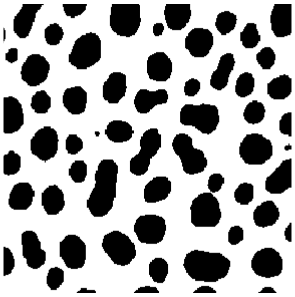
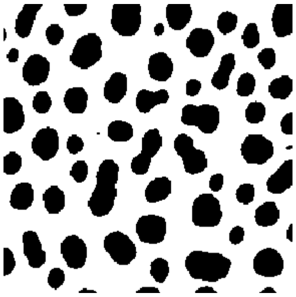
```
NewImage/S=0 fuzzyM; DoWindow/T kwTopWin, "Fuzzy Means Thresholding"
```



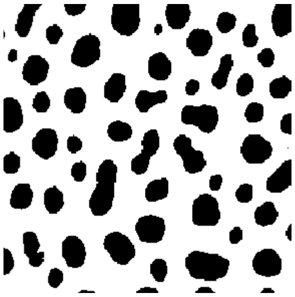
Fuzzy Entropy



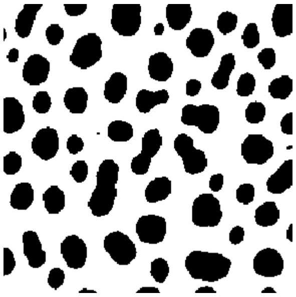
Fuzzy Means

```
// Arifin and Asano method
ImageThreshold/Q/M=6 root:images:blobs
Duplicate/O M_ImageThresh A_and_A
NewImage/S=0 A_and_A; DoWindow/T kwTopWin, "Arifin and Asano Thresholding"

// Otsu method
ImageThreshold/Q/M=7 root:images:blobs
Duplicate/O M_ImageThresh otsu
NewImage/S=0 otsu ; DoWindow/T kwTopWin, "Otsu Thresholding"
```



Arifin and Asano



Otsu

In the these examples, you can add the /C flag to each ImageThreshold command and remove the /Q flag to get some feedback about the quality of the threshold; the correlation coefficient will be printed to the history.

It is easy to determine visually that, for this data, the adaptive and the bimodal algorithms performed rather poorly.

You can improve the results of the adaptive algorithm by running the adaptive threshold also on the transpose of the image, so that the operation becomes column based, and then combining the two outputs with a binary AND.

Spatial Transforms

Spatial transforms describe a class of operations that change the position of the data within the wave. These include the operations **ImageTransform** with multiple keywords, **MatrixTranspose**, **ImageRotate**, **ImageInterpolate**, and **ImageRegistration**.

Rotating Images

You can rotate images using the **ImageRotate** operation (see page V-405). There are two issues that are worth noting in connection with image rotations where the rotation angle is not a multiple of 90 degrees. First, the image size is always increased to accommodate all source pixels in the rotated image (no clipping is done). The second issue is that rotated pixels are calculated using bilinear interpolation so the result of N