EEE-6512: Image Processing and Computer Vision

September 20, 2017

Lecture #5: Point and Geometric Transformations

Damon L. Woodard, Ph.D.
Dept. of Electrical and Computer Engineering
dwoodard@ece.ufl.edu

Chapter Outline

- Different ways to transform an image into another image
- Simple Geometric Transformations
- Graylevel Transformations
- Graylevel Histograms
- Multispectral Transformations
- Multi-Image Transformations
- Change Detection
- Compositing
- **Interpolation**
- Warping

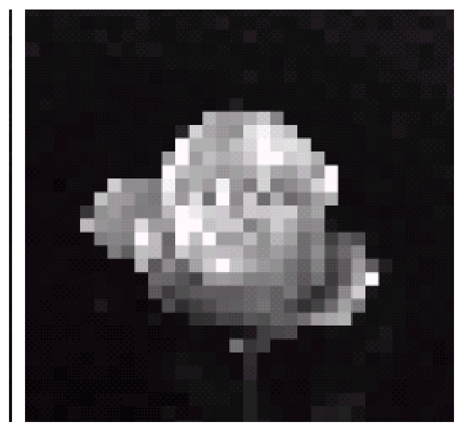
Interpolation

What is interpolation?

- Given a discrete-space image f_d[n,m]
- This corresponds to samples of some continuous space image $f_a\,(\mathrm{x,y})$
- Compute values of continuous space image $f_a\,(\mathrm{x,y})$ at $(\mathrm{x,y})$ locations other than the sample locations

$f_a\,(\mathrm{x,y})$

f_d[n,m]



f'_d[n,m]

Interpolation

To be a true interpolation function, the estimated continuous function must coincide with the sampled data at the sample points.

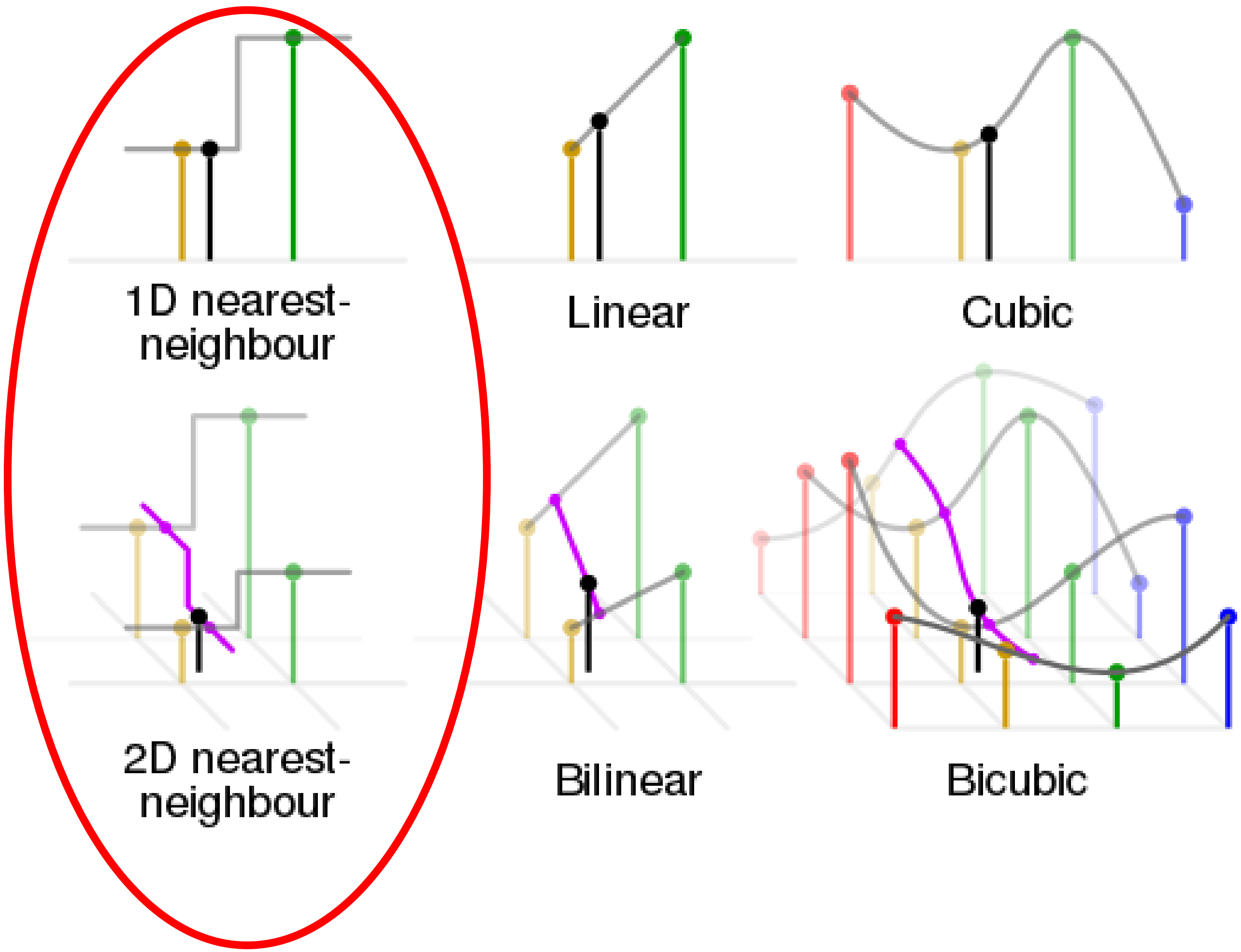
However, it is sometimes desirable to relax this requirement.

Why is interpolation important?

Interpolation is used for:

- Image zooming
- Warping
- Displaying

Subset of Interpolation Techniques



Interpolation - Nearest Neighbor Interpolation

- **Nearest neighbor interpolation:** returns the gray level of the pixel nearest the coordinates:

$$\hat{I}(x, y) \equiv I(\min(\max(\textsc{Round}(x), 0, width - 1)), \min(\max(\textsc{Round}(y), 0, height - 1)))$$

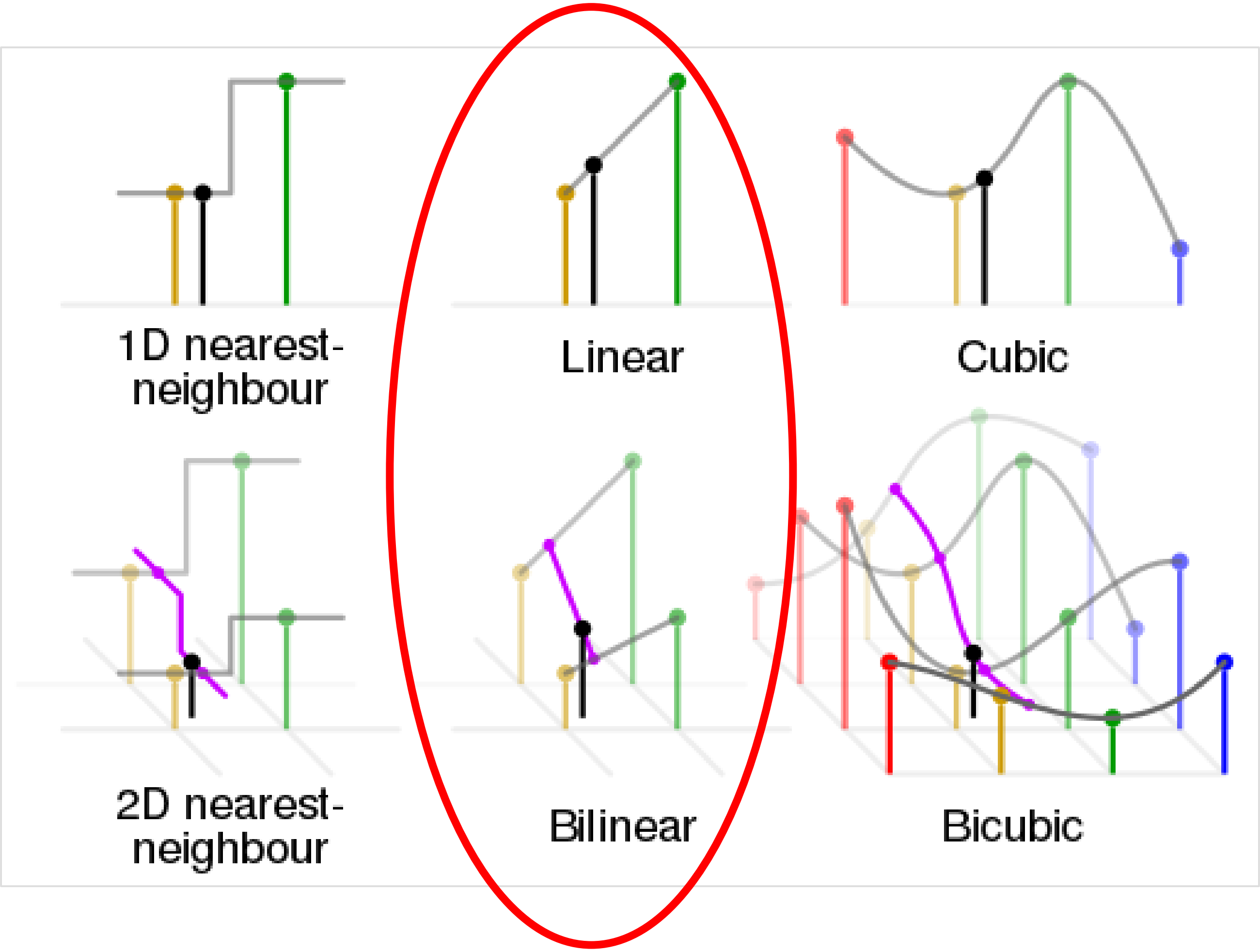
- The notation is simplified considerably if bounds checking can be assumed:

$$\hat{I}(x, y) \equiv I(\textsc{Round}(x), \textsc{Round}(y))$$

Interpolation - Nearest Neighbor Interpolation

- Advantages:
  - Simple to compute
  - Sample values not changed
- Disadvantages:
  - Tends to increase noise and jagged boundaries

Subset of Interpolation Techniques



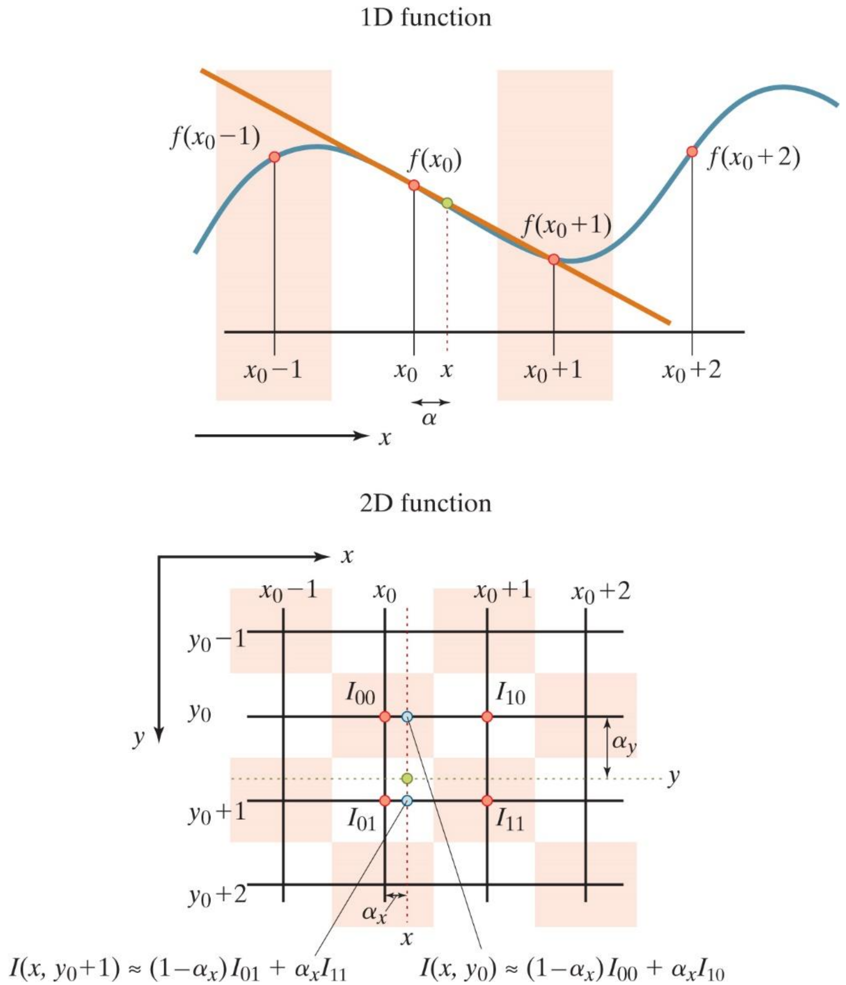
Bilinear Interpolation

- A more accurate approach is **bilinear interpolation**.
  - It is a 2D extension of 1D linear interpolation.
- The interpolated value is the weighted average of the four nearby pixels:

$$\hat{I}(x, y) = \overline{\alpha}_x \overline{\alpha}_y I_{00} + \alpha_x \overline{\alpha}_y I_{10} + \overline{\alpha}_x \alpha_y I_{01} + \alpha_x \alpha_y I_{11}$$

Bilinear Interpolation

Figure 3.33 TOP: Linear interpolation $\hat{f}(x)$ at an arbitrary point x of a discrete function f is computed as the weighted average of the two nearby sampled values, namely $f(x_0)$ and $f(x_0+1)$, where $x_0 = \lfloor x \rfloor$. BOTTOM: Bilinear interpolation $\hat{I}(x,y)$ at a point (x, y) of a discrete image I is computed as the weighted average of the four nearby gray levels, namely I_{00}, I_{10}, I_{01}, and I_{11}. The alternating white and shaded regions indicate the extent of the sampled pixels in the continuous domain.

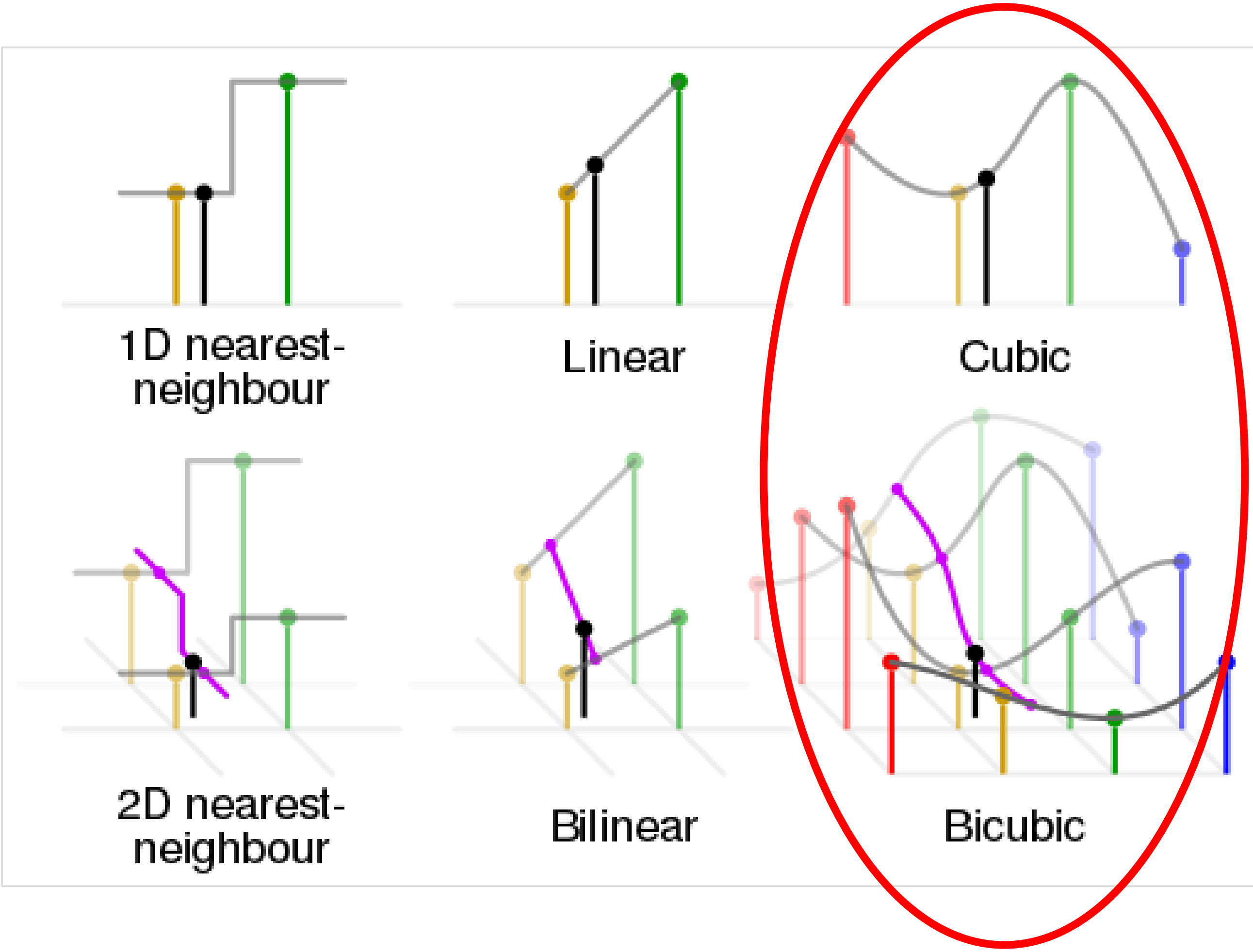


Interpolation – Bilinear Interpolation

- Advantages:
  - Simple to compute
  - Better results than nearest neighbor interpolation

- Disadvantages:
  - More computation needed
  - Loose some of the fine details of the image

Subset of Interpolation Techniques

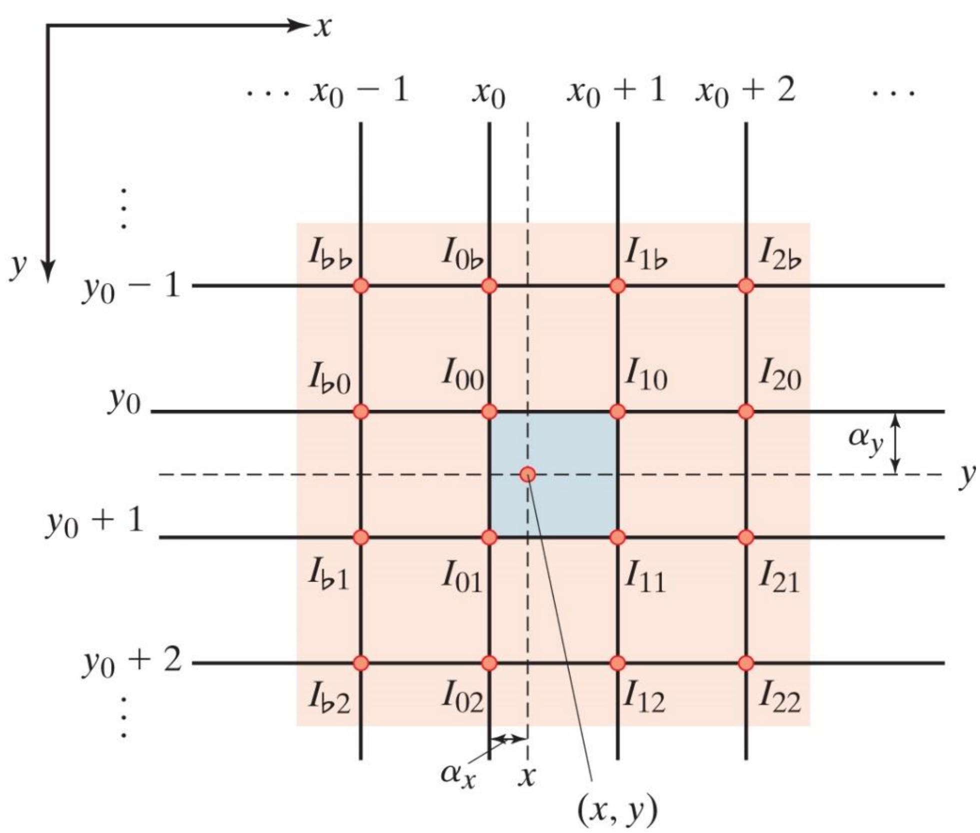


Bicubic Interpolation

- The best performing interpolation techniques of the three is **bicubic interpolation.**
- The interpolated value is the weighted sum of the 16 nearby pixels
- Closer pixels are given a higher weighting in the final calculation.
- This is the standard used in commercial image editing programs.

Interpolation – Bicubic interpolation (cont.)

Figure 3.36 Bicubic interpolation at a point (x, y) is a weighted average of the 16 nearby gray levels.



Interpolation – Bicubic Interpolation

- Advantages:
  - Smoother surface that previous methods
  - Preserves fine image detail
- Disadvantages:
  - Higher complexity
  - More computation needed

Alternate Visualization of linear and bicubic interpolation

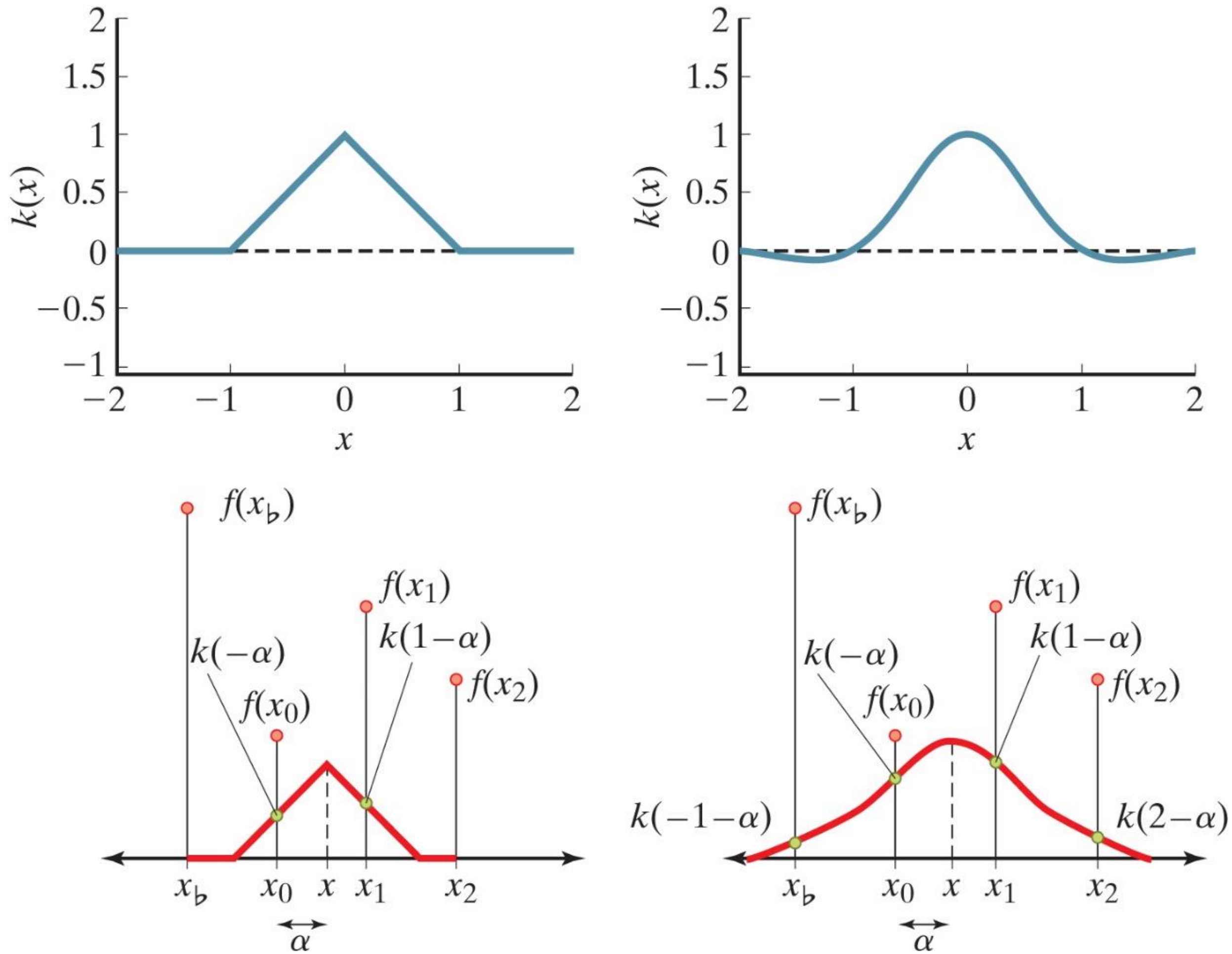


Figure 3.35 TOP: Linear (left) and cubic (right) 1D interpolation kernels. The dashed line indicates $k(x) = 0$ to emphasize that the cubic interpolation kernel contains negative values. BOTTOM: Interpolation involves shifting the kernel so that it is centered at the desired position x, then the neighboring samples are combined using the weights from the kernel.

Keys Filters

- **Cubic convolution filter:** Bicubic interpolation can be improved upon in two ways:
  - First, by reducing its computational expense, and secondly, by relaxing the requirement that the weighting function $k(x)$ be a true interpolation.

$$\hat{f}(x) \equiv \sum_{i=-1}^{2} k(i - \alpha) f(x_0 + i)$$

Keys Filters (cont.)

The kernel $\boldsymbol{k}(\boldsymbol{x})$ is a piecewise cubic spline function specified by two parameters b and c which control the smoothing and spline tension.

The two parameters govern the type of filter, allowing us to generate any smoothly fitting piecewise cubic filter.

A **keys filter** satisfies the constraint:

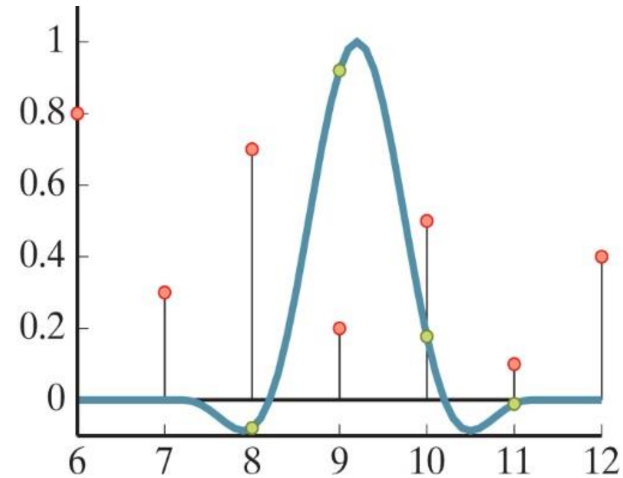
$$b + 2c = 1$$

Lanczos Interpolation

- Another important method is **Lanczos interpolation**, whose interpolation kernel is the well-known **sinc** function multiplied by a truncated sinc function:

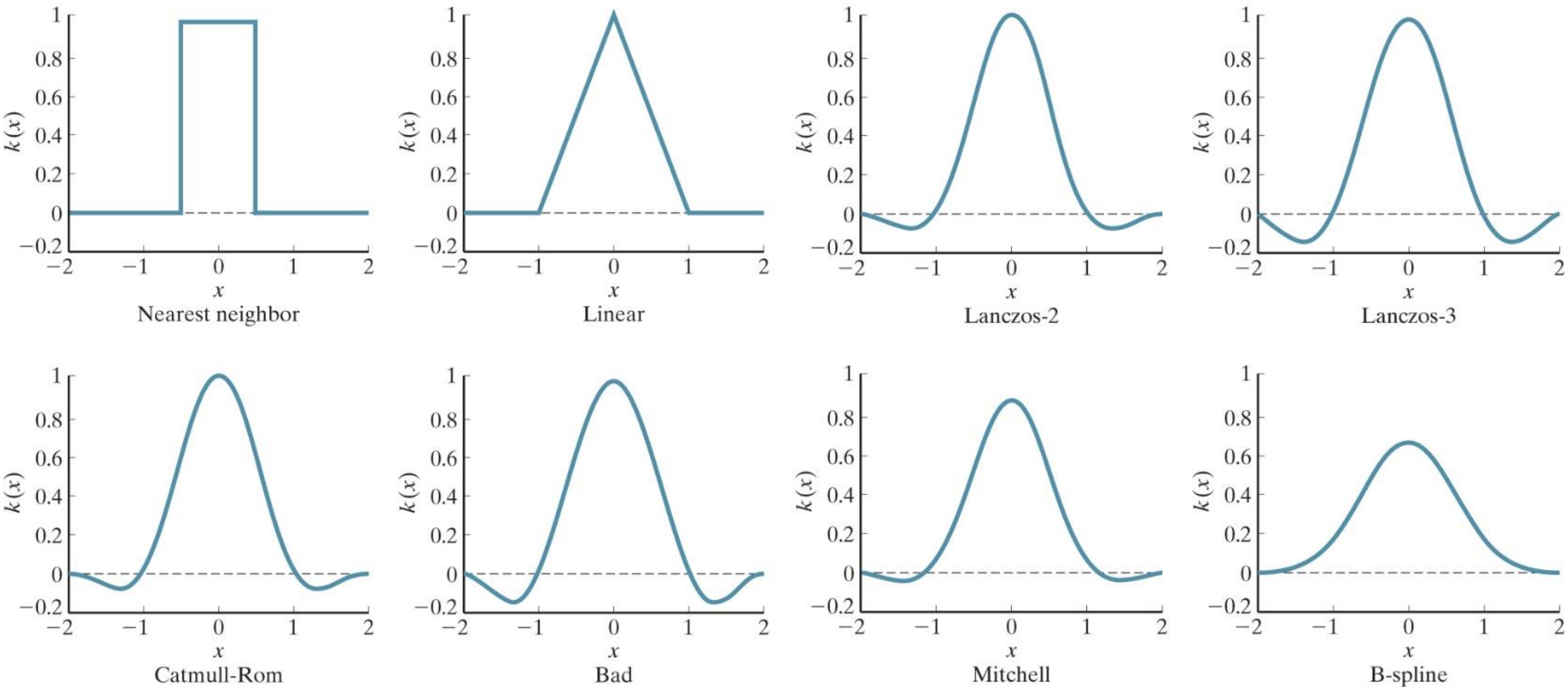
$$k(x) = \begin{cases} (\operatorname{sinc} x)\cdot(\operatorname{sinc} \frac{x}{a}) & \text{if } -a < x < a \\ 0 & \text{otherwise} \end{cases}$$

Figure 3.38 Interpolation of a 1D signal. Here the signal shown by the vertical lollipops is evaluated at $x = 9.2$. The interpolation function is the smooth curve (Lanczos-2 in this case). The 4 green circles indicate the values of the interpolation function that are elementwise multiplied by the corresponding signal values, and then summed to yield the interpolated value.



Various Interpolation Kernels

Figure 3.39 Various interpolation kernels, including those that introduce a small amount of smoothing. The "bad filter" is at $b = 0$, $c = 1$. Note that the last two kernels are not true interpolation functions, because they do not satisfy $k(\pm 1) = k(\pm 2) = 0$.



Comparison of 1D Interpolation Methods

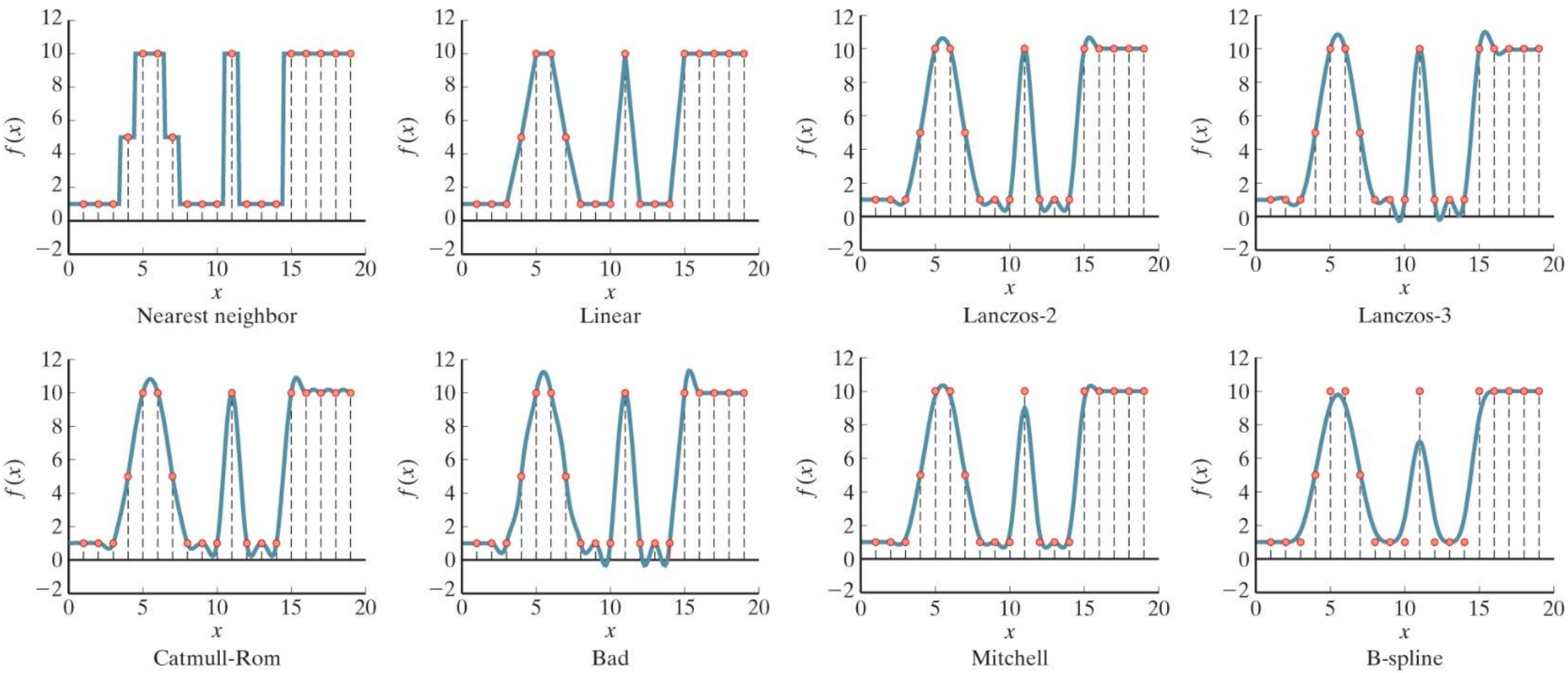


Figure 3.40 Comparison of 1D interpolation methods, some with smoothing, on an example signal. Overall the Catmull-Rom, Mitchell, and Lanczos-2 methods do the best job of providing a smooth fit to the signal without excessive overshoot or ringing.

Questions?