This material starts chapter 5 in the text. Please read Chapter 5 and Chapter 6 over the next 3-4 lectures.

Two-dimensional Convolution

Convolution is important as an expression of how we relate input and output signals from systems that are reasonably linear and time-invariant. Most systems satisfy these constraints to at least some degree, thus convolution is an invaluable tool in analyzing them.

In 2-D, we often replace time-invariance with space invariance. In other words, the impulse response or transfer function of the system does not depend on location within the x-y plane.

Correlation and autocorrelation are derived from statistical processing concepts, but are closely related to the mathematics of convolution. We will consider them together with convolution.

Definition of Convolution

$$h(x) = \int_{-\infty}^{\infty} f(x-x')\,g(x')\,dx$$

or, simply $h(x) = f(x) * g(x)$, where we use the $*$ symbol to denote the convolution integral above.

Convolution also exists in the discrete sample case, in which instance we use the sum notation

$$h_i = \sum_{j=-\infty}^{\infty} f_j\, g_{i-j}$$

We can remember the signs on the arguments in these cases by seeing that we "integrate out" the dummy variable which appears in the argument to both functions:

$$h(x) = \int \underset{-}{f(x - x')}\; \underset{+}{g(x')}\; \underset{\text{variable of integration}}{dx'} \qquad \text{or} \qquad h_i = \underset{\text{var. of int.}}{\sum_j} \underset{+}{f_j}\, \underset{-}{g_{i-j}}$$

Also, note the expansion of the summation definition:

$$h_i = \cdots + f_2 g_{i-2} + f_1 g_{i-1} + f_0 g_i + f_{-1} g_{i+1} \cdots$$

sum of indices $= i$

For limited length sequences, we can either use the truncated series

$$h_i = \sum_{j=1}^{N} f_j g_{i-j}$$

with zero padding or some other rule for the end members, or we can use cyclic convolution

$$h_i = \sum_{j=1}^{N} f_j \, g_{(i-j) \bmod N}$$

The latter is sometimes seen as $h = f \circledast g$. Cyclic convolution is very important when we deal with truncated Fourier series, such as in FFT algorithms.

In two dimensions

Extension to two dimensions is straight-forward:

$$h(x,y) = \iint_{-\infty}^{\infty} f(x-x', y-y')\, g(x',y')\, dx'\, dy'$$

or $\quad h(x,y) = f(x,y) ** g(x,y)$

Since sometimes we want to convolve one 2-D function with another in 1 dimension only, to eliminate confusion we'll adopt the two-asterisk notation whenever possible.

Sampled case: $\quad h_{ij} = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f_{i-k,\, j-l}\; g_{k,l}$

Definition of cross-correlation

$$c(x) = \int_{-\infty}^{\infty} f(x'-x)\, g(x')\, dx' = \int_{-\infty}^{\infty} f(x')\, g(x+x')\, dx'$$

$$c(x,y) = \iint_{-\infty}^{\infty} f(x'-x, y'-y)\, g(x',y')\, dx'\, dy'$$

$$= \iint_{-\infty}^{\infty} f(x',y')\, g(x+x', y+y')\, dx'\, dy'$$

The difference between convolution and cross-correlation lies in the sign of the argument for the first function $f(\ )$ in the integral

$$f(x-x') \quad \text{vs.} \quad f(x'-x)$$

$\uparrow$ Convolution $\qquad\qquad$ $\uparrow$ Cross-correlation

Discrete form for cross-correlation:

$$c_i = \sum_{j=-\infty}^{\infty} f_{j-i}\, g_j = \sum_{j=-\infty}^{\infty} f_j\, g_{i+j}$$

$$c_{ij} = \sum_{k,l=-\infty}^{\infty}\sum f_{k-i,\,l-j}\, g_{k,l} = \sum\sum f_{k,l}\, g_{i+k,\,j+l}$$

Bracewell uses the notation $\star$ to denote cross-correlation and distinguish it from convolution.

Note: cross-correlation is not commutative –

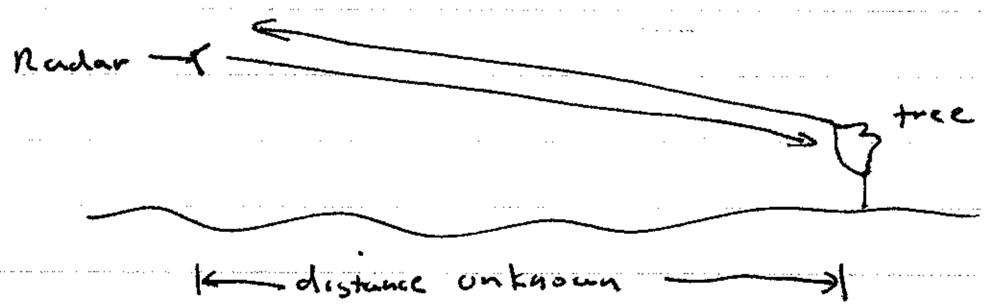
$$f \star g \neq g \star f$$

because of the choice of reversing <u>one</u> function with respect to the other. Sometimes we use the notation "f scans g" to imply that it is the function f that we reverse and slide over g to form the convolution integral.

In two-d, of course, reversal means reversal on both axes, or rotation by ½ turn.

<u>Application: Matched Filtering</u>

Suppose we want to detect a signal that is rather corrupted by system noise. We know the signal exists but not its location in time (or space, in the 2-D version). This is the situation often encountered in a radar system:



If the signal is transmitted with waveform $e(t)$, it returns at unknown time T_1, unknown amplitude A, and with additive noise $n(t)$, so that the received signal $s(t)$ is

$$s(t) = A\, e(t - T_1) + n(t)$$

T_1 is related to the distance d by $d = \frac{cT_1}{2}$, $c =$ speed of light.

The best estimate of the time delay is found by cross-correlating $s(t)$ with $e(t)$ and searching for the peak value:

$$e(t) \star s(t) = e(t) \star A\, e(t - T_1) + e(t) \star n(t)$$

The first term on the right will be a symmetrical peaked function centered at T_1. Matching the echo $s(t)$ with $e(t)$ through cross-correlation gives rise to the term matched filtering.

The 2-D analog is looking for a known shape in a noisy field of 2-D data, where the image is of form

$$s(x,y) = \sum_i A_i\, e(x - X_i, y - Y_i) + n(x,y)$$

Sum allows us to seach for all copies of the shape

Many feature detection algorithms use this principle, but in real life objects can appear quite different when viewed from other angles, and feature detection is usually a very tough problem.

Autocorrelation

Quite often we encounter autocorrelation rather than cross-correlation, especially in discussions about system impulse responses. Letting the two functions f and g in the previous section be equal, then

$$f(x) \star f(x) = \int_{-\infty}^{\infty} f(x'-x)\, f(x')\, dx' = \int_{-\infty}^{\infty} f(x')\, f(x+x')\, dx'$$

In 2-D:

$$f(x,y) \star\star f(x,y) = \iint_{-\infty}^{\infty} f(x',y')\, f(x+x', y+y')\, dx'\, dy'$$

$$= \iint_{-\infty}^{\infty} f(x'-x, y'-y)\, f(x',y')\, dx'\, dy'$$

Once again, in discrete form:

$$c_i = \sum_j f_{j-i} f_j = \sum_{j=-\infty}^{\infty} f_j f_{i+j}$$

and the usual 2-D extension holds.

Quite often we normalize autocorrelation functions so as to have unity value at the origin. This leads to

$$\gamma(x,y) = \frac{\iint_{-\infty}^{\infty} f(x'-x, y'-y)\, f(x',y')\, dx'\, dy'}{\iint_{-\infty}^{\infty} (f(x,y))^2\, dx\, dy}$$

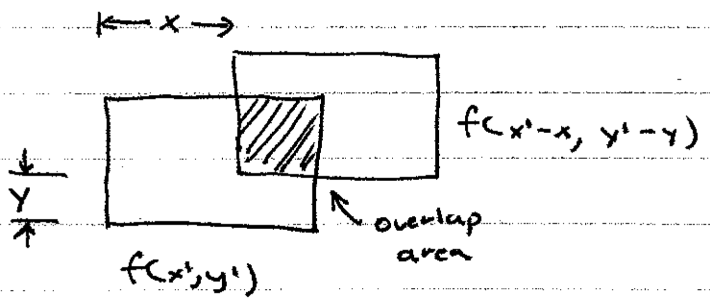
The normalized autocorrelation is simply a scaled version of the unnormalized autocorrelation. The central value is just scaled to be 1.

Picturing Autocorrelation in 2-D

It is often of use to be able to picture the shapes and limits of autocorrelation functions of 2-D data. Recalling that we are starting with $f(x,y)$ and using the relation

$$\iint_{-\infty}^{\infty} f(x'-x, y'-y)\, f(x',y')\, dx'\, dy'$$

we integrate over the entire x'-y' plane to retain a function ~~of~~ $f(x,y)$ only. The two functions in the integral are identical except for a displacement of x and y in the two dimensions.



The product of the two functions is evaluated over the area of overlap only, and integrated over that region to give the value of the autocorrelation for that displacement. While it can be tedious to evaluate the products, it is computationally straight-forward. The product and integral ~~must~~ must be determined for every position on the x-y plane.

However, some insight into the process may be gained by picturing the limits for which the integration is determined. The range of x, y for which non-zero products are found defines the <u>autocorrelation island</u> for the function. This shape defines spectral sensitivity ~~bad~~ bands of a system, for example, and pops up in numerous other situations. Thus it is useful to consider it.

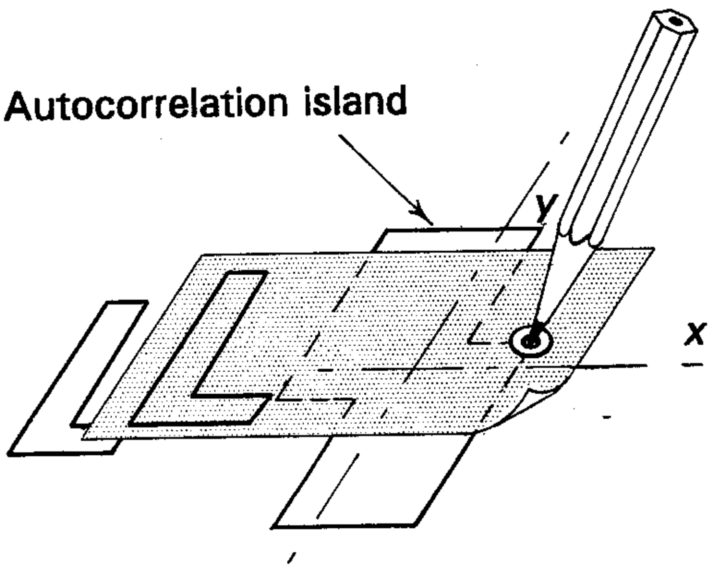


Figure 5-9 Method of tracing shoreline of autocorrelation island by moving the replica (of an ell) to be tangent to the original while maintaining the orientation.

Several example autocorrelation islands are shown below:

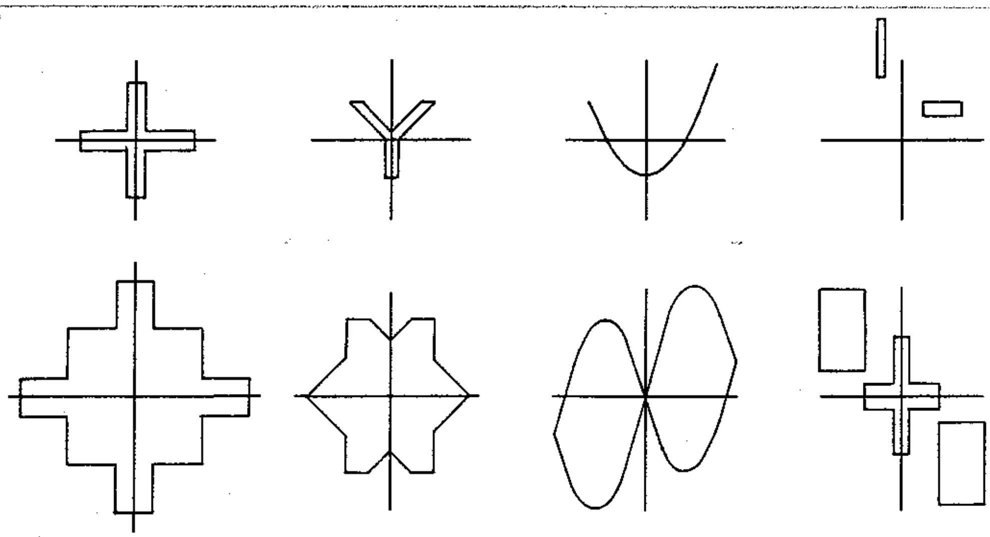


Figure 5-10 Various outlines (above) and their autocorrelation islands (below).

Several properties of autocorrelation islands may be deduced from the examples. First of all, the shapes are invariant to axis-reversal on both axes, in other words, they are the same as their one-half turn versions. A little thought shows the same to be true for the autocorrelation function itself:

Consider $f(x,y)$, and $f \star\star f$

$$f \star\star f = \iint_{-\infty}^{\infty} f(x'-x, y'-y)\, f(x', y')\, dx'dy'$$

Letting $x = -x$, $y = -y$:

$$= \iint_{-\infty}^{\infty} f(x'+x, y'+y)\, f(x', y')\, dx'dy'$$

which is the other formal definition of $f \star\star f$ since multiplication commutes. The two definitions can be shown equal by substitution of variables.

Another feature is that the autocorrelation has a "center" at the origin independent of the location of $f(x,y)$ in the plane. Consider a displaced version of $f(x,y)$:

$$\iint_{-\infty}^{\infty} f(x'-a-x, y'-b-y)\, f(x'-a, y'-b)\, dx'dy'$$

letting $x'' = x'-a$, $y'' = y'-b$, then $dx'' = dx'$, $dy'' = dy'$

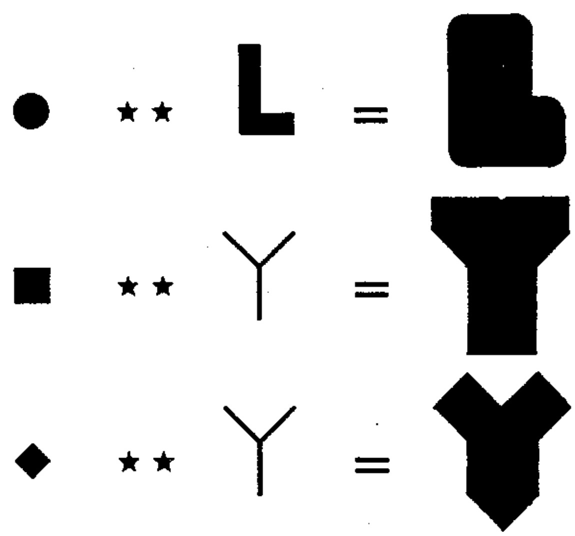
$$= \iint f(x''-x, y''-y)\, f(x'', y'')\, dx''dy''$$

proving invariance of the autocorrelation to a shift.

An application of this property might be a machine to verify check signatures. Using the autocorrelation function instead of the original signature gives robustness to location of the signature.

Cross-correlation Islands - Dilation

Thinking about shapes can be applied to the ideas of cross-correlation islands also. Dilation, or "smearing" of a shape by another is related to calculation of cross-correlation islands. Some simple examples are:



$$f(,) \star\star g(,) = h(,)$$

Figure 5-12 Cross-correlating the function $f(\ ,\ )$ on $g(\ ,\ )$ produces the outline that results from dilating the shape g by the shape f.

These graphical techniques may be of use in graphics arts situations, and also help understand how images may be degraded by the system impulse response functions.

We note that it is also possible to dilate continuous objects with discrete ones, for the purpose of generating multiple offset copies:

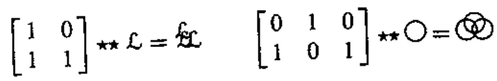


Figure 5-13 Two continuous shapes £ and ○ which have been dilated by a digital object represented by a matrix.

Examples of autocorrelation functions

Lazy pyramid —

Consider a unit rectangle defined by

$$f(x,y) = \begin{cases} 1 & |x| < \frac{1}{2} \text{ and } |y| < \frac{1}{2} \\ 0 & \text{else} \end{cases}$$

$$= {}^2\mathrm{rect}(x,y) = \mathrm{rect}\,x\;\mathrm{rect}\,y$$

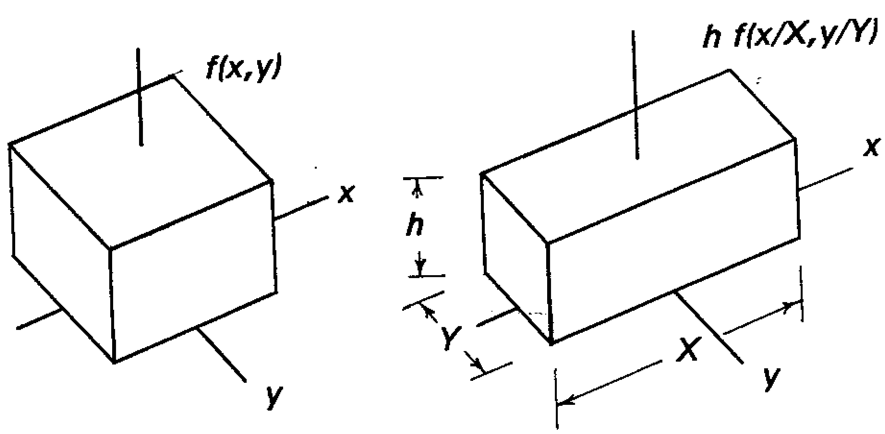
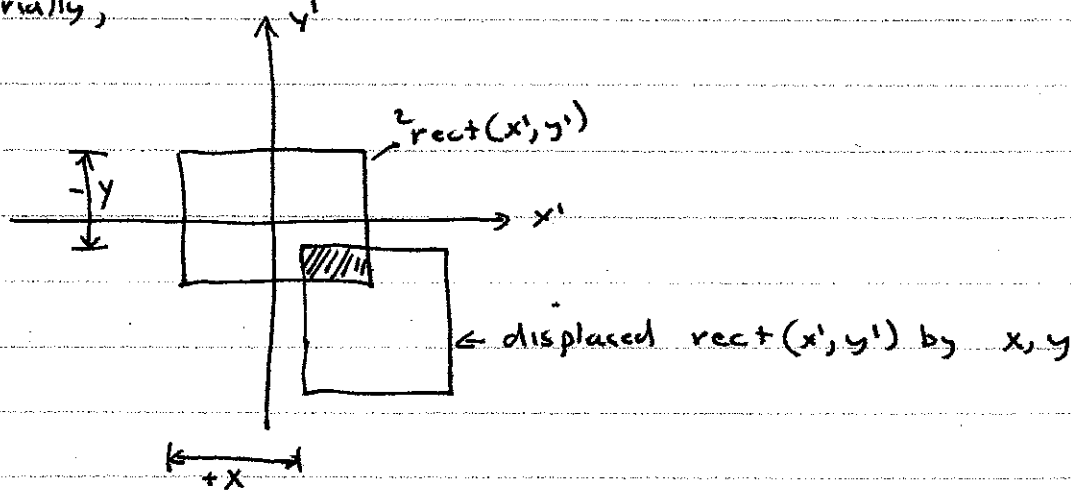


Figure 5-14 The two-dimensional unit rectangle function (left) and an example of how the rectangle function notation may be employed to describe a block of width X, depth Y, and height h (right).

The autocorrelation $c(x,y)$ is

$$c(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} {}^{2}\mathrm{rect}(x'-x, y'-y)\, {}^{2}\mathrm{rect}(x',y')\, dx'\,dy'$$

Pictorially,



The cross-hatch is the area of overlap. Now, if x and y are both positive (a special case)

$$c(x,y) = \int_{x-\frac{1}{2}}^{1/2}\int_{y-\frac{1}{2}}^{1/2} dx'\,dy' \qquad \text{for } 0 \le x < 1 \quad 0 \le y < 1$$

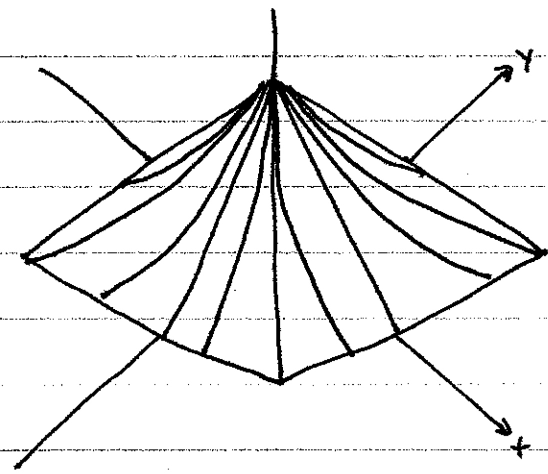
$$= \begin{cases} (1-x)(1-y) & \text{for } 0 \le x < 1 \quad 0 \le y < 1 \\ 0 & \text{else} \end{cases}$$

For the other quadrants, say x is negative. Then the x-limits will be $\int_{-0.5}^{x+0.5}$

The same holds for y, and eventually we will figure out that

$$c(x,y) = (1-|x|)(1-|y|)\, {}^{2}\mathrm{rect}\left(\frac{x}{2}, \frac{y}{2}\right)$$

This function is called the lazy pyramid, because it looks like a pyramid whose sides have slumped down. The ~~sides~~ lines down the direction of steepest descent, down the middle of the sides, are straight.



Thus we say that the lazy ~~pyp~~ pyramid is the autocorrelation function of the unit square-table.

Round-table, or pillbox -

Consider now the round table function, or pillbox:

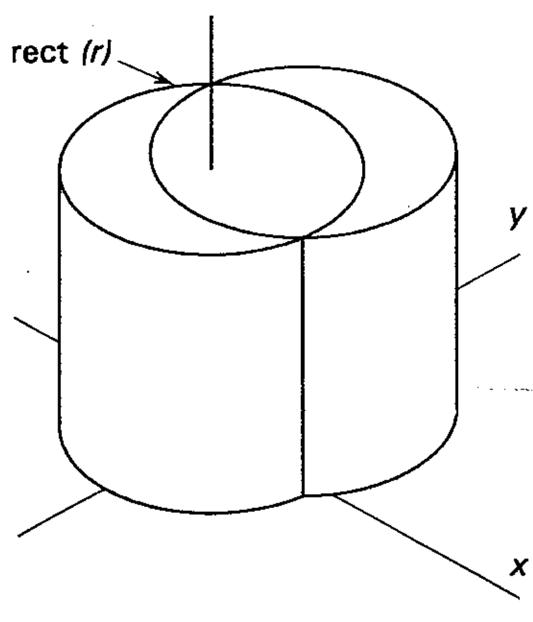
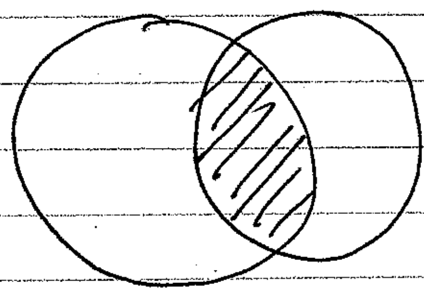


Figure 5-17 The autocorrelation of rect r, the unit round-table function or unit circular pillbox, is determined by integrating the product of rect r with a shifted replica, over the region of overlap.

We can express the pillbox as rect(r). So what is its autocorrelation

$$c(x,y) = \text{rect}(r) \star\star \text{rect}(r)$$

Viewed from above, the region of overlap is shown by



and first we note that it is dependent only on the the separation of the two functions. That is, it possesses circular symmetry which we might expect due to its relationship to the circularly symmetric rect r.

We can evaluate the function with the following construction:

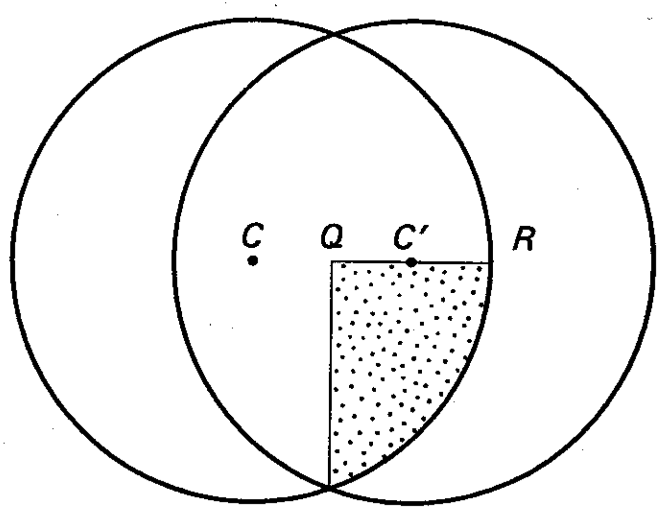


Figure 5-18 The area of overlap of the two unit circles is four times the shaded area PQR. Each circle has radius 0.5, and the separation CC' is r.

Now, the total overlap will be four times the area shaded in the previous picture. That area is

$$\text{area} = 4\left(\text{area of sector } CPR - \text{area of triangle } CPQ\right)$$

$$= 2(CP)^2 \cos^{-1}(CQ/CP) - CC' \sqrt{(CP)^2 - (CQ)^2}$$

$$= \tfrac{1}{2}\cos^{-1} r - \tfrac{1}{2} r\sqrt{1-r^2}$$

Bracewell calls this the chat function.

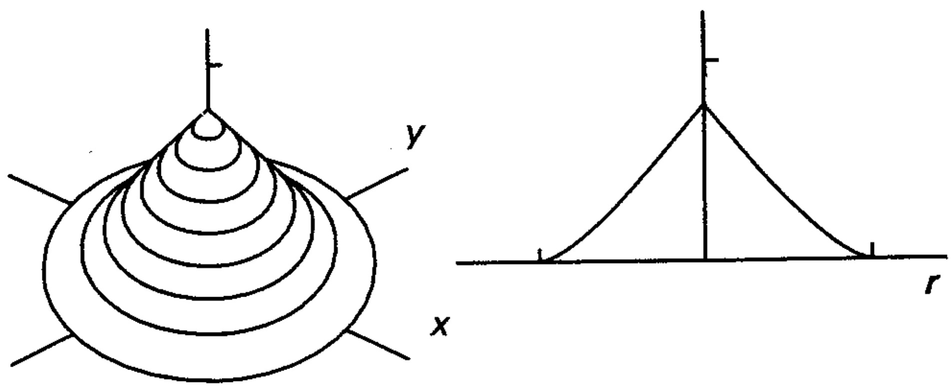


Figure 5-19 A pictorial view of the Chinese hat function (left) and its cross section (right).

Some of its properties: $\text{chat}\ 0 = \frac{\pi}{4}$

$$\int_0^1 2\pi r\,\text{chat}\, r\, dr = \left(\frac{\pi}{4}\right)^2$$

The diameter is 2, since rect r has diameter 1. This function describes the transfer function of a circular lens. Its square aperture analog is the lazy pyramid, describing the transfer function of a uniformly excited square antenna.