COMP2005

Histogram Processing

Histograms

- The histogram of a digital image with grey levels in the range [0, L-1] is a discrete function

$$p(r_k) = n_k$$

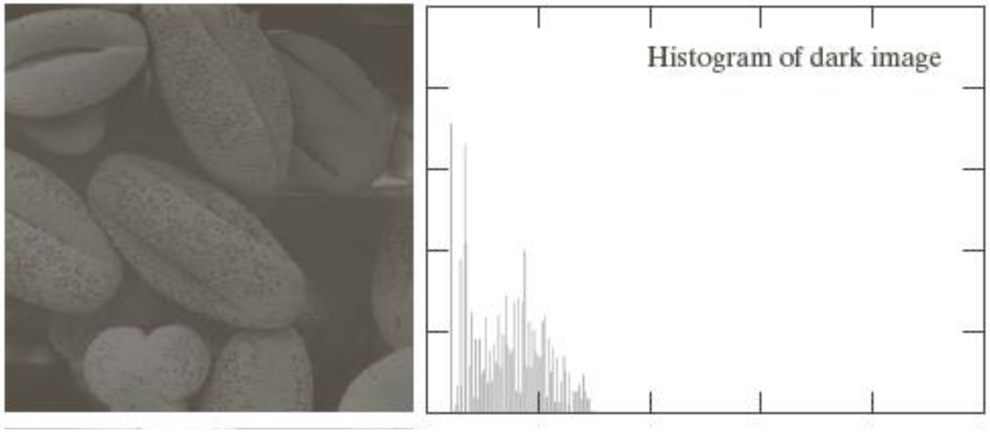
where:

  - r_k is the kth grey level,
  - n_k is the number of pixels in the image with that grey level,
  - $k = 0, 1, \ldots, L-1$

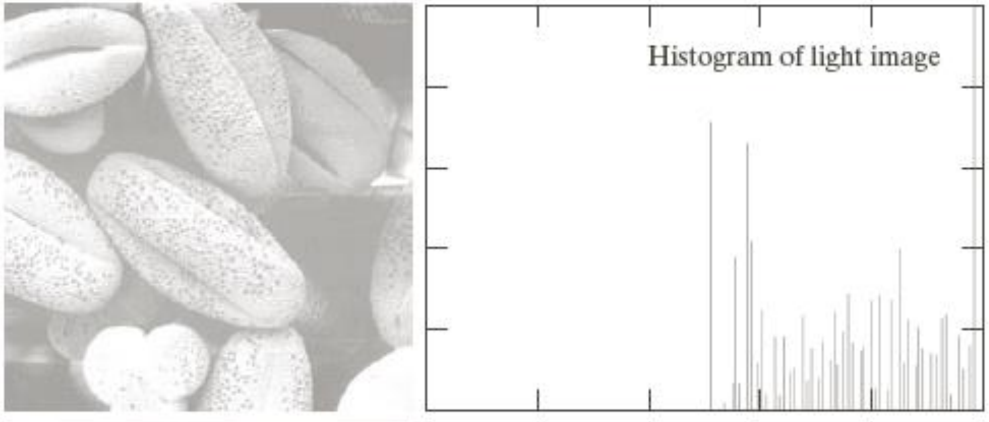
- Histograms provide useful global information about the image, ease computation of some image properties, and can be manipulated to improve the image

Histograms

- Dark

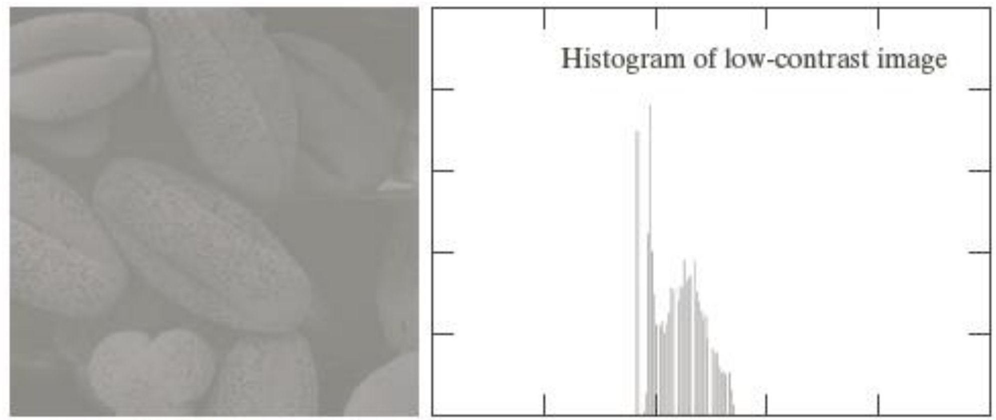


- Light

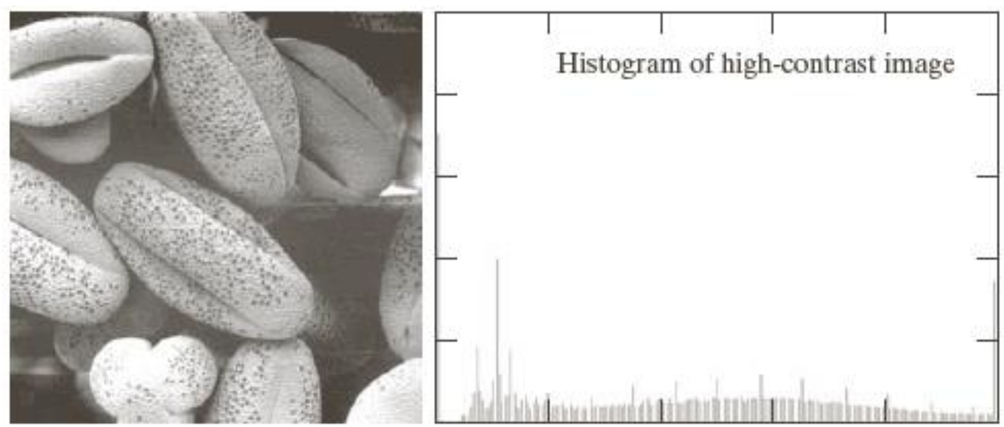


Histograms

- Low Contrast



- High Contrast



Normalised Histograms

- A normalised histogram is a discrete function

$$p(r_k) = n_k / n$$

where:

  - n = width x height is the total number of pixels in the image
- The bins in a normalised histogram sum to one
- Each bin gives the probability of the corresponding grey level appearing in the image
- The probabilistic interpretation is valuable in e.g. contrast enhancement and automatic thresholding

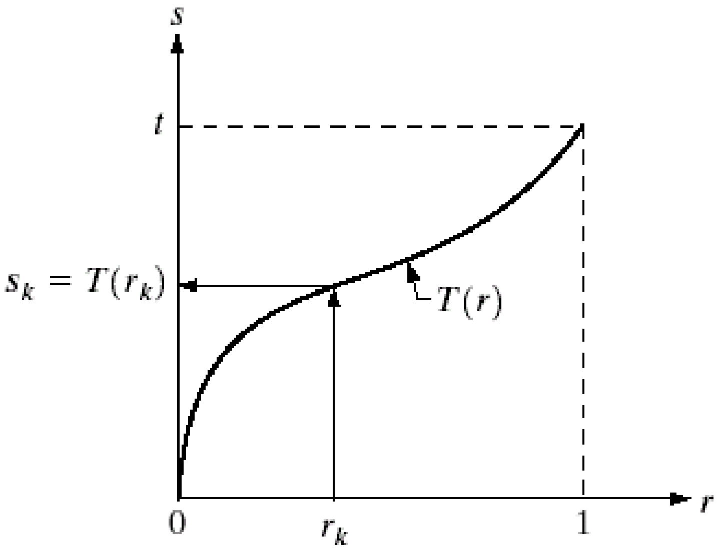
Histogram Equalisation

- Goal is to improve the contrast of an image
  - To transform an image in such a way that the transformed image has a nearly uniform distribution of pixel values
  - More general than linear or piecewise contrast stretching
- Histogram transforms
  - Map an input histogram r onto a new histogram s
  - Assume *r* has been normalized to the interval [0,1], with *r* = 0 representing black and *r* = 1 representing white

$$s = T(r) \qquad 0 \leq r \leq 1$$

Histogram Equalisation

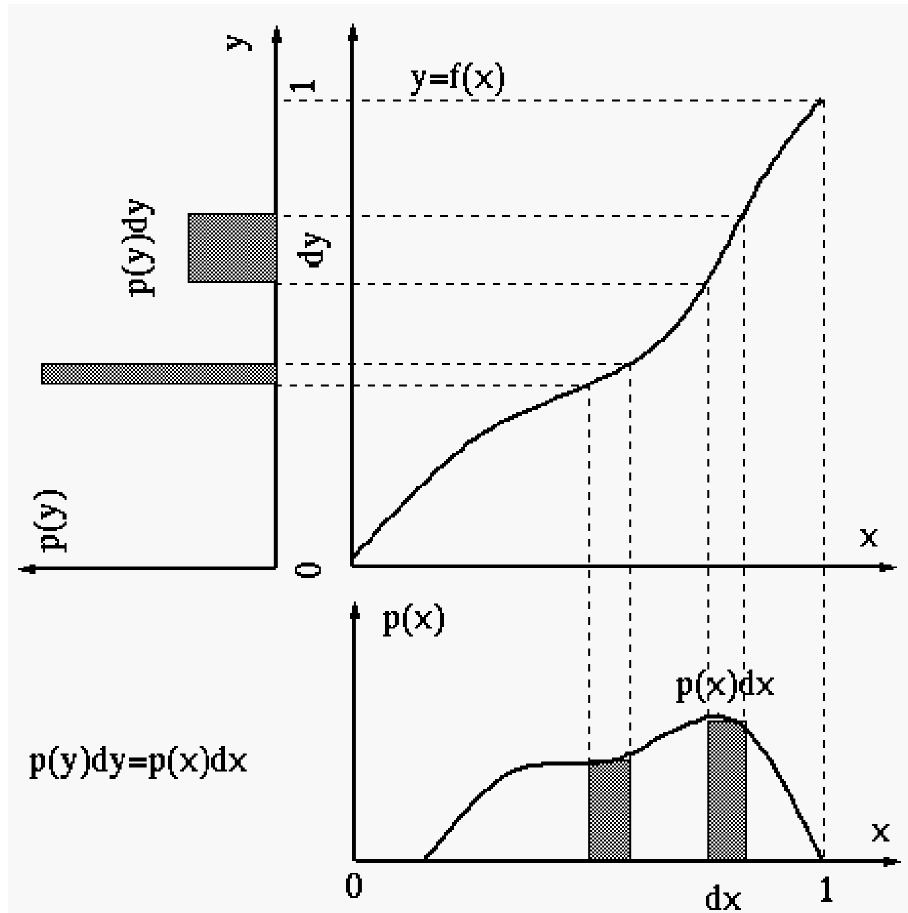
- The transformation function satisfies the following conditions
  - $T(r)$ is single-valued and strictly monotonically increasing in the interval 0 ≤ r ≤ 1
  - 0 ≤ T(r) ≤ 1 for 0 ≤ r ≤ 1
- This means it is possible to invert the process
- It also gives us the relationship tha allows the derivation of histogram equalisation



Histogram Equalisation

$P(y).dy = P(x).dx$

So $P(y) = P(x).\frac{dx}{dy}$



Histogram Equalisation

- In Gonzalez and Woods' notation….
- Let $p_r(r)$ and $p_s(s)$ denote the probability density function of random variables r and s, respectively
- If $p_r(r)$ and $T(r)$ are known, then the probability density function $p_s(s)$ of the transformed variable s can be obtained

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right|$$

- $r = T^{-1}(s)$

Histogram Equalisation

- If we choose as the transformation function the cumulative distribution function or CDF:

$$T(r) = \int_0^r p_r(w)\,dw$$

$$\frac{ds}{dr} = \frac{dT(r)}{dr} = \frac{d}{dr}\left[\int_0^r p_r(w)\,dw\right] = p_r(r)$$

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right| = p_r(r)\left|\frac{1}{p_r(r)}\right| = 1 \quad 0 \le s \le 1$$

$T(r)$ depends on $p_r(r)$, but the resulting $p_s(s)$ is always uniform.

Histogram Equalisation

- We have a discrete histogram, not a PDF of a continuous random variable
- The probability of occurrence of gray level r_k in an image is

$$p_r(r_k) = \frac{n_k}{n} \qquad k = 0, 1, 2, \ldots, L - 1$$

- The transformation function is

$$s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j) = \sum_{j=0}^{k} \frac{n_j}{n} \qquad k = 0,1,2,\ldots, L-1$$

- An output image is obtained by mapping each pixel with level r_k in the input image into a corresponding pixel with level s_k.

In Practice

- To perform histogram equalisation
  - Compute the CDF of the input image.
  - For each pixel in the input image, the corresponding output pixel intensity is calculated by using the CDF as a look-up function.
  - CDF values will be in the range 0 - 1, scale the equalised image to fit the range supported by the output image format.
- The histogram of the output image will be approximately uniform

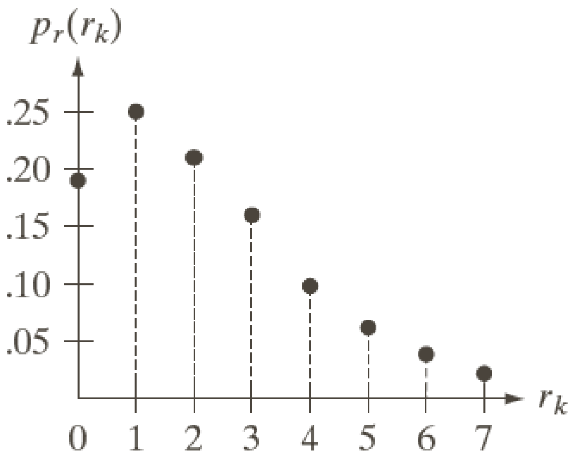
In Practice

- Consider a 64 x 64 pixel, 3 bit (8 grey level) image

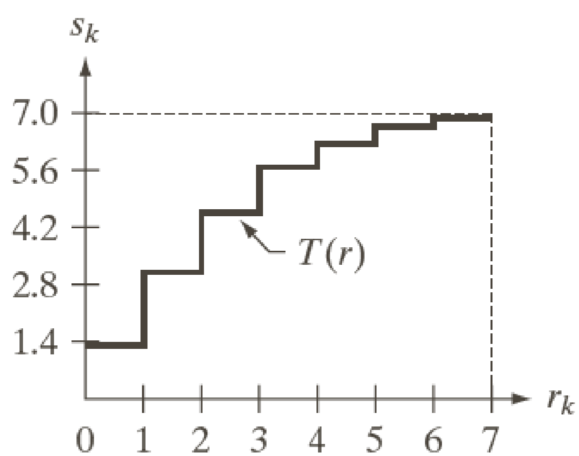
(Example from Gonzalez & Woods)

r_k	n_k	$P_r(r_k)$	$T(r_k)$	s_k	Round
0	790	0.19	0.19	1.33	1
1	1023	0.25	0.44	3.08	3
2	850	0.21	0.65	4.55	5
3	656	0.16	0.81	5.67	6
4	329	0.08	0.89	6.23	6
5	245	0.06	0.95	6.65	7
6	122	0.03	0.98	6.86	7
7	81	0.02	1.00	7.00	7

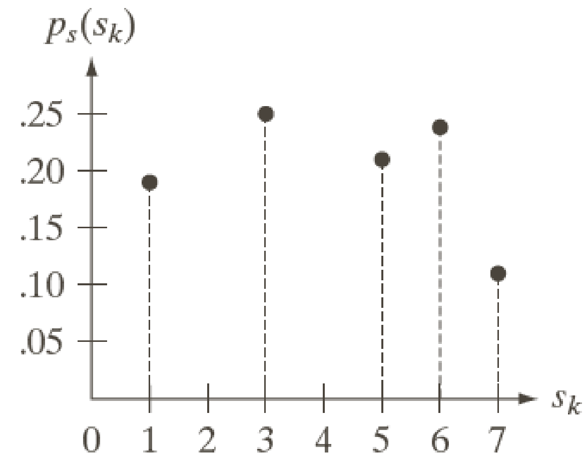
In Practice



Input histogram



$T(r_k)$, scaled back to 0 - 7



Equalised histogram

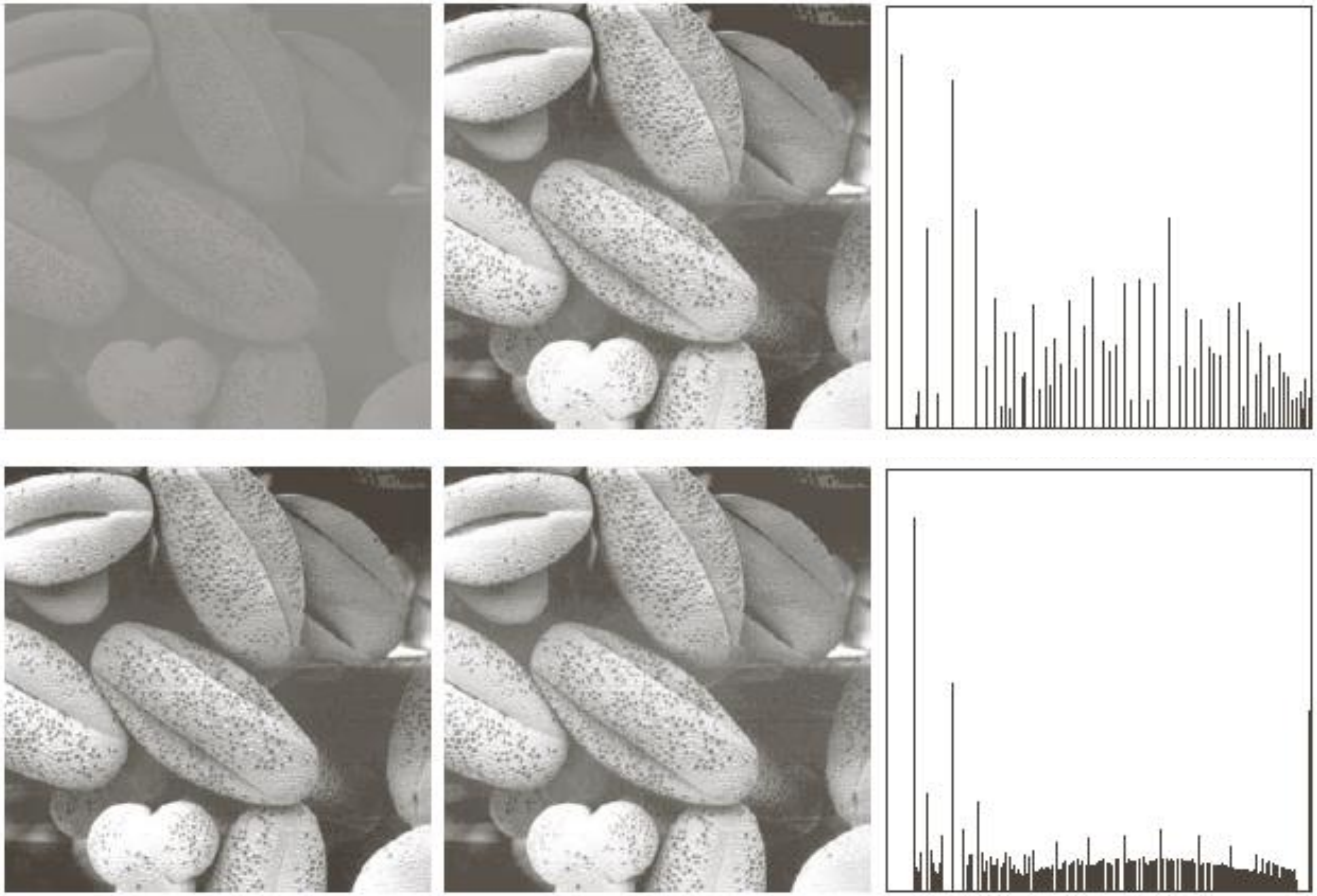
Results

- Light

- Dark

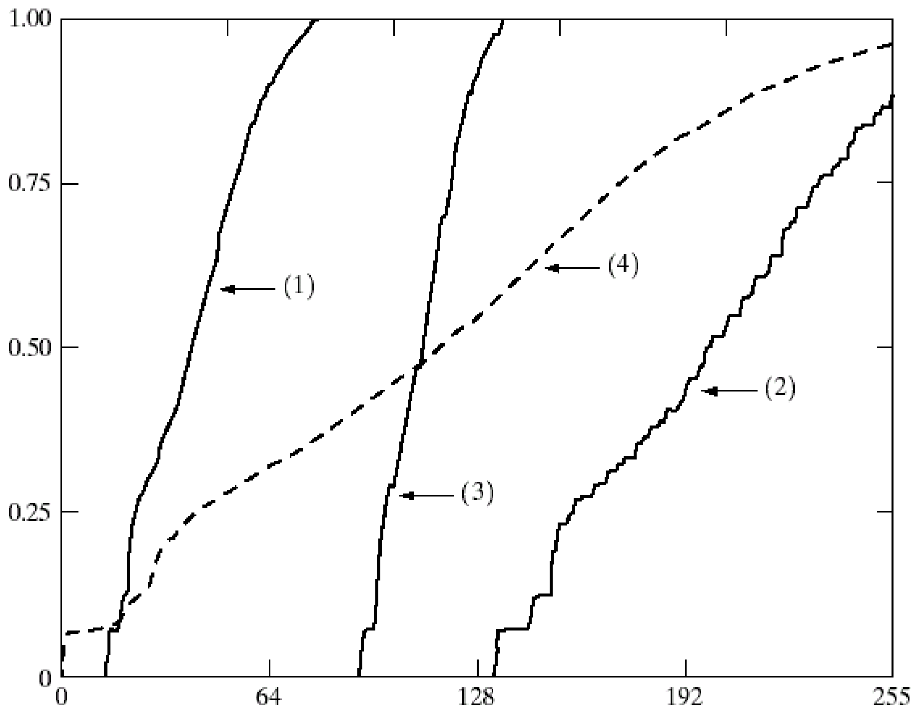
Results

- Low contrast
- High Contrast



Results

- Each of the four transformations above used a different transform, tuned to the input histogram



Strengths & Weaknesses

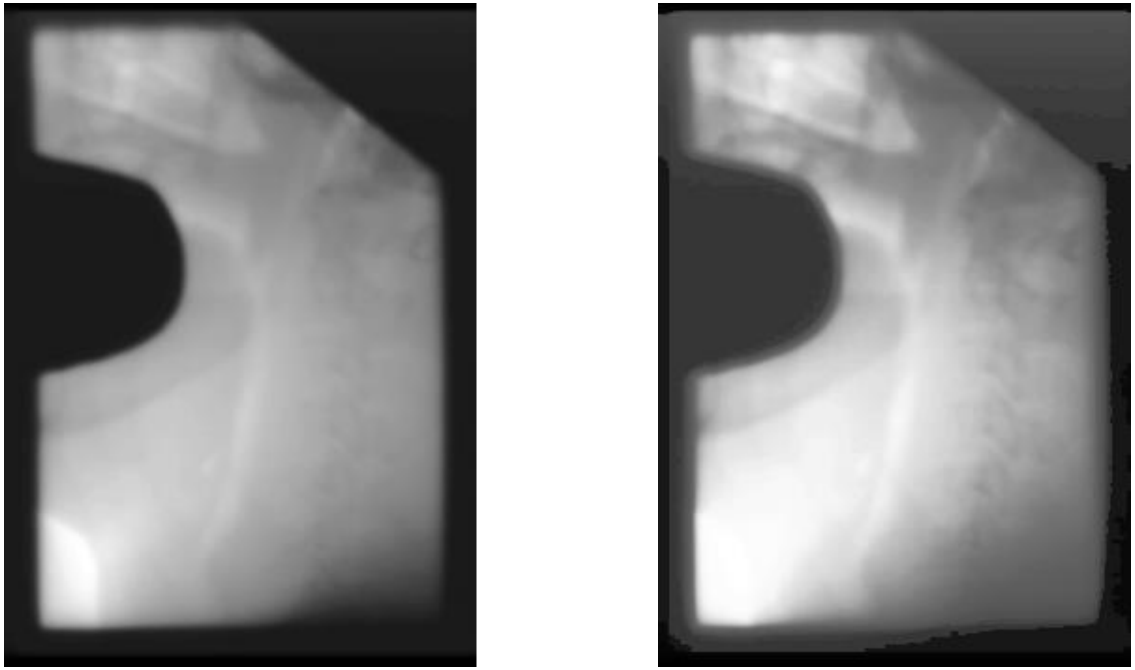
- Histogram equalisation works well when the input images
  - aren’t too noisy
  - don’t have large bright or dark areas

Strengths & Weaknesses

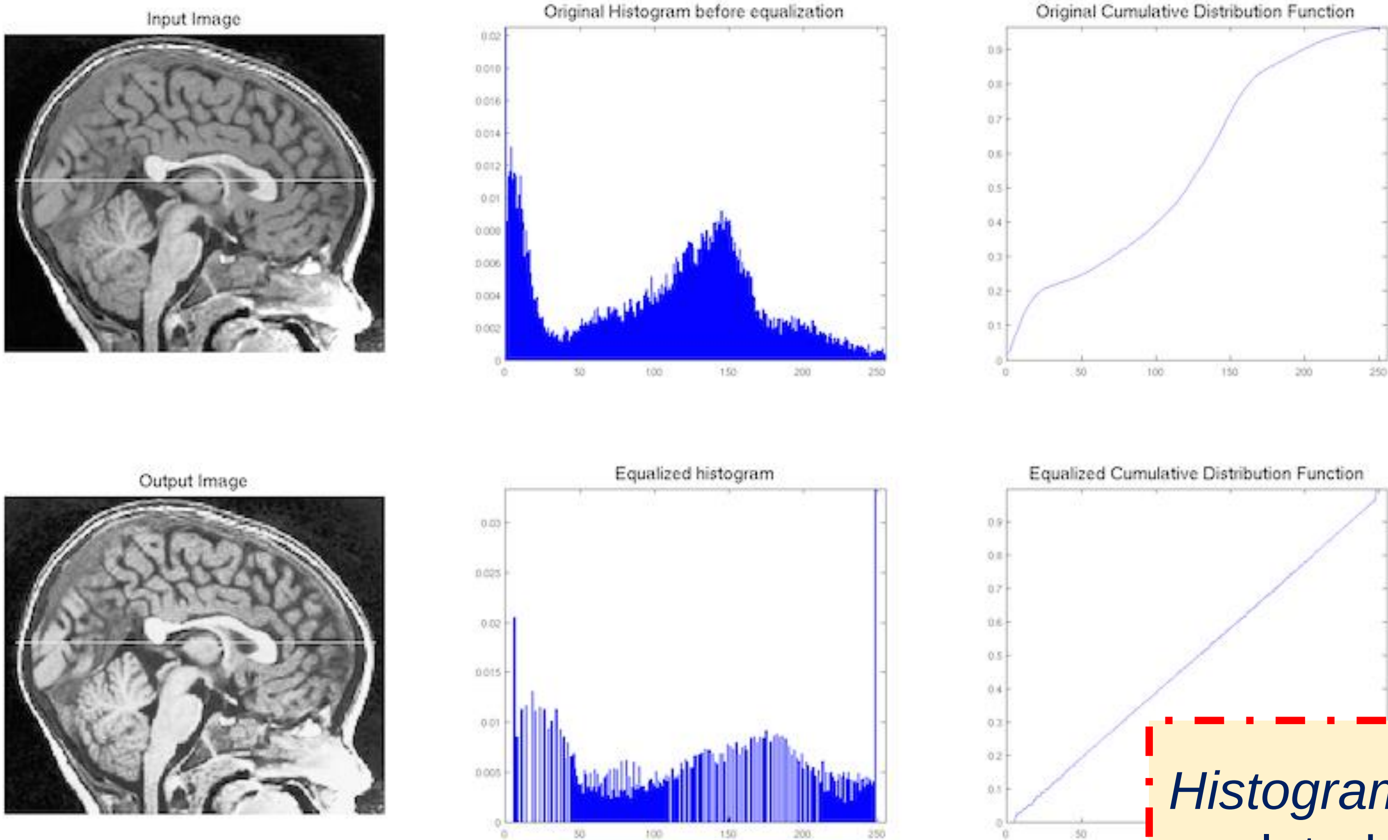
- Here the bright sky has dominated the process, equalisation has introduced an artificial boundary between sunlight and sky but not enhanced the three people

Strengths & Weaknesses

- Here the bright area of interest is enhanced, but the noise in the upper dark region is also more obvious



Medical Applications (MRI)



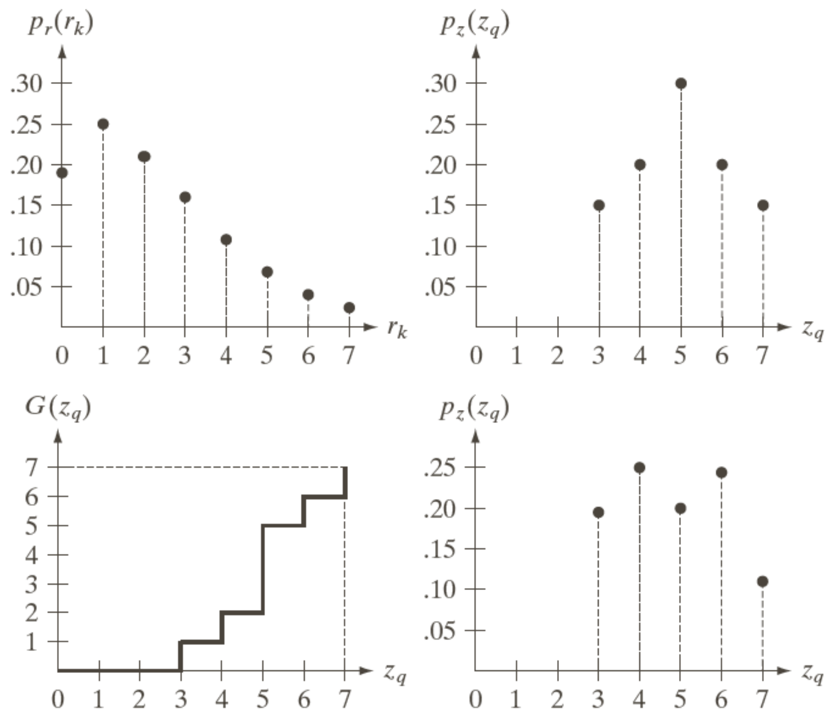
Histogram Specification is a related method which transforms an image's histogram so that it matches a target histogram

More Generally…

- *Histogram Specification* is a related method which transforms an image's histogram so that it matches a target histogram

Input histogram

Target histogram



Transform

Result

Conclusion

- Histograms, particularly normalised histograms, provide useful summary information
- Images can be manipulated by manipulating their histograms
- Histogram equalisation is a powerful and widely used image enhancement operation
- Global equalisation has drawbacks which can be addressed using local processing
- The method generalises to Histogram Specification

PART 2 : APPLICATIONS

Image Matching with Colour Histograms

Storing & Retrieving Images

- Given a large image database, find all the images containing e.g. horses
- We will focus on individual images, but many of the problems & methods discussed extend to video databases

Text-based Approaches

- Annotation: Relevant words/phrases are added to each image
- Retrieval is via text search
- But annotation is
  - subjective
  - laborious
  - unnecessary????

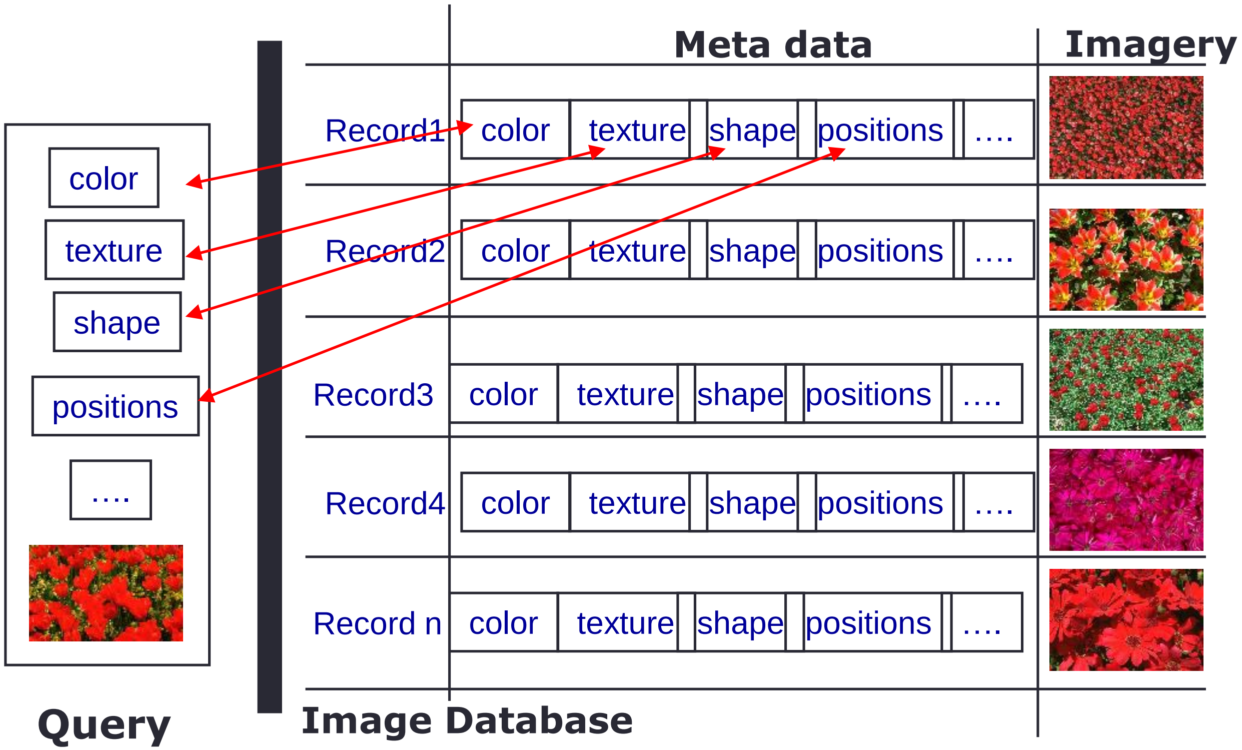
Mother, Child, Vegetable, Yellow, Green, Purple ….

Content-based Retrieval

- Indexes the image database on visual features
  - colour
  - shape
  - Texture
- Queries are expressed in those terms or via visual examples

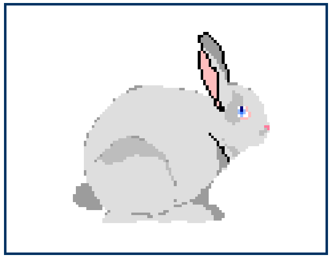
- Simple approaches compute metric distances between the query image and each image in the database
- Advanced approaches use AI techniques, machine learning, etc., and may be interactive
- Some simple measures and datastructures can be very useful

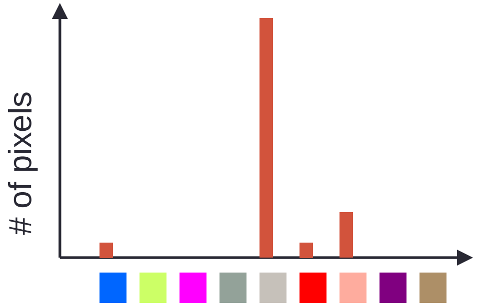
Content-based Retrieval



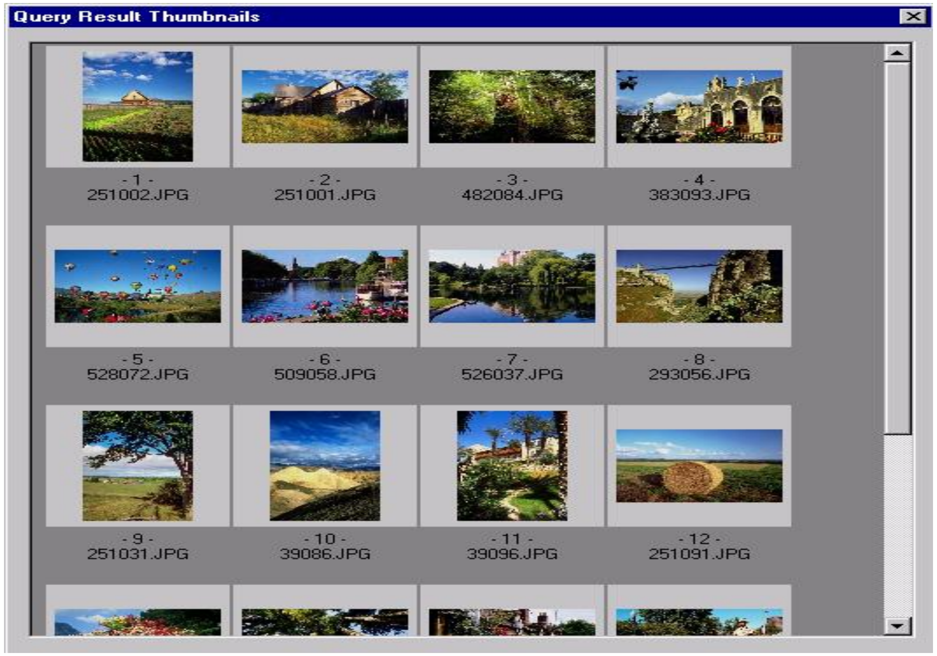
Colour Histograms

- Choose a colour space
  - RGB, HSV,,…
- Divide the axes to create a reasonable number of divisions
  - Trade-off detail against memory/computational cost
- Build a histogram
- Normalise if images are different sizes or colour resolution





Why Colour Histograms?



- Images with similar colour distributions look similar: colour distribution = colour histogram

Why Colour Histograms?

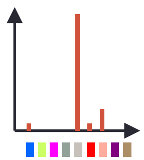
- Colour correlates well with class identity

- Human vision works hard to preserve colour constancy: presumably because colour is useful

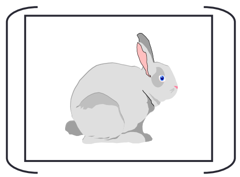
- Histograms
  - Are invariant to translation and rotation
  - Change slowly as viewing direction changes
  - Change slowly with object size
  - Change slowly with occlusion

- Colour histograms summarise target objects quite well, and should match a good range of images

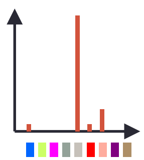
Colour Histograms



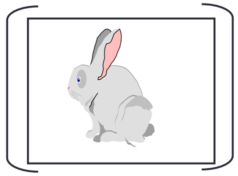
Colour Distribution



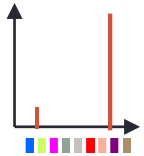
= (10,0,0,0,100,10,30,0,0)



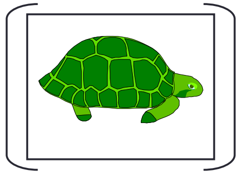
Colour Distribution



= (10,0,0,0,90,10,40,0,0)



Colour Distribution



= (0,40,0,0,0,0,0,0,110,0)

- Points in a high dimensional space or compact representations of images?

Common Distance Metrics

- Euclidean or straight-line distance or L2-norm, D^2

$$D^2(H^1,H^2)=\underbrace{\sqrt{\sum_i \left(H_i^1-H_i^2\right)^2}}_{\text{Root-mean square error}}=\left\|H^1-H^2\right\|_2$$

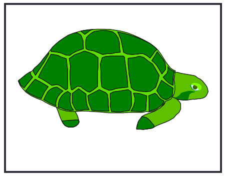
- City-block distance or L1-norm, D^1

$$D^1(H^1,H^2)=\sum_i \underbrace{\left|H_i^1-H_i^2\right|}_{\text{sum of absolute differences}}=\left\|H^1-H^2\right\|_1$$

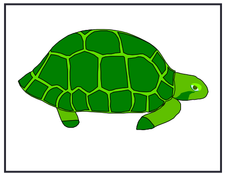
- *How far apart are two points?*

Some Problems

- Colour quantisation
  - Noise and/or different camera responses can give similar images very different histograms



(0,40,0,0,0,0,0,110,0)

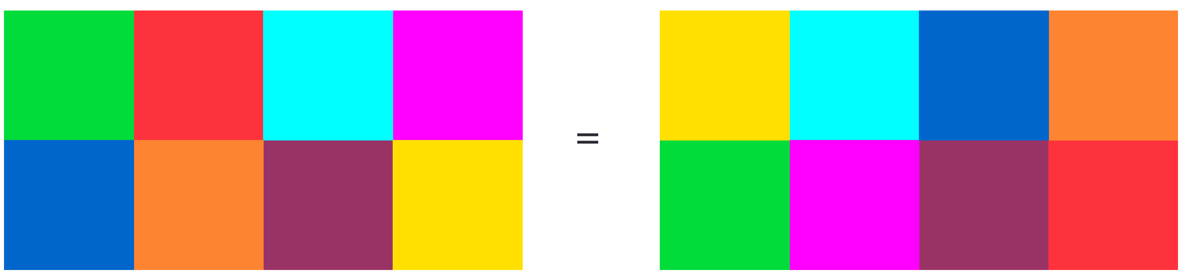


(0,0,40,0,0,0,0,0,110)

- Histogram resolution
  - May need many bins (4096) to accurately store colour distributions
  - Expensive
- The illumination may be coloured
  - Same object may generate a different histogram under different lighting

Some Problems

- Colour histograms ignore spatial information
- More advanced methods take spatial relationships into account



- But the comparatively simple image processing methods and representations methods covered so far can do useful things

Histogram Intersection

- Measures how much of the query may be present in the target image (and vice-versa)
- A bin in the target histogram can have a larger value than the corresponding query bin (and vice-versa)

$$HI(H^1, H^2) = \sum_{i=1}^{n} \min(H_i^1, H_i^2)$$

$H^1 = (10, 0, 0, 0, 100, 10, 30, 0, 0)$

$H^2 = (0, 40, 0, 0, 0, 6, 0, 110, 0)$

$HI(H^1, H^2) = 0 + 0 + 0 + 0 + 0 + 6 + 0 + 0$
$= 6$

- *How much do two representations overlap?*

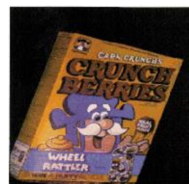
Histogram Intersection

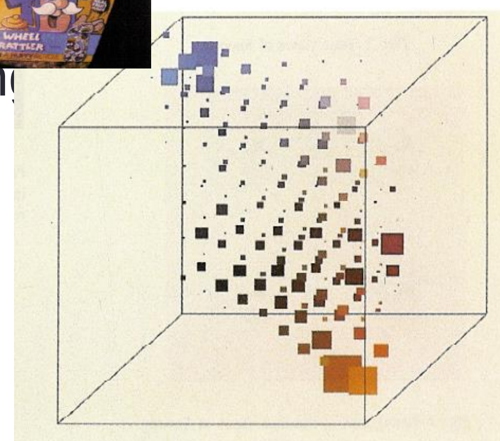
- In the first histogram intersection paper (Ballard and Swain 1991):
  - A database of 66 colour histograms
  - 32 query images
- Recognition rate was almost 100%

?

Histogram Intersection

- Ballard and Swain used opponent colour axes
  - RG = R – G
  - BY = 2 * B – R – G
  - WB = R + G + B (intensity)
- & matched images under a range of conditions
  - Normal condition
  - Varying in view
  - Varying in image resolutions
  - Occlusion (of bottom 1/3 and/or side 1/3 of image)
  - Varying in bin resolutions
  - Varying in light intensity





Histogram Intersection

Condition	Placement			
	1st	2nd	3rd	>3rd
Full size	29	3	0	0
128 x 90	29	3	0	0
64 x 45	27	5	0	0
32 x 22	14	7	1	0
16 x 11	15	6	4	0
8 x 5	4	4	3	21
Bottom occluded	27	4	1	0
Bottom & side occluded	22	6	5	0

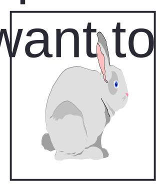
← Good until images are very small

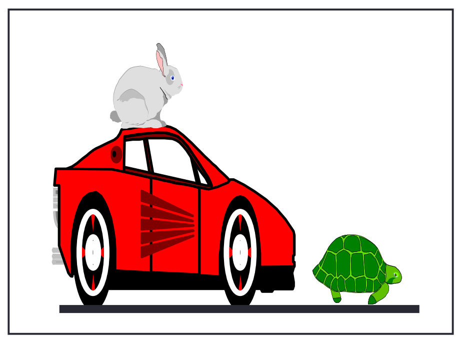
← Not bad when object only partly visible

Using Histograms: Object Location

Object Location

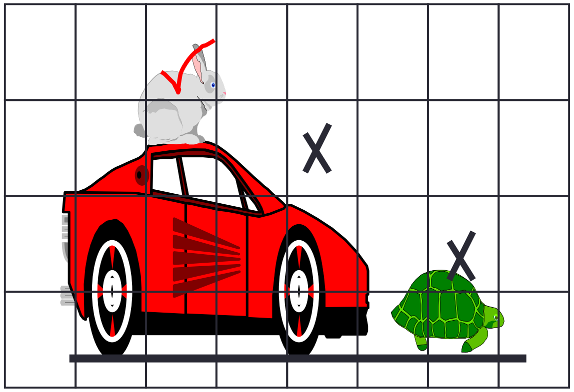
- Matching whole images isn't always appropriate…..
  - e.g. if the target object is only expected to fill part of the image or you want to know where it is





Region/Object-based Queries

- We know the rabbit is mostly grey
- Divide the image into windows and see how grey each window is
  - Highlight pixels in the image that are similar to those in the query
  - Look for regions with lots of these pixels



✓ Similar colours

✗ Dissimilar colours

The Histogram Ratio

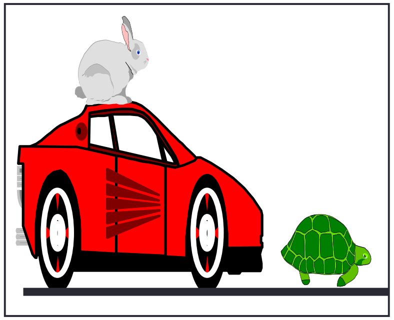
- BUT: the image is usually much bigger than the query region
  - Other objects in the image may be the same colour as parts of the query object
  - So some colours are not reliable cues: if you're looking for a zebra at night look for white pixels

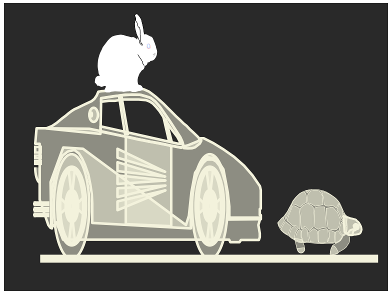
$$R_j = \min\left(\frac{M_j}{I_j}, 1\right)$$

- Compute the ratio of corresponding Model and Image histogram bins
- If the image has many more pixels of a given colour R_j is small and that colour is not useful
- If the model has more, R_j is 1 and that colour is useful

Backprojection

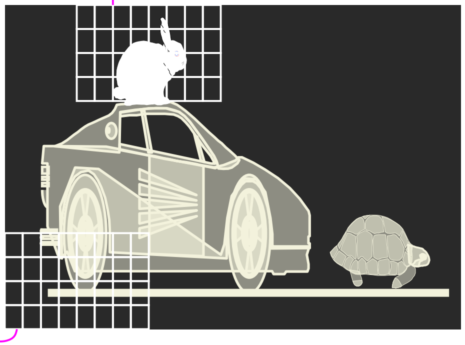
- The greater the value of $R_{j,}$ the more valuable the colour(s) represented by bin j
  - Consider each image pixel
  - If that pixel maps to histogram bin k, replace the pixel value with a grey value = R_k
- This is still image processing: we've processed an image to create an image
  - The pixel values are related to the likelihood of their showing the target





Backprojection

High average brightness



Low average brightness

- Regions with high average brightness are likely to contain the rabbit – its more complex, but still a histogram matching approach (see Ballard and Swain)

We’ve seen Backprojection before

- Skin shows a very clear peak and narrow range in U,V