Computational Vision
U. Minn. Psy 5036
Daniel Kersten
Lecture 5

■ **Initialize standard library files:**

```
<<Graphics`Graphics`
<<Statistics`DiscreteDistributions`
<<Statistics`ContinuousDistributions`
Off[General::spell1];
```

Goals

Last time

■ Ideal Observer Analysis: Essential idea

Ideal observer

Model the data (image) generation process

Define the inference task

Determine optimal performance

Compare human performance to the ideal

Ideal normalizes for information available

Explain discrepancies in terms of:

functional adaptation

mechanism

Today

Review some probability and statistics

Pattern detection: The signal-known-exactly ideal

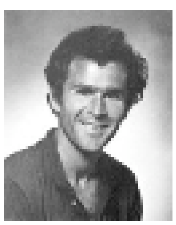
Demo of 2AFC for pattern detection in noise

What does the eye see best?

Make the question precise by asking:

For what patterns does the human visual system have the highest detection efficiencies relative to an ideal observer?

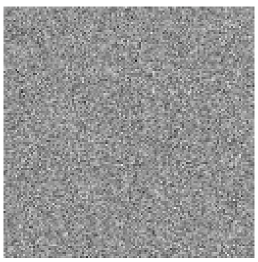
- **Faces, or a particular face?**



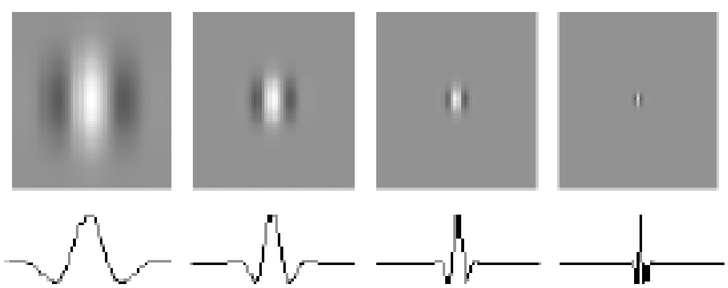
- **Animals, or particularly dangerous ones?**

- **Or something simple, like a spot?**

- **Or something complex, like a "frozen" noise image?**



- **Or some pattern motivated by neurophysiology? E.g. the kinds of spatial patterns preferred by single neurons in the primary visual cortex ...**



Probability Overview

For terminology, a fairly comprehensive outline, and overview, see notebook: ProbabilityOverview.nb in the syllabus web page.

For the section below, we'll use the properties of independence.

■ Expectation & variance

Analogous to center of mass:

Definition of expectation or average:

$$\text{Average}[X] = \bar{X} = E[X] = \Sigma\, x[i]\, p[x[i]] \sim \sum_{i=1}^{N} x_i / N$$

$$\mu = E[X] = \int x\, p(x)\, \mathrm{dx}$$

Some rules:

E[X+Y]=E[X]+E[Y]

E[aX]=aE[X]

E[X+a]=a+E[X]

Definition of variance:

σ^2 = Var[X] = E[[X-μ]^2] = $\sum_{j=1}^{N} ((p(x(j)))\,(x(j)-\mu))^2 = \sum_{j=1}^{N} p_j (x_j - \mu))^2$

$$\text{Var}[X] = \int (x-\mu)^2\, p(x)\, \mathrm{dx} \sim \sum_{i=1}^{N} (x_i - \mu)^2 / N$$

Standard deviation:

$$\sigma = \sqrt{\text{Var}[X]}$$

Some rules:

$$\text{Var}[X] = E[X^2] - E[X]^2$$
$$\text{Var}[aX] = a^2\, \text{Var}[X]$$

■ Statistics for independent random variables

Independence means that knowledge of one event doesn't change the probability of another event.

p(X)=p(X|Y)

p(X,Y)=p(X)p(Y)

If p(X,Y)=p(X)p(Y), then

$$E[X\,Y] = E[X]\,E[Y] \text{ (uncorrelated)}$$

$$\text{Var}[X+Y] = \text{Var}[X] + \text{Var}[Y]$$

Ideal pattern detector for a signal which is exactly known ("SKE" ideal)

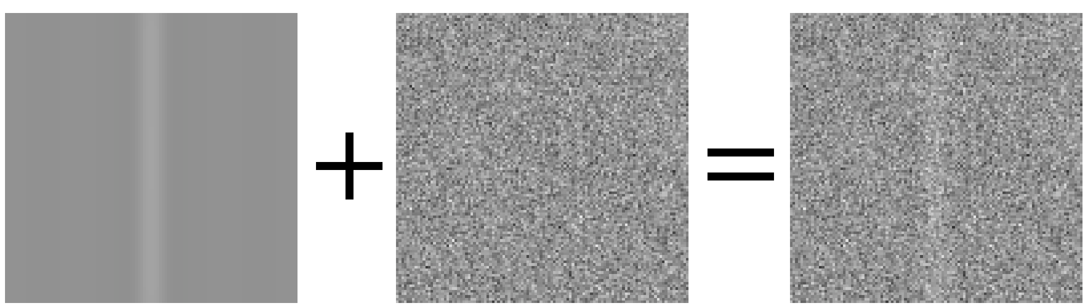
In this notebook we will study an ideal detector called the signal-known-exactly ideal (SKE). This detector has a built-in template that matches the signal that it is looking for. The signal is embedded in "white gaussian noise". "white" means the pixels are not correlated with each other--intuitively this means that you can't reliably predict what one pixel's value is from any of the others. Assignment 1 simulates the behavior of this ideal. In the absence of any internal noise, this ideal detector behaves as one would expect a linear neuron to behave when a target signal pattern exactly matches its synaptic weight pattern. There are some neurons in the the primary cortex of the visual system called "simple cells". These cells can be modeled as ideal detectors for the patterns that match their receptive fields. In actual practice, neurons are not noise-free, and not perfectly linear.

Calculating the Pattern Ideal's d' based on signal-to-noise ratio

■ The signal + gaussian noise generative model

x = s + n, where s is a vector of image intensities, e.g. corresponding to a face, snake, spot, ...or a gabor pattern

x = n, where n is white gaussian noise



■ Overview

We are going to do two things:

1. Show that a simple decision variable for detecting a known fixed pattern in white gaussian noise is the dot product, or cross-correlation, of the observation image **x** with the known signal image **s**.

r = **x•s** , or alternatively written as

$$r = \sum_{i=1}^{N} x(i)\, s(i)$$

2. Show that d' is given by:

$$d' = \frac{\sqrt{s \bullet s}}{\sigma}$$

s and **x** are a vectors, i.e. lists, of the image intensities, and σ is the standard deviation of the added gaussian noise.

■ 1. Cross correlation produces an ideal decision variable: Proof

What is the optimal decision variable? Starting from the maximum a posteriori rule, we saw that basing decisions on the likelihood ratio is ideal, in the sense of minimizing the probability of error. So the likelihood ratio is a decision variable. But it isn't the only one, because any monotonic function is still optimal. So our goal is to pick a decision variable which is simple, intuitive, and easy to compute. But first, we need an expression for the likelihood ratio:

$$\frac{p(x \mid \textbf{signal plus noise})}{p(x \mid \textbf{noise only})} \tag{1}$$

where **x** is the vector representing the image measurements actually observed

x = s + n, under signal plus gaussian noise condition

x = n, under gaussian noise only condition

Consider just one pixel of intensity x. Under the signal plus noise condition, the values of x fluctuate about the average signal intensity s with a Gaussian distribution (**gp[ ]**) with mean s and standard deviation σ.

So under the signal plus noise condition, the likelihood **p[x|s]** is the **gp[x-s; σ]**:

```
gp[x_,s_,σ_]:= (1/(σ*Sqrt[2 Pi])) Exp[-(x-s)^2/(2 σ^2)]
```

```
gp[x,s,σ]
```

$$\frac{e^{-\frac{(x-s)^2}{2\sigma^2}}}{\sqrt{2\pi}\,\sigma}$$

Again, consider just one pixel of intensity x. Under the noise only condition, the values of x fluctuate about the average intensity corresponding to the mean of the noise, which we assume is zero.

So under the noise only condition, the likelihood p[x|n] is:

```
gp[x,0,σ]
```

$$\frac{e^{-\frac{x^2}{2\sigma^2}}}{\sqrt{2\pi}\,\sigma}$$

But we actually have a whole pattern of values of x, which make up an image vector **x**. So consider a pattern of image intensities represented now by a vector **x = {x[1],x[2],...x[N]}**. Let the mean values of each pixel under the signal plus noise condition be given by vector **s = {s[1],s[2],...,s[N]}**. The joint probability of an image observation **x**, under the signal hypothesis is:

```
Product[gp[x[i],s[i],σ],{i,1,N}]
```

$$\prod_{i=1}^{N} \frac{e^{-\frac{(x(i)-s(i))^2}{2\sigma^2}}}{\sqrt{2\pi}\,\sigma}$$

This is because we are assuming independence. In general, whether we can assume independence depends on the problem. In our case, the samples are independent by definition--as "experimenters" we generate the noise as independent samples.

Independence between pixels means we can multiply the individual probabilities to get the global joint image probability. (See **ProbabilityOverview.nb**)

The joint probability of an image observation **x**, under the noise only hypothesis is:

```
Product[gp[x[i],0,σ],{i,1,N}]
```

$$\prod_{i=1}^{N} \frac{e^{-\frac{x(i)^2}{2\sigma^2}}}{\sqrt{2\pi}\,\sigma}$$

Now we have what we need for the likelihood ratio:

```
Product[gp[x[i], s[i], σ], {i, 1, N}] / Product[gp[x[i], 0, σ], {i, 1, N}]
```

$$\frac{\prod_{i=1}^{N} \frac{e^{-\frac{(x(i)-s(i))^2}{2\sigma^2}}}{\sqrt{2\pi}\,\sigma}}{\prod_{i=1}^{N} \frac{e^{-\frac{x(i)^2}{2\sigma^2}}}{\sqrt{2\pi}\,\sigma}}$$

So at this point, we could just stop and use this product to make ideal decisions. E.g. if the product is bigger than 1, choose the signal hypothesis, and if less than 1 choose the noise hypothesis. But we can get a much simpler rule with a little more work.

This is because any monotonic function, **f()** of the likelihood ratio would give the same performance (i.e. choose signal if **f(likelihood ratio)>f(1)**, and noise otherwise), let's try one--the natural logarithm will turn the product into a sum:

```
Log[ ∏_{i=1}^N gp[x[i], s[i], σ] / ∏_{i=1}^N gp[x[i], 0, σ] ]
```

$$\log\left(\frac{\prod_{i=1}^{N} \frac{e^{-\frac{(x(i)-s(i))^2}{2\sigma^2}}}{\sqrt{2\pi}\,\sigma}}{\prod_{i=1}^{N} \frac{e^{-\frac{x(i)^2}{2\sigma^2}}}{\sqrt{2\pi}\,\sigma}}\right)$$

which is equal to:

$$\text{Log}\left(\prod_{i=1}^{N} \frac{e^{-\frac{(x(i)-s(i))^2-x(i)^2}{2\sigma^2}}}{\sqrt{2\pi}\,\sigma}\right) \tag{2}$$

which is monotonic with:

$$\text{Log}\left[\prod_{i=1}^{N} e^{\frac{2\,x(i)\,s(i)}{2\sigma^2}}\right] \tag{3}$$

which simplifies to

$$(1/\sigma^2)\sum_{i=1}^{N} x(i)\,s(i) \tag{4}$$

which is monotonic with:

```
Sum[x[i] s[i],{i,1,N}]
```

$$\sum_{i=1}^{N} s(i)\,x(i)$$

$$r = \sum_{i=1}^{N} x(i)\,s(i) \tag{5}$$

In other words, we've proven that the dot product, r, (or cross-correlation or matched filter) provides a decision variable which is optimal--in the sense that if we use the rule, the probability of error will be least. Now, let's calculate d'.

■ 2. Derive formula for d'

By definition

```
d'=(μ2 - μ1)/σ
```

$$\frac{\mu 2-\mu 1}{\sigma}$$

where u2 is the mean of the decision variable,r,under the signal hypothesis,and u1 is the mean under the noise-only hypothesis.

To get d', we need formulas for the means and standard deviation for the decision variable, r under the two hypotheses, signal plus noise vs. noise only.

First, suppose the switch is set for signal trials. What is the average and standard deviation of r?

$$\mu 2 = \text{Average}[r] = \text{Average}\left[\sum_{i=1}^{N} x(i)\, s(i)\right] = \sum_{i=1}^{N} \text{Average}[x(i)]\, s(i) = \sum_{i=1}^{N} s(i)\, s(i) = \sum_{i=1}^{N} s(i)^2 \tag{6}$$

$$\mu 2 = \sum_{i=1}^{N} s(i)^2 \tag{7}$$

(Because x(i)=s(i)+n(i), Average[x(i)]=s(i).)

And the variance is:

$$\text{Var}\left(\sum_{i=1}^{N} x(i)\, s(i)\right) = \sum_{i=1}^{N} s(i)^2\, \text{Var}[x(i)] = \sigma^2 \sum_{i=1}^{N} s(i)^2 \tag{8}$$

(Because Var[Y + Z] = Var[Y] + Var[Z], but one is a constant, so because Var[constant + n]=Var[n].

Also, recall that Var[c Y] = c^2 Var[Y])

Now, suppose the switch is set for noise only trials. The average is:

$$\mu 1 = \text{Average}[r] = \text{Average}\left[\sum_{i=1}^{N} x(i)\, s(i)\right] = \sum_{i=1}^{N} \text{Average}[x(i)]\, s(i) = \sum_{i=1}^{N} 0\, s(i) = 0 \tag{9}$$

The variance is the same as for the signal case:

$$\text{Var}\left(\sum_{i=1}^{N} x(i)\, s(i)\right) = \sum_{i=1}^{N} s(i)^2\, \text{Var}[x(i)] = \sigma^2 \sum_{i=1}^{N} s(i)^2 \tag{10}$$

So d' is:

```
Sum[s[i]^2, {i, 1, N}]/Sqrt[(σ^2 Sum[s[i]^2,{i,1,N}])]
```

$$\frac{\sum_{i=1}^{N} s(i)^2}{\sqrt{\sigma^2 \sum_{i=1}^{N} s(i)^2}}$$

```
FullSimplify[Sum[s[i]^2, {i, 1, N}]/Sqrt[(σ^2 Sum[s[i]^2,{i,1,N}])],σ>0]
```

$$\frac{\sqrt{\sum_{i=1}^{N} s(i)^2}}{\sigma}$$

Or:

$$d' = \frac{\sqrt{\sum_{i=1}^{N}\ s(i)^2}}{\sigma} = \frac{\sqrt{s.s}}{\sigma} \tag{11}$$

Calculating the Pattern Ideal's d' for a two-alternative forced-choice experiment from a z-score of the proportion correct.

Recall that we had an expression for d' for a yes/no experiment in which we measured hit and false alarm rates.

We've seen the expression for d' for a 2AFC experiment in an earlier lecture, but I'll repeat it here.

For a 2AFC experiment, the observer gets two images to compare. One has the signal plus noise, and the other just noise. But the observer doesn't know which one is which. An ideal strategy is to compute the cross-correlation decision variable for each image, and pick the image which gives the larger cross-correlation. This strategy will result in a single number, the proportion correct, Pc.

d' for a 2AFC task is given by the formula:

$$d' = -\sqrt{2}Z(P_c)$$

$$P = \frac{1}{\sqrt{2\pi}}\int_z^\infty e^{-x^2/2}\,dx$$

You can use the inverse of a standard mathematical function called Erf[] to get Z from a measured P.

```
z[p_] := Sqrt[2] InverseErf[1 - 2 p];
```

where Z(*) is the z-score for P_c , the proportion correct.

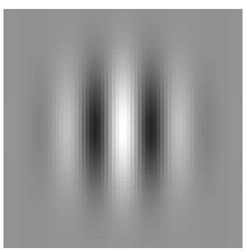
```
dprime[x_] := N[-Sqrt[2] z[x]]
```

Demo of 2AFC for pattern detection

So what can you do with this particular ideal observer analysis? Take a look at:

Burgess, A. E., Wagner, R. F., Jennings, R. J., & Barlow, H. B. (1981). Efficiency of human visual signal discrimination. Science, 214, 93-94.

vs.



The signal + gaussian noise generative model

x = s + n, where s is a vector of image intensities corresponding to a gabor pattern

x = n, where n is white gaussian noise

Gabor patterns as signals

■ Basis set: Cartesian representation of Gabor functions:

```
ndist=NormalDistribution[0,1];

cgabor[x_,y_, fx_, fy_,sx_,sy_] :=
Exp[-((x/sx)^2 +  (y/sy)^2)] Cos[2 Pi(fx x + fy y)];
```

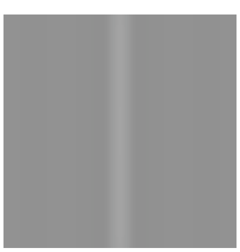
■ Various frequencies , vertical orientations, and fixed width

```
vtheta = Table[0, {i1,4}];
vf = {2,4};
hf = {0.0,0.0,0.0};
xwidth = {0.15,4};
ywidth = {4,4};
npoints = 128;
signalcontrast=0.15;
noisecontrast=0.2;
```

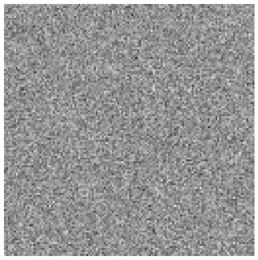
```
lr = -1; ur = 1; step = (ur - lr) / (npoints - 1);
signal =
  Table[signalcontrast cgabor[y, x, vf[[1]], hf[[1]], xwidth[[1]],
     ywidth[[1]]], {x, lr, ur, step}, {y, lr, ur, step}];
noise = noisecontrast Table[Random[ndist], {npoints}, {npoints}];
```

■ Signal, noise, signal + noise

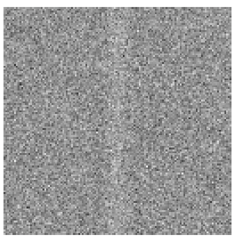
```
spn = ListDensityPlot[signal, Mesh → False, Frame → False,
   PlotRange → {-1, 1}];
```



```
spn = ListDensityPlot[noise, Mesh → False, Frame → False,
   PlotRange → {-1, 1}];
```



```
spn = ListDensityPlot[signal + noise, Mesh → False, Frame → False,
   PlotRange → {-1, 1}];
```



See GaborSKEDetection.nb.

Next time

High-level vision as Bayesian decision theory

- **Introduction to higher-level perceptual decisions as inference**

- **Bayesian decision theory**

- **Various types of inference Tasks: synthesis, inference (detection, classification, estimation), learning**

Exercises

Statistical Sampling

Earlier we used statistical sampling or "random number generation". We have done simple "Monte Carlo" simulations to make a histogram for detected photons. Here, you have a chance to get a closer look at the process.

Most standard programming languages come with standard subroutines for doing pseudo-random number generation. Unlike the Poisson or Gaussian distribution, these numbers are **uniformly distributed--**that is, the probability of being a certain value is the same over the sampling range. *Mathematica* comes with a standard function, **Random[]** that enables us to generate random numbers that are uniform, Poisson, Normal. (There are some others in the packages too, like the **ChiSquareDistribution**).

We did a "Monte Carlo" simulation of a photon absorption experiment. To do this, we needed to generate samples from a Poisson distribution, rather than a uniform distribution. *Mathematica* enables us to do this with the function specification **Random[pdist]**.

```
??Random
```

```
Random[ ] gives a uniformly distributed pseudorandom Real in the range
   0 to 1. Random[type, range] gives a pseudorandom number of the
   specified type, lying in the specified range. Possible types
   are: Integer, Real and Complex. The default range is 0 to 1. You
   can give the range {min, max} explicitly; a range specification
   of max is equivalent to {0, max}. Random[distribution] gives a
   random number with the specified statistical distribution. More…

Attributes[Random] = {Protected}
```

■ Exercise--make your own random number generator for non-gaussian sampling

If you wanted to write your own routine to sample from a non-uniform distribution, you could do it by first generating a table with the cumulative distribution function (whose probability values increase monotonically from zero to one), and then using Random[] to select a probabililty between zero and one, and then "reading across and down" your cumulative plot, to read off the value of the random variable--this is your sample. Try it for the Poisson distribution. Compare it with *Mathematica*'s add-in routines.

In the last lecture, we defined a function to generate **ntimes** samples, and then make a list of a 1000 values. Then do the sampling experiment to get the list. A thousand values can take a while to generate, so when you try this line below, you may want to start with 100 samples or so. Count up how many times the result was 20 or less. To do this, we will use two built-in functions: **Count[]**, and **Thread[]**. You can obtain their definitions using the ?? query.

```
sample[ntimes_] :=
    Table[Random[pdist],{ntimes}];
    z = sample[1000];
    Count[Thread[z<=20],True]
```

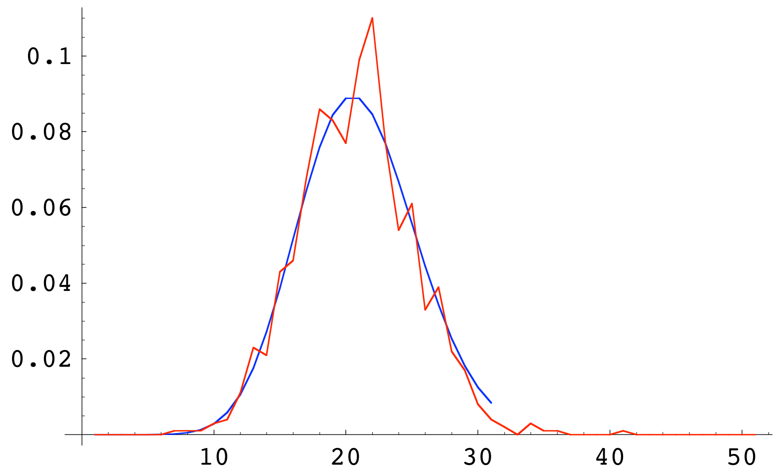
```
567
```

So far, are we in good agreement with theory?--about half (500/100) of the samples should be less than 20. We can make a better comparison by comparing the plots of the histogram from the sampling experiment with the theoretical prediction. Let's make a table with that summarizes the frequency.

```
domain = Range[0,50];
Freq = Map[Count[z,#]&,domain];
```

Now plot up the results. Note that we normalize the Freq values by the number values in z using **Length[]**.

```
theoreticalz = Table[PDF[pdist,n], {n,0,30}];
theoreticalg = ListPlot[theoreticalz,
        PlotJoined->True, PlotStyle->{RGBColor[0,0,1]},
        DisplayFunction->Identity];
simulatedg = ListPlot[N[Freq/Length[z]],
        PlotJoined->True, PlotStyle->{RGBColor[1,0,0]},
        DisplayFunction->Identity];
        Show[theoreticalg,simulatedg,
    DisplayFunction->$DisplayFunction];
```



As you can see (and saw in an earlier lecture), the computer simulation matches fairly closely what theory predicts. The next exercise shows that the mean and variance are in good agreement, that is if you've sampled enough times.

Exercise: Use the functions `Apply[Plus,list]` and **Length[list]** to define a function that calculates the mean of a list. Find the mean of z. Then define a function to calculate the variance of z. Compare the mean and variance of the above Poisson sampling experiment to the theoretical prediction.

■ Extra notes

By the way, you can look at the two distributions in table form. Let us first get the cumulative frequency using the **FoldList[]** function,

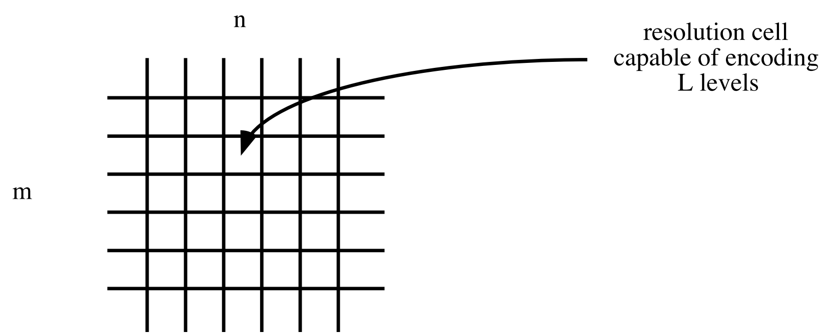
```
CumFreq = FoldList[Plus, 0, Freq]//Rest;
```

and then outputing the data in "MatrixForm":

```
Transpose[{domain,Freq, CumFreq}]//MatrixForm
```

Exercise: Calculate the information capacity of the eye

Consider an m x n pixel image patch. Is there a quantum limit to the number of light levels that can be represented in a resolution cell? (The size of a resolution cell is determined by the modulation transfer function of the optical device under consideration, which in this case would be the eye. We look later at how to estimate the spatial resolution of an imaging system).



Let SN be the maximum number of photons that land in a resolution cell. One can't discriminate this level from any other with an infinitely small degree of precision. Requiring a sensitivity of d', determines the next dimmest light level:

$$S_{N-1} = S_N - d'\sqrt{S_N}$$

This effectively quantizes the dynamic range of a resolution cell. Write a small iterative program to count the number of levels down to S1 = zero. Say the number of levels is L, or $\mathrm{Log}_s L = l$ bits. Of course, one has to decide a priori what is a suitable discrimination level. But once done, the information capacity can be estimated by *lmn* bits.

References

Applebaum, D. (1996). Probability and Information . Cambridge, UK: Cambridge University Press.

Burgess, A. E., Wagner, R. F., Jennings, R. J., & Barlow, H. B. (1981). Efficiency of human visual signal discrimination. Science, 214, 93-94.

Cover, T. M., & Joy, A. T. (1991). *Elements of Information Theory*. New York: John Wiley & Sons, Inc.

Duda, R. O., & Hart, P. E. (1973). Pattern classification and scene analysis . New York.: John Wiley & Sons.

Green, D. M., & Swets, J. A. (1974). Signal Detection Theory and Psychophysics . Huntington, New York: Robert E. Krieger Publishing Company.

Kersten, D. (1984). Spatial summation in visual noise. Vision Research, 24,, 1977-1990.

Ripley, B. D. (1996). *Pattern Recognition and Neural Networks*. Cambridge, UK: Cambridge University Press.

Schrater, P. R., Knill, D. C., & Simoncelli, E. P. (2000). Mechanisms of visual motion detection. *Nat Neurosci, 3*(1), 64-68.

Van Trees, H. L. (1968). Detection, Estimation and Modulation Theory . New York: John Wiley and Sons.

Watson, A. B., Barlow, H. B., & Robson, J. G. (1983). What does the eye see best? Nature, 31,, 419-422.