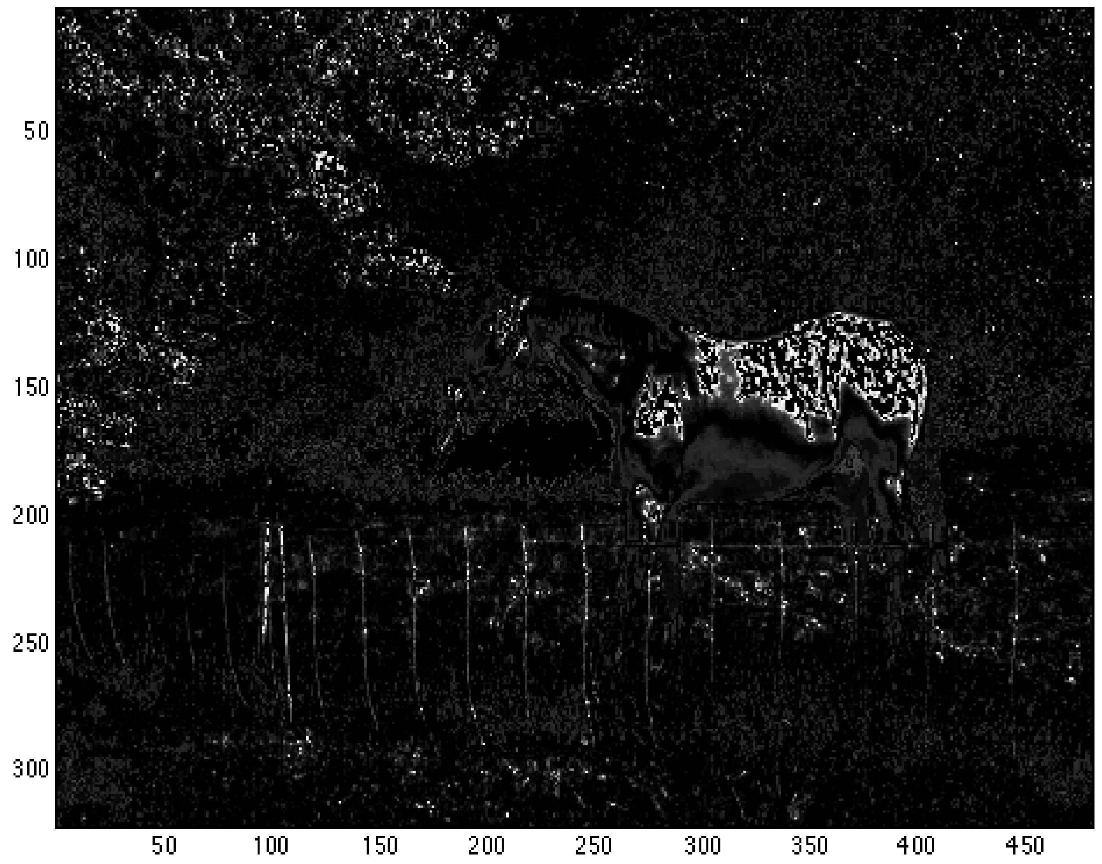




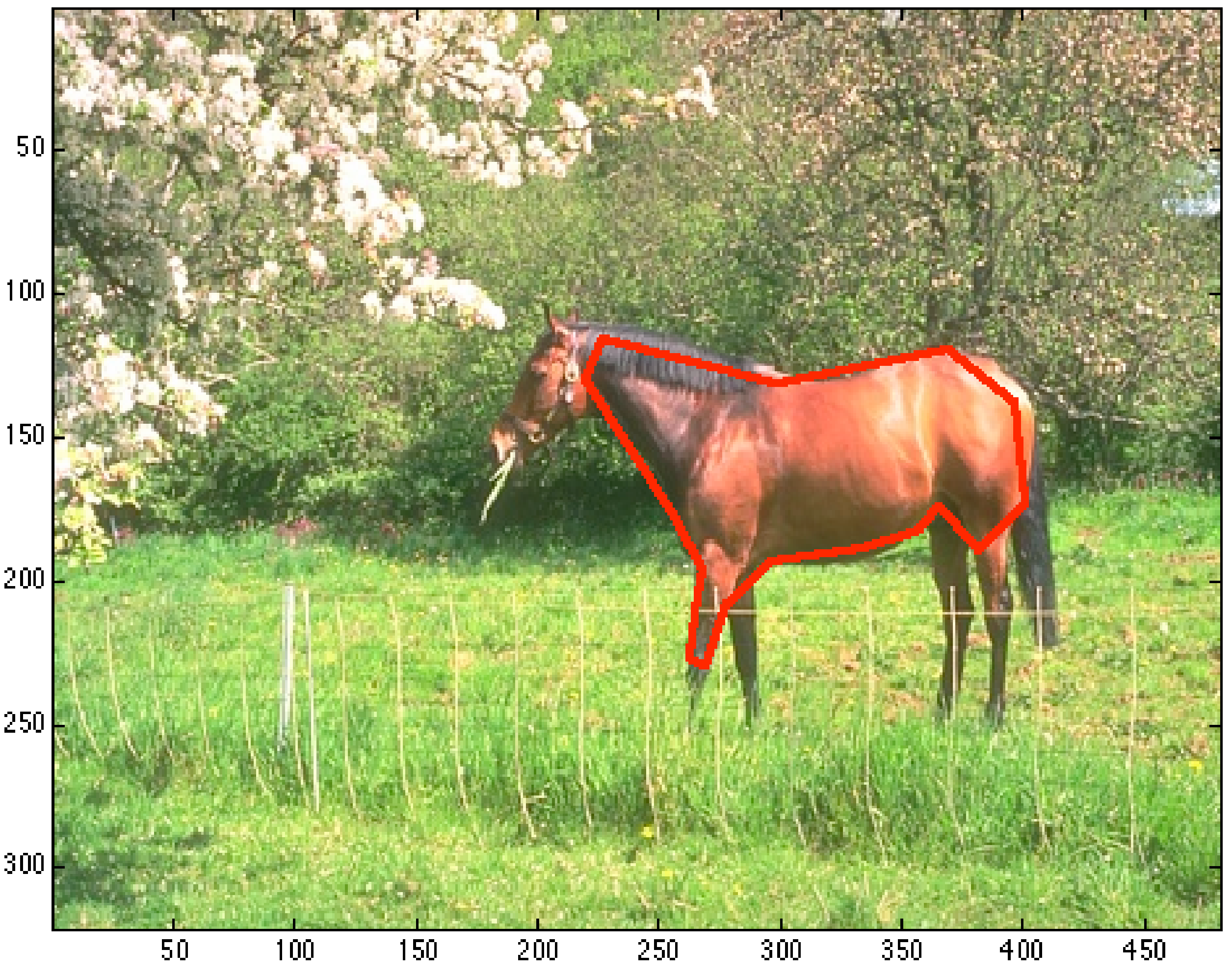


Now evaluate

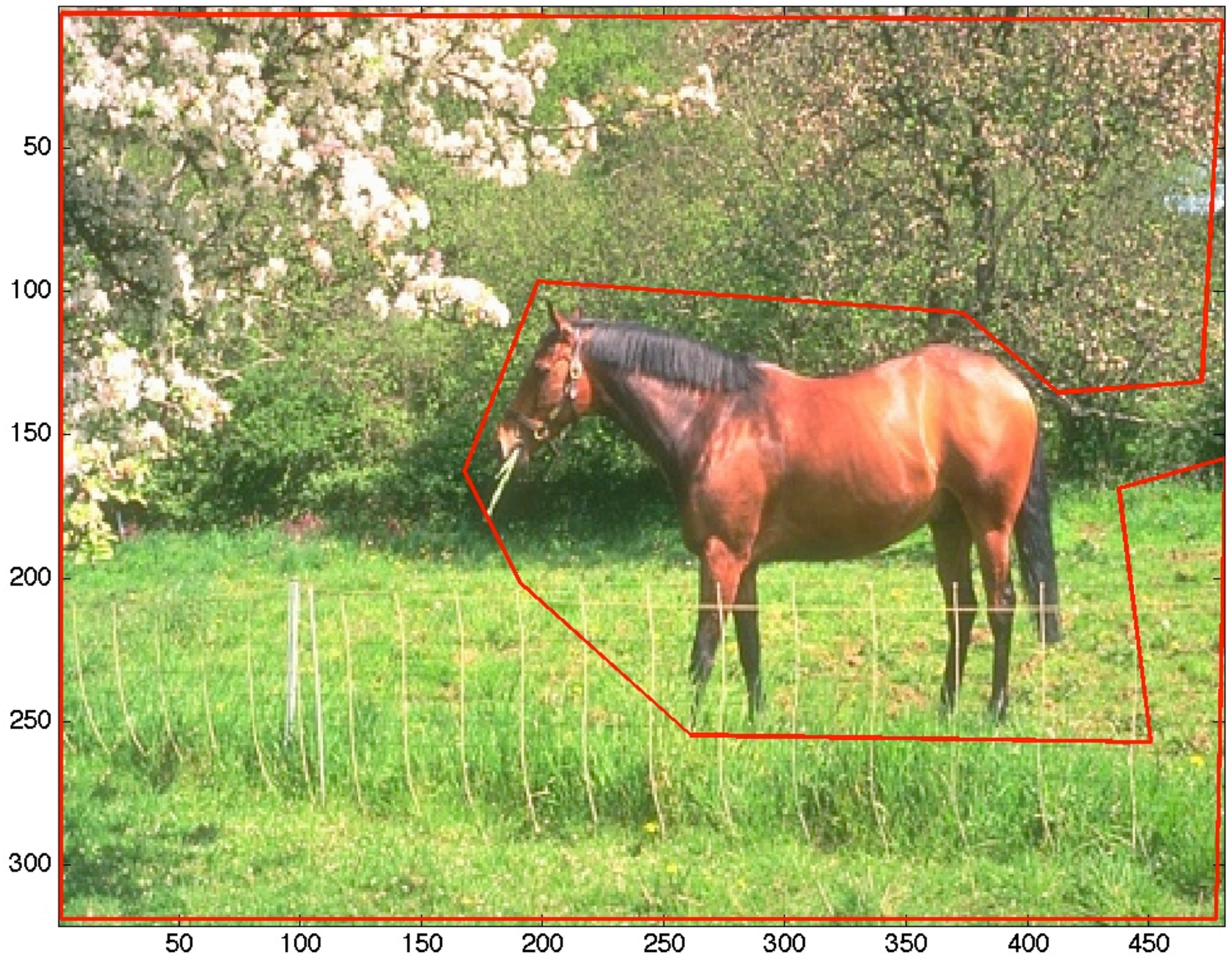
$$\prod_{j=1:Nmeasurements} p(r_j \mid horse) / p(r_j \mid background)$$



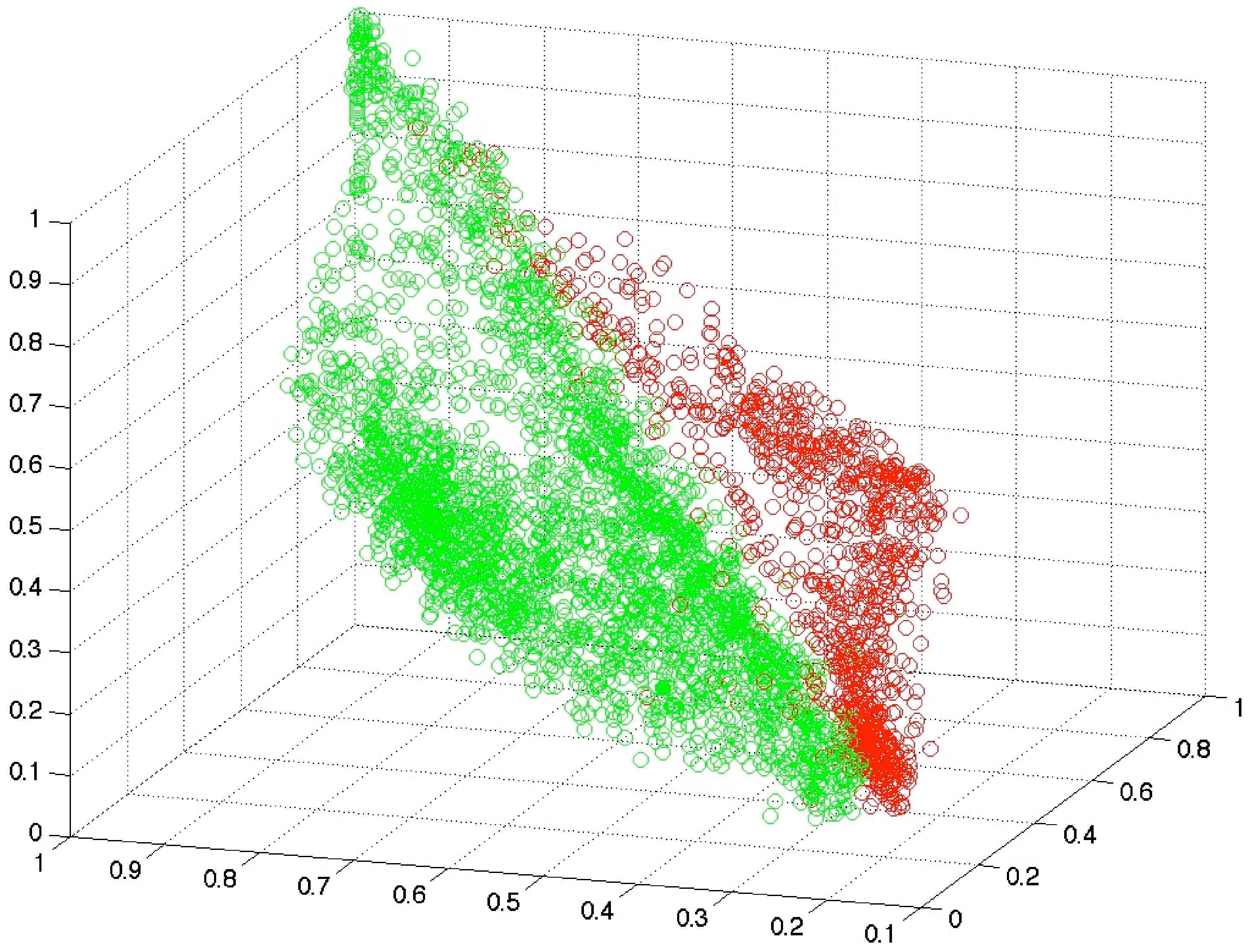
Foreground



background

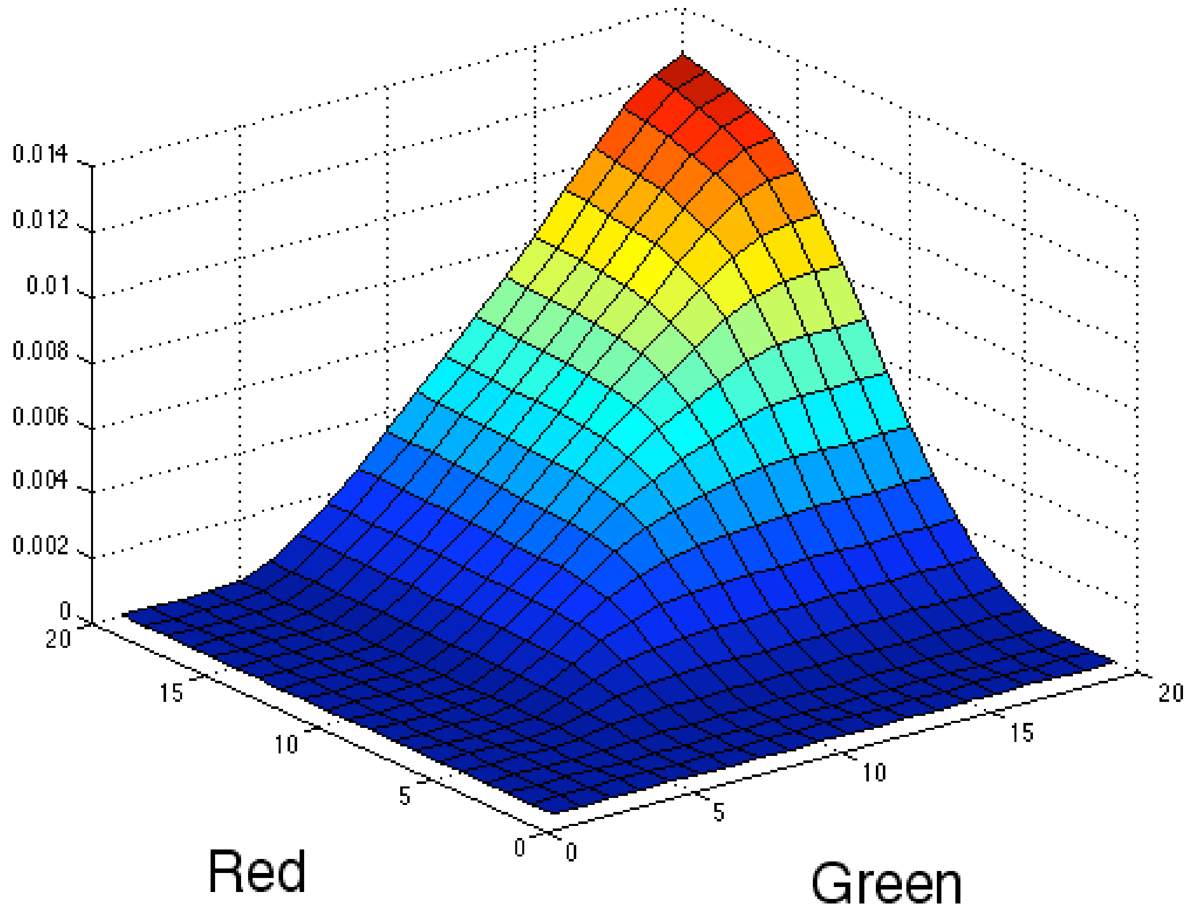


Pixels in color space

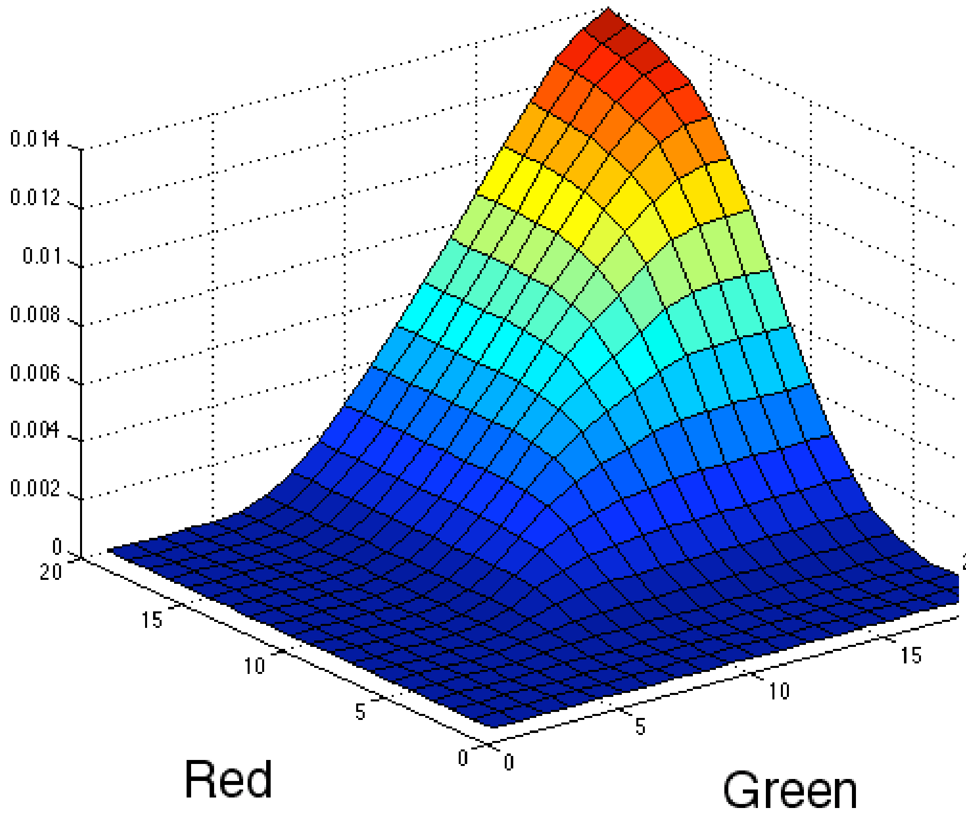


Histograms

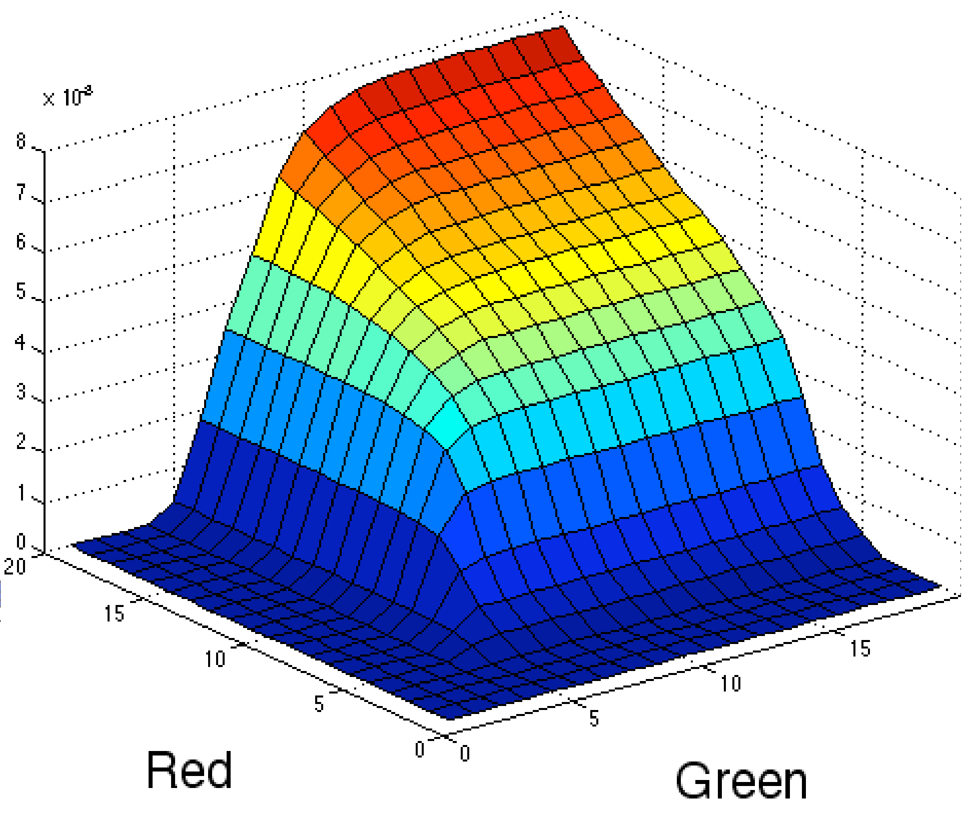
Entire Image



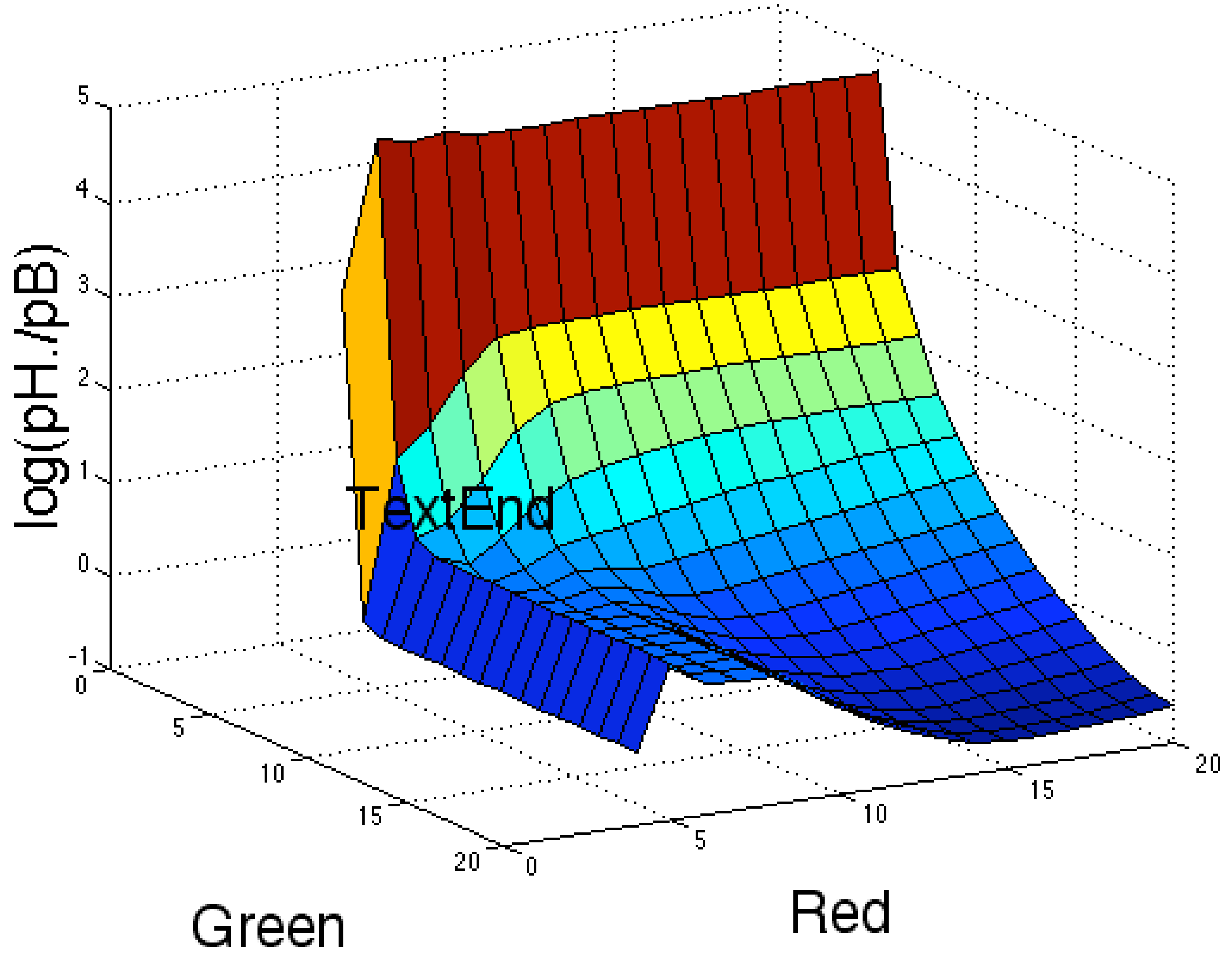
Background

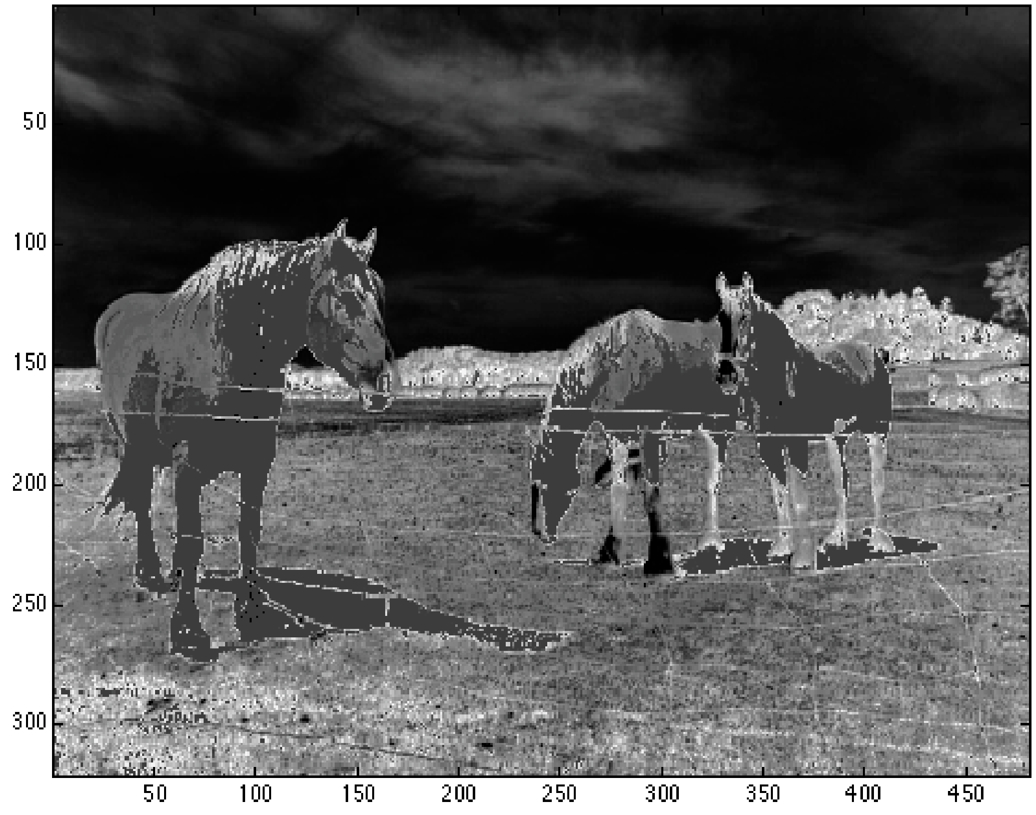


Horse



Log(p(r,g|horse)/p(r,g|backgrnd))

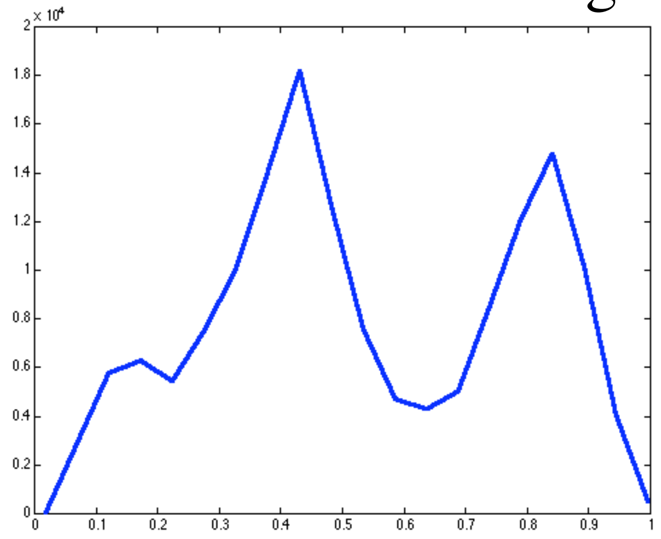


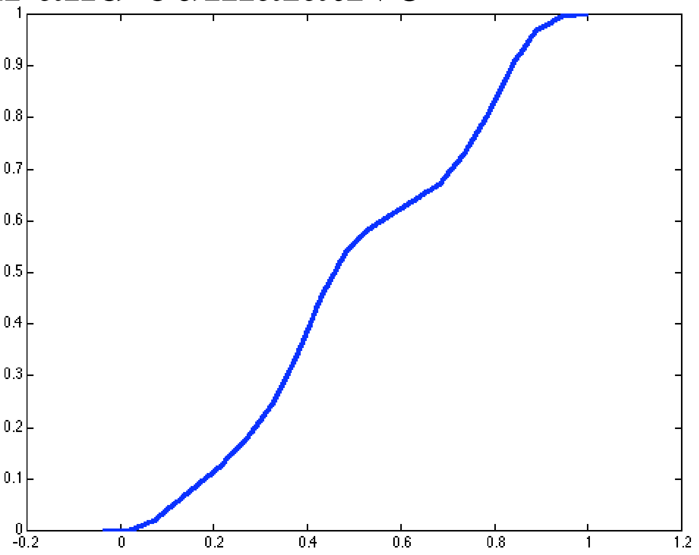


50
100
150
200
250
300
50
100
150
200
250
300
350
400
450

Histogram Matching

Find current histogram and cumalative





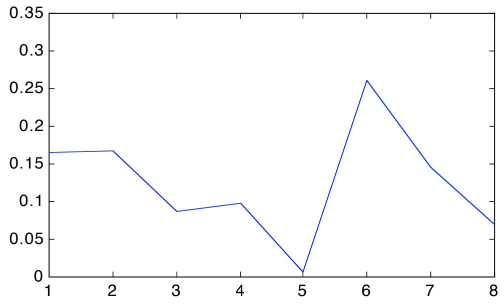
Use current cumulative as an inverse transform,
Use desired cumulative as forward transform.

Histogram Matching Code

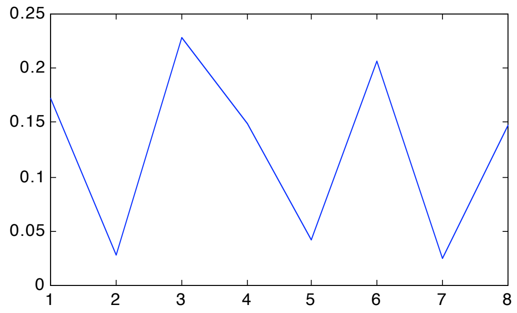
```
function matchedvalues = histogrammatch(oldvalues, desiredCount, desiredbinvals)
% matchedvalues = histogrammatch(oldvalues, desiredCount, desiredbinvals)
% nonlinearly transform your numbers to enforce a desired histogram ( a table of values and counts)
% writen by: P. Schrater 2003
[oldcount,oldbinvals]=hist(oldvalues(:),sqrt(length(oldvalues(:))));
%eliminate zero counts and find cumulative table
zind = find(oldcount==0);
oldcount(zind)=[]; oldbinvals(zind)=[];
cumprobold = cumsum(oldcount)/sum(oldcount);

% assign each oldvalue its cumulative prob
pvaluesold = interp1(oldbinvals,cumprobold,oldvalues(:));

% now do same for desired:
%eliminate zero counts and find cumulative table
zind = find(desiredCount==0);
desiredCount(zind)=[]; desiredbinvals(zind)=[];
cumprobnew = cumsum(desiredCount)/sum(desiredCount);

% translate our pvalues back to values by running through the new probability table
matchedvalues = interp1(cumprobnew,desiredbinvals,pvaluesold);

matchedvalues = reshape(matchedvalues,size(oldvalues));
```

Mutual Info

Py =
0.1647
0.1680
0.0872
0.0981
0.0064
0.2609
0.1457
0.0690



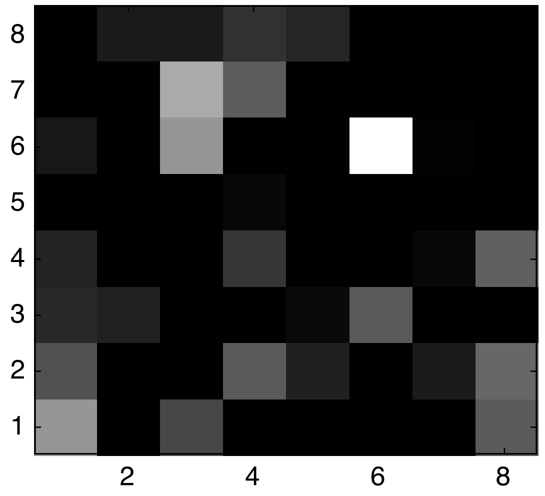
Px =
0.1728
0.0286
0.2276
0.1495
0.0425
0.2064
0.0244
0.1481



Dependent distribution

p(x,y)

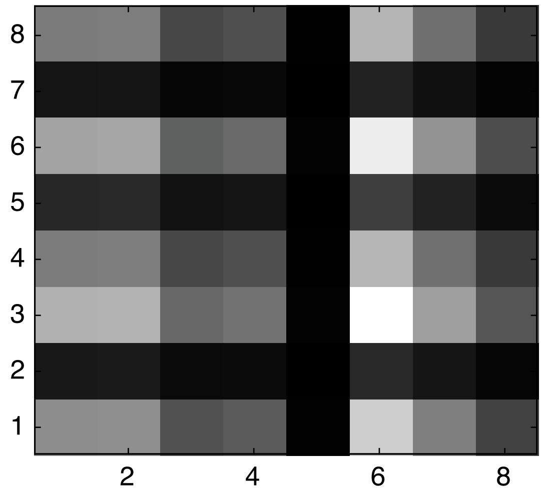
Pxy = P(y|x).*repmat(px,8,1);

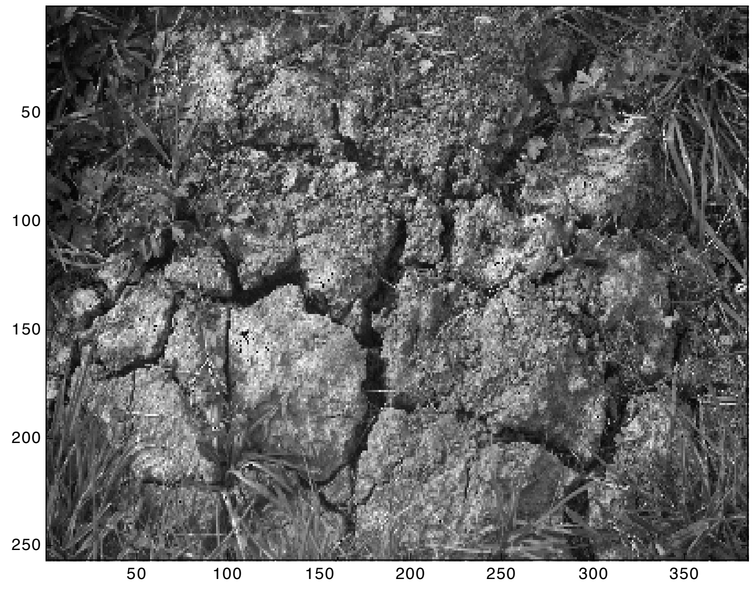


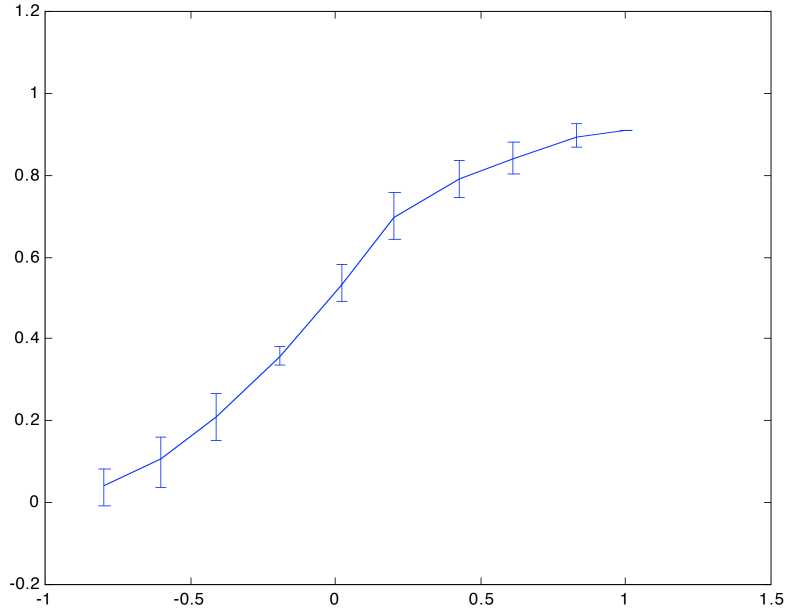
Independent distribution

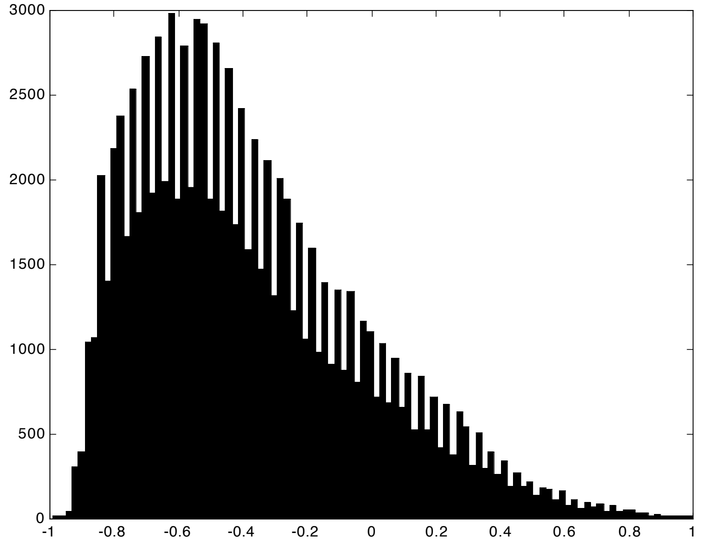
p(x)p(y)

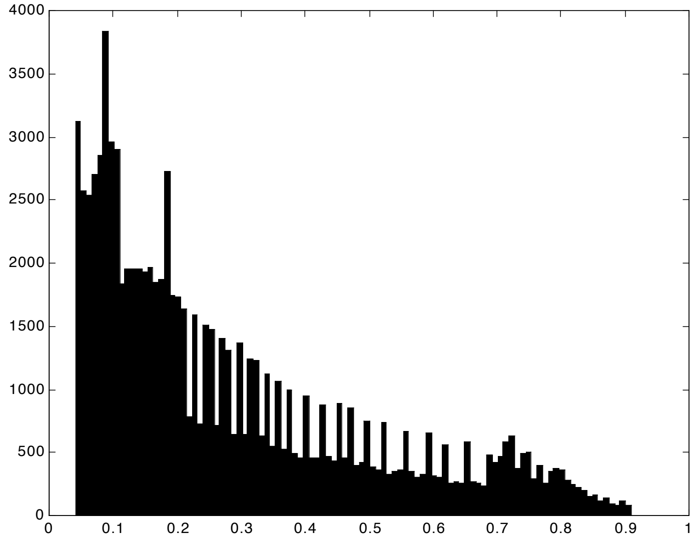
Pxyind = Px*Py' = [8x8]



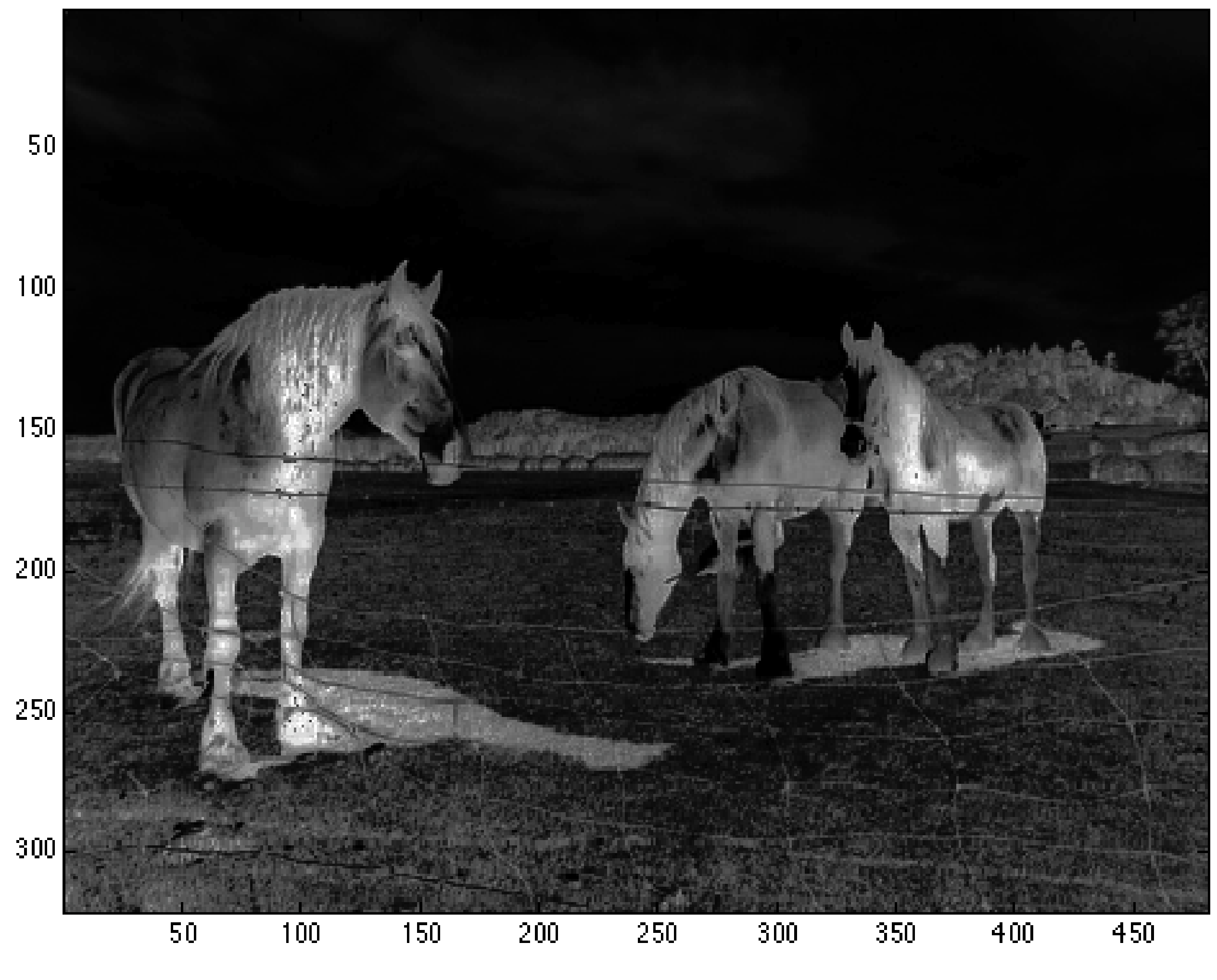

50
100
150
200
250
50
100
150
200
250
300
350


1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-1
-0.5
0
0.5
1
1.5


3000
2500
2000
1500
1000
500
0
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1


4000
3500
3000
2500
2000
1500
1000
500
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

After histogram Equalization



Moral of the story

- You can’t learn much from one picture:
  - One image does not capture variation due to:
    - camera-based color correction
    - Changes in lighting between images
    - Changes in viewpoint and distance between images
  - These sources are extremely important to model
    - Preprocess your images
    - Use large training set.

Intrinsic difficulty segmentation Problem

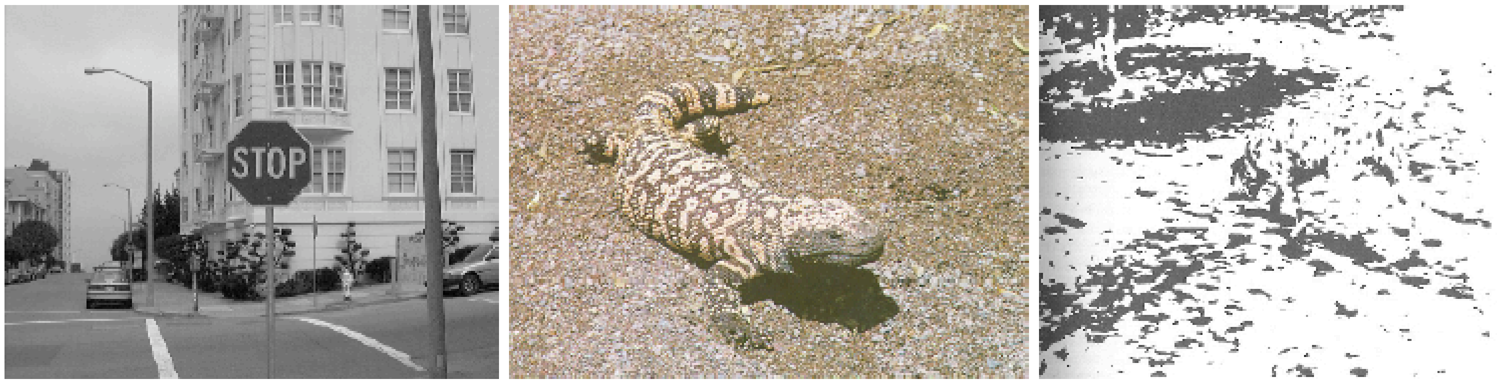


Figure 1: Left to right, three detection tasks of increasing degrees of difficulty. The stop sign (left) is easy to find. The gila monster (centre) is harder. The dalmation dog (right) is almost impossible.