THE TOPHAT AND H-DOME TRANSFORMS

The *tophat* transform is a composite operation that uses the morphological opening or closing. The *h-dome* transform uses grayscale reconstruction. Both give local extrema (minima and maxima) in grayscale images that can be used as seeds in a segmentation algorithm. To motivate the use of these image transforms as generators of seeds, we give a quick review of segmentation here.

Binary segmentation most simply is carried out by selecting connected components in the image, using a seed image, a mask image (which is typically the actual image), and filling selected connected components in the mask that have seeds in them. Here, the mask is used to clip the filling process. In a more complicated situation, typified by a binary image of touching coffee beans, you need to split binary connected components. This can be done using the distance function and looking for low saddle points that are most easily cut through. How do you do that? Put the distance function upside-down. Then the seeds, which are the local maxima of the distance function (the points that are farthest from the boundaries) are at the bottom of the inverted function.

The boundaries are at the highest point, and the "low passes" through the saddle are on the boundaries of the catchment basins. Segmentation is achieved by filling the basins to define the *watersheds*, which are all the points that drain into the same basin.

Similarly, grayscale segmentation usually proceeds by finding *markers*, or "seeds," and an image of catchment basins surrounded by walls at the boundaries. The catchment filling "mask" can be computed in various ways; a popular one is to use a properly smoothed morphological gradient, which has large peaks at places where the image intensity is rapidly varying -- a likely position to find a segmentation boundary. Filling then proceeds from the seeds into these basins.

There are various ways to get the seeds. Two popular ones are *tophat* and *h-dome* transforms. The tophat is simpler, and you can envision its operation as follows. Suppose you have a dark grayscale image with some *small bright* regions. To identify those regions, apply the tophat, using a SE that is larger than the extent of the regions. The opening is a *Min* operation that removes those bright regions that are smaller in dimension than the SE used in the opening. Then, subtracting this image with the thin peaks cut off from the original image gives you just those peaks, plus some low amplitude noise. The tophat is typically followed by a thresholding operation on the peaks. We've just described the *white* tophat.

There is a *black* tophat that is a dual to the white tophat, and it subtracts the original image from the closing with a SE. The black tophat gives large pixel values in the result where there were *small dark* regions in the original image.

The *h-domes* are another method for finding local maxima. They use a grayscale reconstruction (seed filling) method, where the original image is the mask and the seed is derived from the mask by subtracting a constant value "h" from each pixel value. Reconstruction expands the seed into the original image (mask). This is visualized as having each local maximum of the seed expand horizontally until it hits a value of the mask that is of equal or greater value, at which point it can expand no further. To complete the h-dome transform, the filled seed is then subtracted from the original image, resulting in an image composed of the "domes" of local maxima of the original, none of which can exceed the value h in size.

You can find implementations of the tophat transform pixTophat() and the hdome transform pixHDome() in morphapp.c. We have implemented both black and white tophat transforms, even though they are trivially related. Intuitively, applying a black tophat to find small dark regions should be equivalent to applying a white tophat to the inverse of the image. (The inverse of an image is found by replacing each pixel by its reflection in the light-dark axis; specifically, if the pixel has value *v*, it is replaced by the value *255 - v*.) And, in fact, this is correct.

These operations have exactly that simple duality relation: *the white tophat on an image is equal to the black tophat on the inverse of the image, and v.v.* The proof, which uses the duality of the opening and closing operations, is very simple. Let the opening of an image I with a structuring element S be given by $O_S(I)$, and the closing of I by S be $C_S(I)$. From here on, due to a typographic limitation of HTML, we'll suppress the "S". Denote the inverse of an image I byI, where I = 255 - I. The opening and closing are dual, in the sense that O(I) = C(I). (Note that this is a different sense of "dual" than for the black and white tophats, which we are in the process of showing.) Denote the white tophat by Tw(I) = I - O(I) and the black tophat by Tb(I) = C(I) - I. Then the black tophat on the inverse of I is given by

$$Tb(I) = C(I) - I = O(I) - I = 255 - O(I) - 255 + I = I - O(I) = Tw(I)$$

which is the result that was to be proven. By the way, the duality relations for opening and closing, O(I) = C(I), or C(I) = O(I), can be stated in words in several ways. Here's one for the second of the pair: *The closing of an image can be equivalently found by inverting the opening of the inverted image.* This holds for binary as well as grayscale, of course.

High frequency noise, which is not well filtered by the white tophat, can be greatly reduced by doing a closing first.

Likewise, an opening is often necessary before a black tophat. The size of the SE in the opening or closing can be comparable to that used in the tophat.

(Parenthetical note. For seeing the results clearly, we provide a function pixMaxDynamicRange() that expands the dynamic range to cover the full set of 256 values for an 8 bpp image. We let you do this with either a linear or log transform. The latter is useful for images that cover a large dynamic range to begin with and have small values that you wish to see and which appear black on a linear scale.)

Source: http://www.leptonica.com/grayscale-morphology.html