ImageSeedFill

```
ImageSeedFill [flags] [keyword], seedX=xLoc, seedY=yLoc, target=setValue,
    srcWave=srcImage
```

The ImageSeedFill operation takes a seed pixel and fills a contiguous region with the target value, storing the result in the wave M_SeedFill. The filled region is defined by all contiguous pixels that include the seed pixel and whose pixel values lie between the specified minimum and maximum values (inclusive). ImageSeedFill works on 2D and 3D waves.

Parameters

keyword is one of the following names:

adaptive=*factor* — Invokes the adaptive algorithm where a pixel or voxel is accepted if its value is between the specified minimum and maximum or its value satisfies:

$$|val - avg| < factor * stdv.$$

Here *val* is the value of the pixel or voxel in question, *avg* is the average value of the pixels or voxels in the neighborhood and *stdv* is the standard deviation of these values. By choosing a small *factor* you can constrain the acceptable values to be very close to the neighborhood average. A large *factor* allows for more deviation assuming that the *stdv* is greater than zero.

This requirement means that a connected pixel has to be between the specified minimum and maximum value **and** satisfy the adaptive relationship. In most situations it is best to set wide limits on the minimum and maximum values and allow the adaptive parameter to control the local connectivity.

fillNumber=*num* — Specifies the number, in the range 1 to 26, of voxels in each 3x3x3 cube that belong to the set. If fillNumber is exceeded, the operation fills the remaining members of the cube. If you do not specify this keyword, the operation does not fill the cube. Used only in the fuzzy algorithm.

fuzzyCenter=*fcVal* — Specifies the center value for the fuzzy probability with the fuzzy algorithm (see **Details**). The default value is 0.25. Its standard range is 0 to 0.5, although interesting results might be obtained outside this range.

fuzzyProb=*fpVal* — Specifies a probability threshold that must be met by a voxel to be accepted to the seeded set. The value must be in the range 0 to 1. The default value is 0.75.

fuzzyScale=*fsVal* — Determines if a voxel is to be considered in a second stage using fuzzy probability. *fsVal* must be nonzero in order to invoke the fuzzy algorithm. The scale is used in comparing the value of the voxel to the value of the seed voxel. The scale should normally be in the range 0.5 to 2.0.

fuzzyWidth=*fwVal* — Defines the width of the fuzzy probability distribution with the fuzzy algorithm (see **Details**). In most situations you should not need to specify this parameter. The default value is 1.

min=*minval* — Specifies the minimum value that is accepted in the seeded set. Not needed when using fuzzy algorithm.

max=*maxval* — Specifies the maximum value that is accepted to the seeded set. Not needed when using the fuzzy algorithm.

seedP=*row* — Specifies the integer row location of the seed pixel or voxel. This avoids roundoff issues when srcWave has wave scaling. You must provide either seedP or seedX with all algorithms. It is sometimes convenient to use this with cursors e.g., seedP=pcsr(a).

seedQ=*col* — Specifies the integer column location of the seed pixel or voxel. This avoids roundoff difficulties when srcWave has wave scaling. You must provide either seedQ or seedY with all algorithms.

seedR=layer	Specifies the integer layer position of the seed voxel. When srcWave is a 3D wave you must use either seedR or seedZ.
seedX=*xLoc*	Specifies the pixel or voxel index. If srcWave has wave scaling, seedX must be expressed in terms of the scaled coordinate. This keyword or seedP is required with all algorithms.
seedY=*yLoc*	Specifies the pixel or voxel index. If srcWave has wave scaling, seedY must be expressed in terms of the scaled coordinate. This keyword or seedQ is required with all algorithms.
seedZ=*zLoc*	Specifies the voxel index. If srcWave has wave scaling, seedZ must be expressed in terms of the scaled coordinate. You must use this keyword or seedR whenever srcWave is 3D.
srcWave=*srcImage*	Specifies the source image wave.
target=*val*	Sets the value assigned to pixels or voxels that belonging to the seeded set.

Flags

/B=*bValue*	Specifies the value assigned to pixels or voxels that do not belong to the filled area. If you omit /B, these pixels or voxels are assigned the corresponding values of the wave specified by the srcWave keyword.
/C	Uses 8-connectivity where a pixel can be connected to any one of its neighbors and with which it shares as little as a single boundary point. The default setting is 4-connectivity where pixels can be connected if they are neighbors along a row or a column. This has no effect in 3D, where 26-connectivity is the only option.
/K=*killCount*	Terminates the fill operation after *killCount* elements have been accepted.
/O	Overwrites the source wave with the output (2D only).
/R=*roiWave*	Specifies a region of interest (ROI). The ROI is defined by a wave of type unsigned byte (/b/u), that has the same number of rows and columns and layers as the image wave. The ROI itself is defined by the entries/pixels whose value are 0. Pixels outside the ROI can take any nonzero value. The ROI does not have to be contiguous. See ImageGenerateROIMask for more information on creating ROI waves.

Details

In two dimensions, the operation takes a seed pixel, optional minimum and maximum pixel values and optional adaptive coefficient. It then fills a contiguous region (in a copy of the source image) with the target value. There are two algorithms for 2D seed fill. In direct seed fill (only min, max, seedX and seedY are specified) the filled region is defined by all contiguous pixels that include the seed pixel and whose pixel values lie between the specified minimum and maximum values (inclusive). In adaptive fill, there is an additional condition for the pixel or voxel to be selected, which requires that the pixel value must be within the standard deviation of the average in the 3x3 (2D) or 3x3x3 (3D) nearest neighbors. If you do not specify the minimum and maximum values then the operation selects only values identical to that of the seed pixel.

In 3D, there are three available algorithms. The direct seed fill algorithm uses the limits specified by the user to fill the seeded domain. In adaptive seed fill the algorithm requires the limits as well as the adaptive parameter. It fills the domain by accepting only voxels that lie within the adaptive factor times the standard deviation of the immediate voxel neighborhood. To invoke the third algorithm you must set fuzzyScale to a nonzero value. The fuzzy seed fill uses two steps to determine if a voxel should be in the filled domain. In the first step the value of the voxel is compared to the seed value using the fuzzy scale. If accepted, it passes to the second stage where a fuzzy probability is calculated based on the number of voxels in the 3x3x3 cell which passed the first step together with the user-specified probability center (fuzzyCenter) and width (fuzzyWidth). If the result is greater than fuzzyProb, the voxel is set to belong to the filled domain.

If the /O flag is not specified, the result is stored in the wave M_SeedFill.

If you specify a background value with the /B flag, the resulting image consists of the background value and the target value in the area corresponding to the seed fill. Although the wave is now bi-level, it retains the same number type as the source image.