```
    // Display
    NewImage M_InterpolatedImage
    NewImage oneD
End
```

See Also

The **interp**, **Interp3DPath**, **ImageRegistration**, and **Loess** operations. The **ContourZ** function. For examples see **Interpolation and Sampling** on page III-359.

References

Unser, M., A. Aldroubi, and M. Eden, B-Spline Signal Processing: Part I-Theory, *IEEE Transactions on Signal Processing*, *41*, 821-832, 1993.

Douglas B. Smythe, "A Two-Pass Mesh Warping Algorithm for Object Transformation and Image Interpolation" ILM Technical Memo #1030, Computer Graphics Department, Lucasfilm Ltd. 1990.

ImageLineProfile

ImageLineProfile [***flags***] **xWave=*xwave*, yWave=*ywave*, srcWave=*srcWave*** [**, width=*value*, widthWave=*wWave***]

The ImageLineProfile operation provides sampling of a source image along an arbitrary path specified by the two waves: *xWave* and *yWave*. The arbitrary path is made of line segments between every two consecutive vertices of *xWave* and *yWave*. In each segment the profile is calculated at a number of points (profile points) equivalent to the sampling density of the original image (unless the /V flag is used). Both *xWave* and *yWave* should have the same scaling as *srcWave*. If *srcWave* does not have the same scaling in both dimensions you should remove the scaling to compute an accurate profile.

At each profile point, the profile value is calculated by averaging samples along the normal to the profile line segment. The number of samples in the average is determined by the keyword width. The operation actually averages the interpolated values at N equidistant points on the normal to profile line segment, with `N=2(width+0.5)`. Samples outside the domain of the source image do not contribute to the profile value.

The profile values are stored in the wave W_ImageLineProfile. The actual locations of the profile points are stored in the waves W_LineProfileX and W_LineProfileY. The scaled distance measured along the path is stored in the wave W_LineProfileDisplacement.

When the averaging width is greater than zero, the operation can also calculate at each profile point the standard deviation of the values sampled for that point (see /S flag). The results are then stored in the wave W_LineProfileStdv. When using this operation on 3D RGB images, the profile values are stored in the 3 column waves M_ImageLineProfile and M_LineProfileStdv respectively.

Parameters

srcWave=*srcWave*	Specifies the image for which the line profile is evaluated. The image may be a 2D wave of any type or a 3D wave or RGB data.
xWave=*xwave*	Specifies the wave containing the x coordinate of the line segments along the path.
yWave=*ywave*	Specifies the wave containing the y coordinate of the line segments along the path.
width=*value*	Specifies the width (diameter) in pixels (need not be an integer value) in a direction perpendicular to the path over which the data is interpolated and averaged for each path point. If you do not specify width or use width=0, only the interpolated value at the path point is used.
widthWave=*wWave*	Specifies the width of the profile (see definition above) on a segment by segment basis. *wWave* should be a 1D wave that has the same number of entries as xWave and yWave. If you provide a widthWave any value assigned with the width keyword is ignored. All values in the wave must be positive and finite.