Image Sampling Methods

GRASS Tutorial on `s.sample`

James Darrell McCauley

04 January 1993

Abstract

The sampling methods used by the GRASS program `s.sample` [1] are presented. These are: nearest neighbor sampling, bilinear interpolation, and cubic convolution interpolation.

Interpolation software often writes output as images rather than punctual form. With a defined resolution in the x-y plane, picture elements, known as *pixels* or *cells*, assume a value for a small rectangular region. This value is the result of interpolating at the center of the pixel.

This output representation poses problems for cross-validation. Data for cross validation, like the data used for interpolation, are not gridded but rather scattered. Their locations in the x-y plane will not necessarily coincide with the centers of pixels. Therefore, a z-value at the exact location of the point used for cross-validation may not be obtainable from the output image. However, there are ways of approximating values. Three such methods are nearest neighbor, bilinear interpolation, and cubic convolution interpolation [2]. These are implemented in the GRASS program `s.sample` [1] and discussed below.

Nearest neighbor sampling is probably the simplest approach to approximating values. In this method, the value of the pixel whose center is nearest to the cross-validation point is used. Figure 1 shows this graphically.

Bilinear interpolation is another method. It uses a pixel values (ϕ) of a 2×2 neighborhood of pixels to perform three linear interpolations (see fig. 1). First, two interpolations are performed, one along the row of pixels above the point and one along the row of pixels below the point, yielding the interpolants $\phi\,(i, j')$ and $\phi\,(i+1, j')$ given by:

$$\phi\,(i, j') = j'\phi\,(i, j+1) + (1-j')\,\phi\,(i, j) \quad (1)$$

$$\phi\,(i+1, j') = j'\phi\,(i+1, j+1) + (1-j')\,\phi\,(i+1, j) \quad (2)$$

These equations assume a grid spacing of unity. The position of the cross validation point, (i', j'), is measured relative to (i, j). The final step is to interpolate linearly over the two interpolants $\phi\,(i, j')$ and $\phi\,(i+1, j')$, yielding:

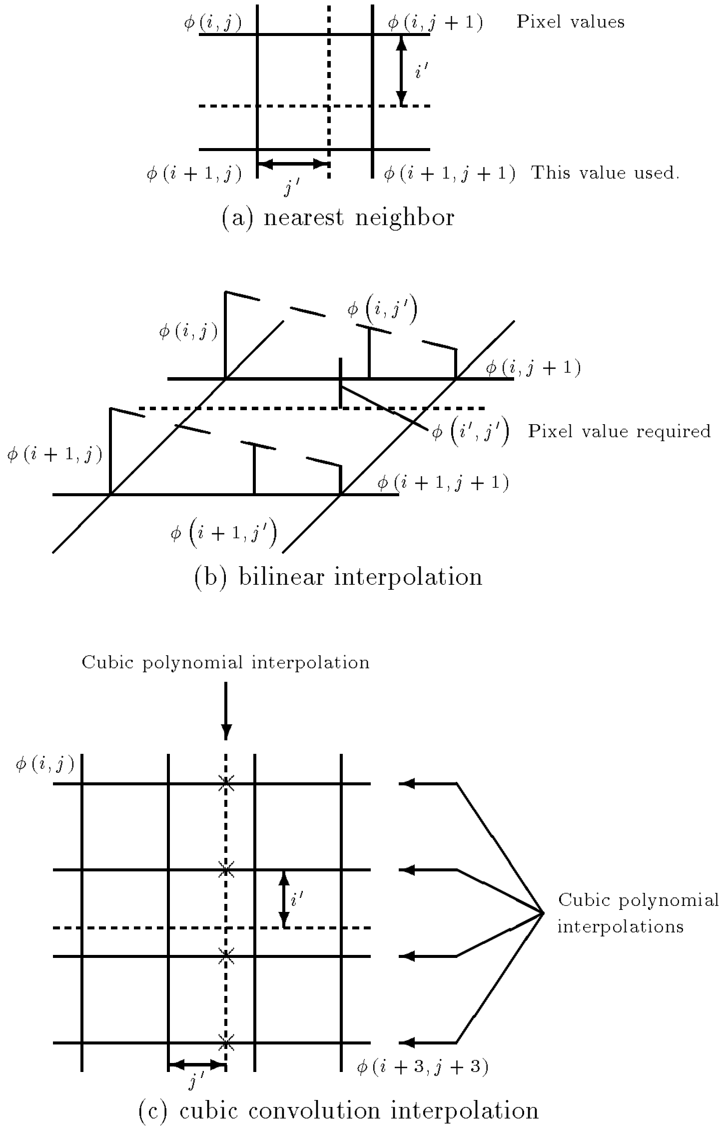

Figure 1: Image sampling using (a) nearest neighbor, (b) bilinear interpolation, and (c) cubic convolution interpolation; pixel centers are at crossings, unknown point is located at (i', j'), grid spacing of unity is assumed. Adapted from Richards [2].

$$
\begin{aligned}
\phi(i', j') &= (1-i')\left[j'\phi(i, j+1) + (1-j')\phi(i, j)\right] \\
&\quad + i'\left[j'\phi(i+1, j+1) + (1-j')\phi(i+1, j)\right]
\end{aligned} \tag{3}
$$

where $\phi(i', j')$ is the estimated value of the cross validation point.

Cubic convolution interpolation is another method of finding a value at (i', j'). It uses a 4×4 neighborhood of pixels. The algorithm that is used to perform this operation follows the following equation:

$$
\begin{aligned}
\phi(i, j') &= j'\{j'(j'[\phi(i, j+3) - \phi(i, j+2) + \phi(i, j+1) - \phi(i, j)] \\
&\quad + [\phi(i, j+2) - \phi(i, j+3) - 2\phi(i, j+1) + 2\phi(i, j)]) \\
&\quad + [\phi(i, j+2) - \phi(i, j)]\} \\
&\quad + \phi(i, j+1)
\end{aligned} \tag{4}
$$

This expression is evaluated for each row of the 4×4 neighborhood shown in figure 1 to yield the four interpolants $\phi(i, j')$, $\phi(i+1, j')$, $\phi(i+2, j')$, $\phi(i+3, j')$. These are interpolated parallel to pixel columns according to:

$$
\begin{aligned}
\phi(i', j') &= i'\{i'(i'[\phi(i+3, j') - \phi(i+2, j') + \phi(i+1, j') - \phi(i, j')] \\
&\quad + [\phi(i+2, j') - \phi(i+3, j') - 2\phi(i+1, j') + 2\phi(i, j')]) \\
&\quad + [\phi(i+2, j') - \phi(i, j')]\} \\
&\quad + \phi(i+1, j')
\end{aligned} \tag{5}
$$

Nearest neighbor sampling is the simplest method and retains original pixel values. This may be desirable if the purpose for sampling is to perform classifications [2]. Bilinear interpolation is very intuitive and requires few operations. Cubic convolution also requires few operations but requires 4 times as many pixels to be stored. The size of the neighborhood considered for sampling becomes important when considering locations near edges of image boundaries.

References

[1] James Darrell McCauley. *s.sample: software to sample raster maps. GRASS 4.2 Reference Manual.* U.S. Army Corps of Engineers, Construction Engineering Research Laboratories, Champaign, Illinois, to appear 1994.

[2] John A. Richards. *Remote Sensing Digital Image Analysis.* Springer-Verlag, Berlin, 2nd edition, 1993.