How Good is the Bathymetry?

I've always wondered how well the Akima interpolation method does. The Akima algorithm is used in the scientific community extensively in all kinds of applications. Why is another matter of discussion.

The question here is: How well does it work with the bathymetric data of Deep Creek Lake?

I spent a day working up a test problem and trying out various ways to demonstrate the "goodness of fit."

I defined a grid of 100 x100 ft or miles, or whatever; units do not matter. For that grid, I computed a random set of x,y coordinates and z values. For 'x' and 'y' they were selected from a uniform distribution between 0 and 100, for 'z' they were selected from a normal distribution with the resulting values normalized to 100, equivalent to the maximum depth value used in all of the bathymetric work.

I used the same color palette and breaks in water depth levels that were used to generate the final bathymetry map.

The idea was to produce the contours and to plot the points on top of the map in the color that they should have assumed and visually check to see if they were located in the appropriate places.

In addition to the Akima interpolation procedure, I recently found another method in one of the R packages, labeled as MBA (R has thousands of functions collected in hundreds of packages and, because of the sheer number of them, its easy to miss some very useful ones).

The result of the current analysis is basically four graphs (Figures 1-4):

1. A bar graph showing the colors and range of the color coding as provided by the palette used
2. A scatter plot with just the color coded 'z' points. Note that the values of the points are also shown
3. A graph with the Akima interpolation results and the color coded points super imposed.
4. A graph with the MBA interpolation results similar to the Akima one.

An additional pair of images was run with a 10 fold increase of data points and the results shown in Figure 5.

Appendix A has the listing of the R script that produced these results.

PLV: 8/16/2013

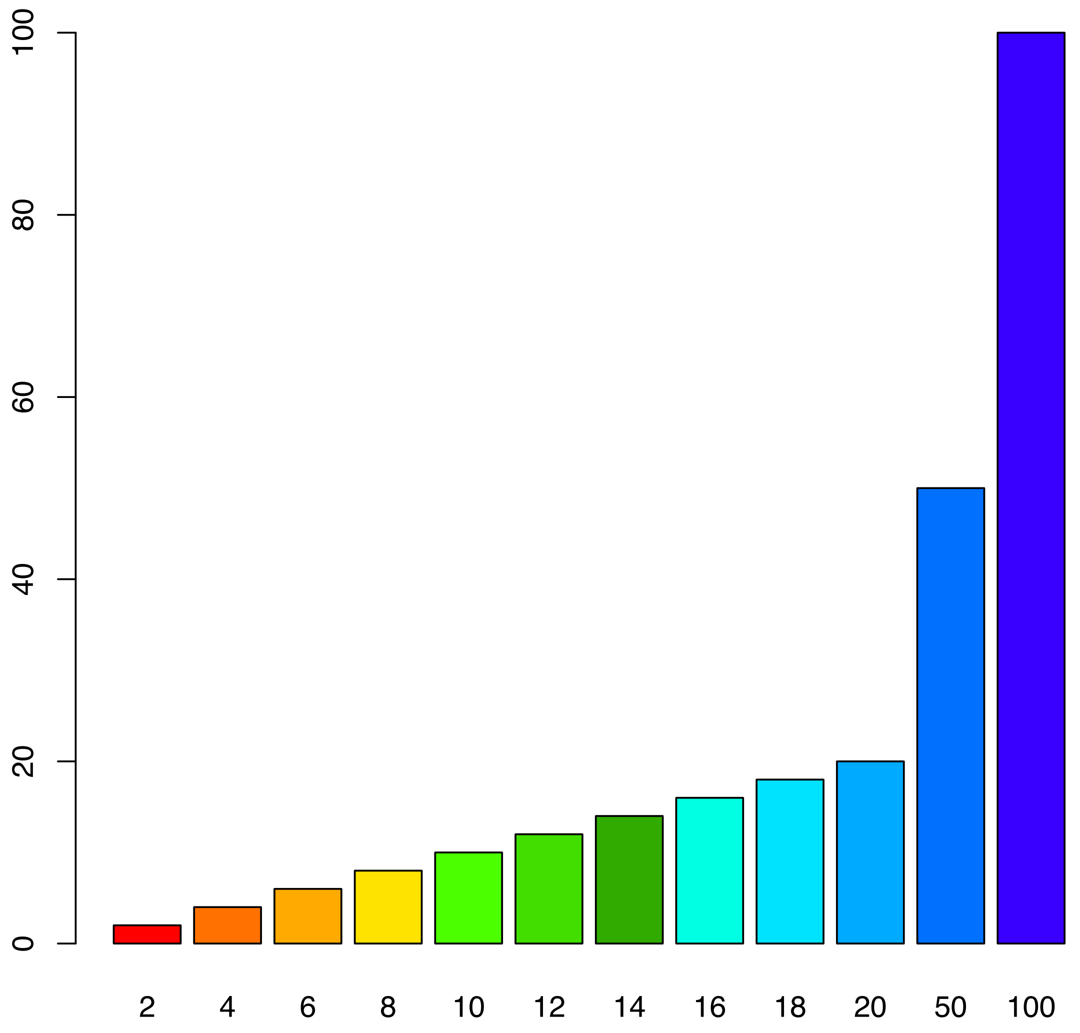


Figure 1

The color palette used. The numbers below the bars signify the end of the bar interval. Hence '2' means it goes from 0 to 2; 50 means it goes from 20 to 50.

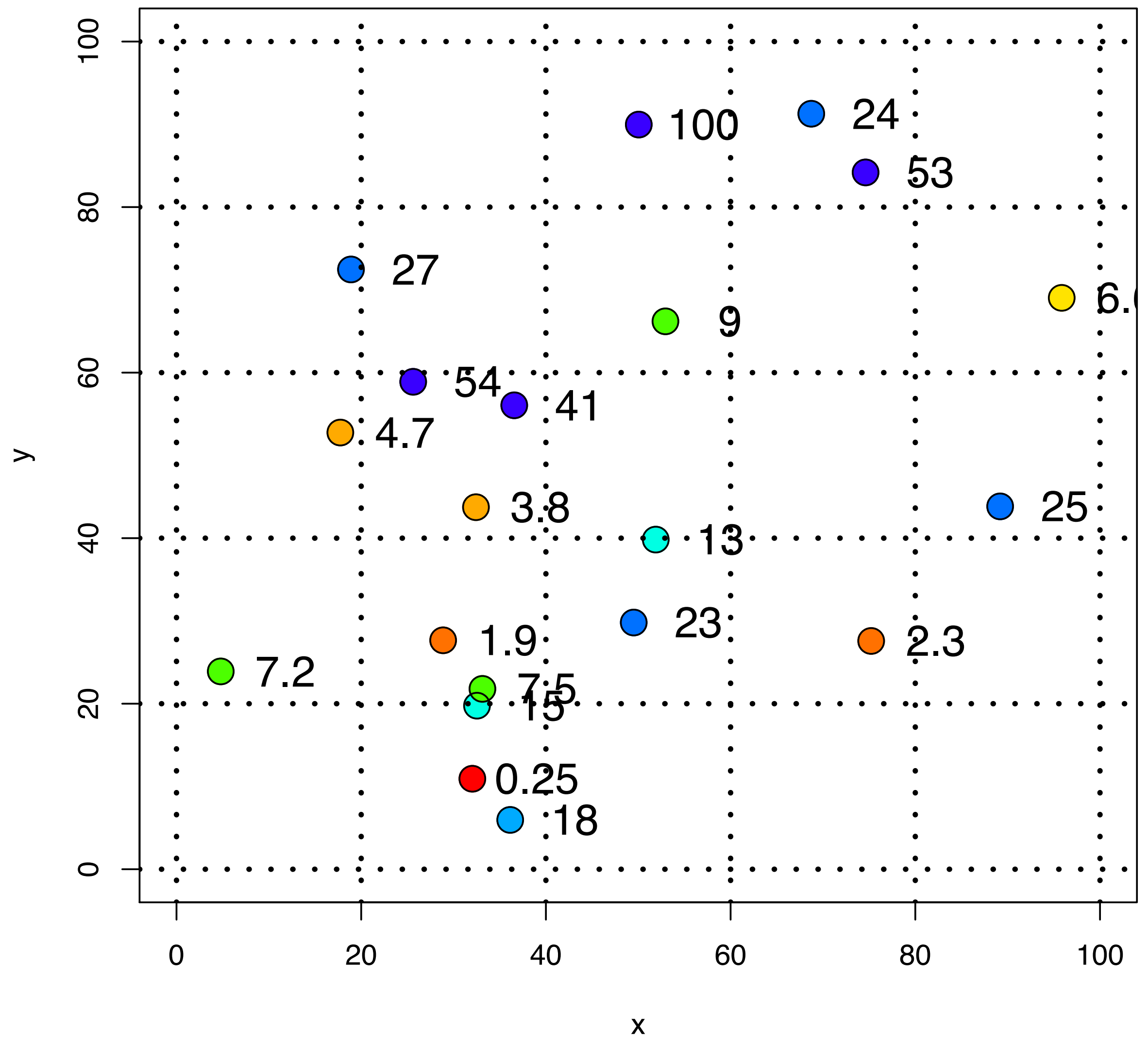


Figure 2

The points are color coded according to the color palette shown in Figure 1. The numbers next to the points are the z-values.

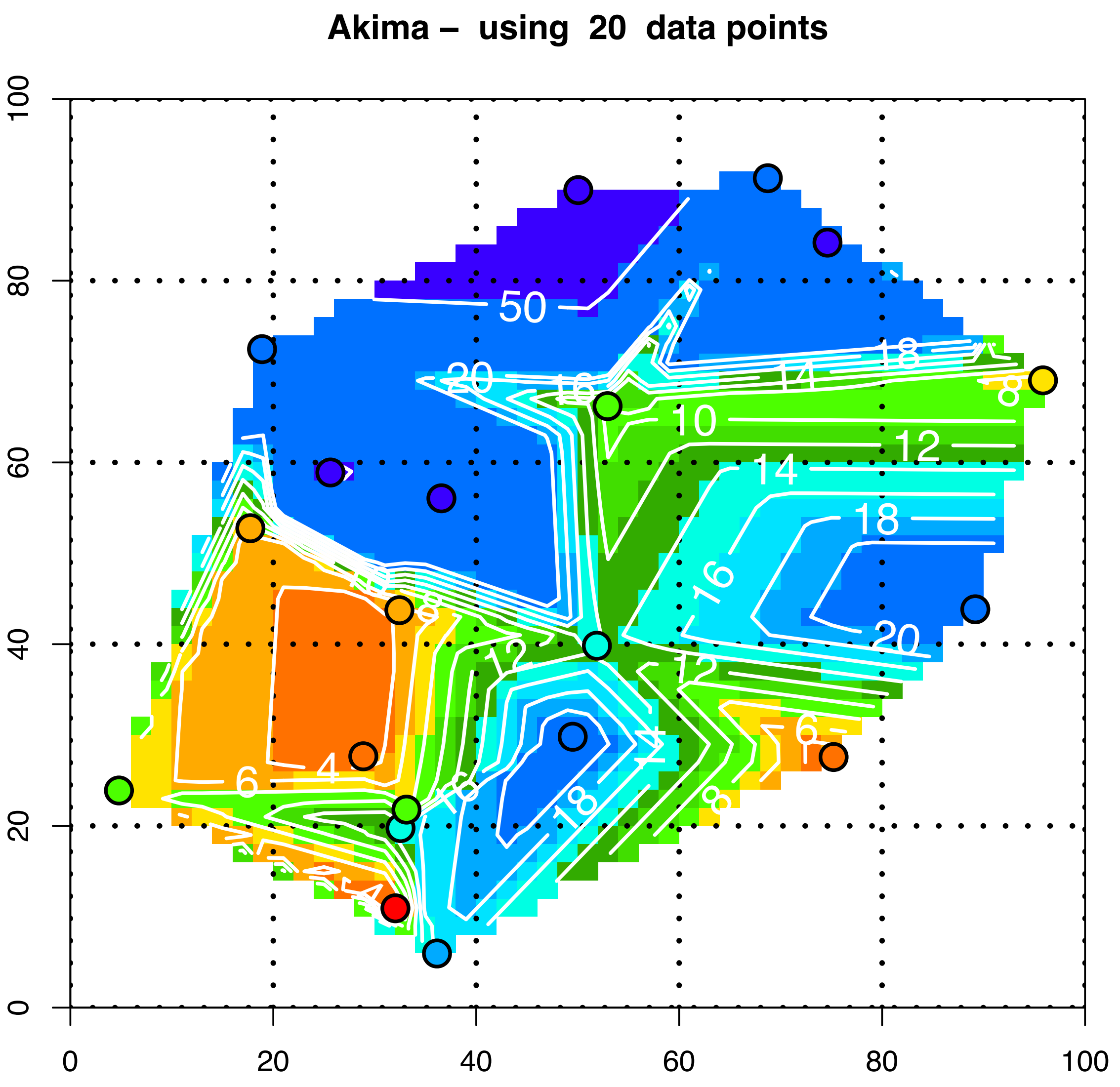


Figure 3

This graph shows the results of the Akima interpolation method. Whether the shapes of the contours are right or wrong can be debated. They do lie, more or less, where one expects them to be. In other words, the points appear to lie in their appropriate locations.

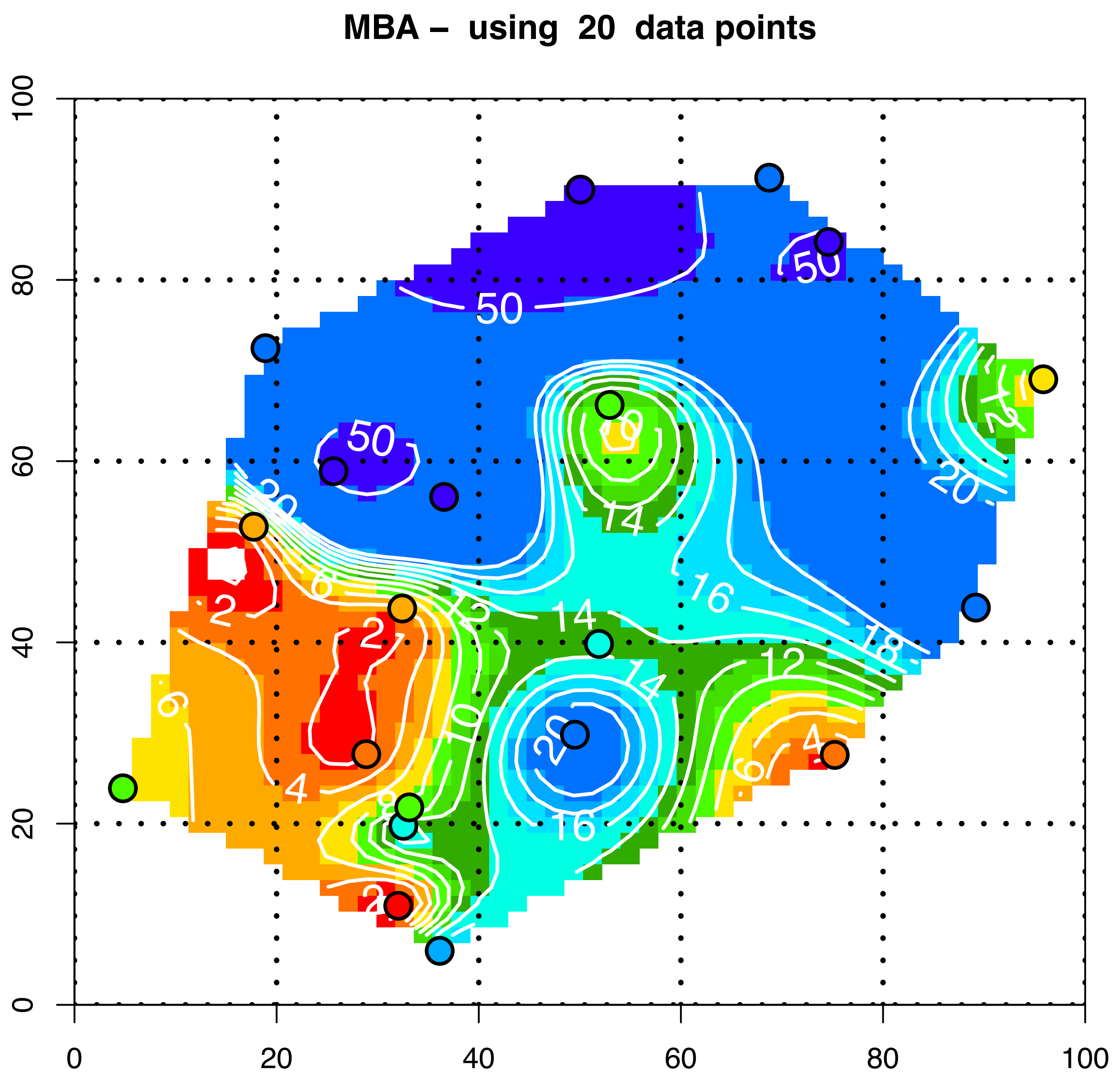
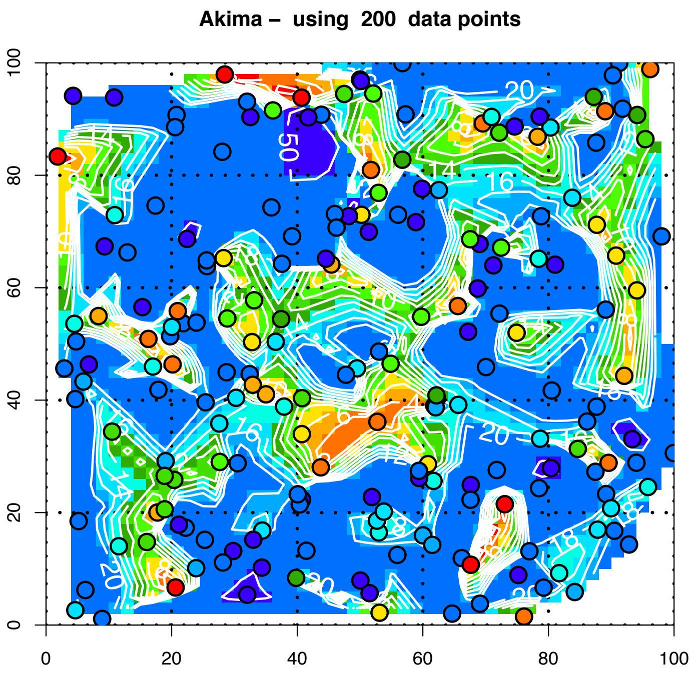


Figure 4

This graph show the results of the MBA interpolation method. The contours are much more pleasing to look at. The points also fall on the correct locations. These results compare reasonably well with the Akima results, but they are different.



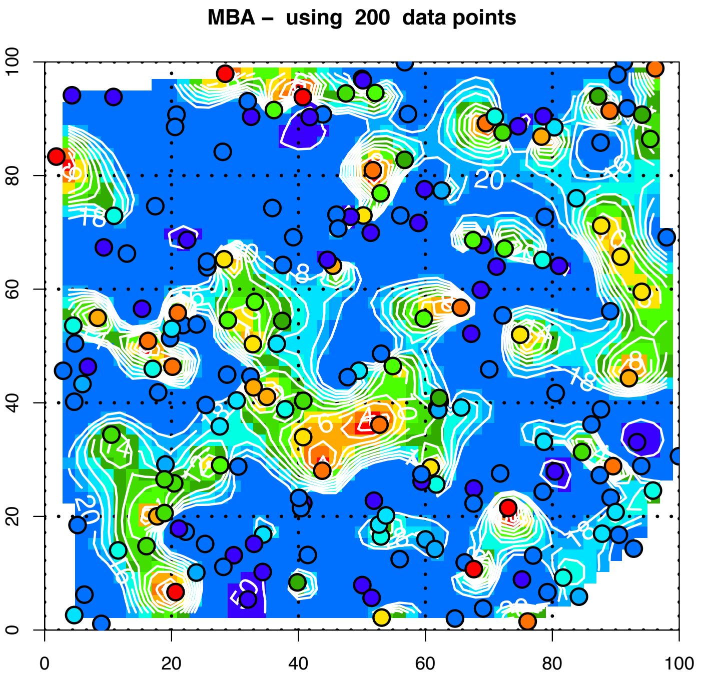


Figure 5

In figure 5 the number of data points has been increased 10 fold, from 20 to 200.

Appendix A

```
#This exercise is all about testing to make sure that the data is plotted correctly.

# Define the color palette
levelcolors <- palette(c("#FF0000", "#FF7100", "#FFAA00", "#FFE300", "#4cff00", "#41de00", "#32ab00", "#00FFE3", "#00E3FF", "#00AAFF", "#0071FF", "#3900FF"))
colorbreaks <- c(0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 50, 1000)
ncolors <- length(colorbreaks)-1

#A bar plot of the palette
barplot(colorbreaks[2:length(colorbreaks)],col=palette(), names.arg=colorbreaks[2:length(colorbreaks)])

n <- 20                       # n can be changed.  It's the number of points to be plotted.
set.seed(30323)               # Fixed so that results are repeatable
# Ref: http://blog.revolutionanalytics.com/2009/02/how-to-choose-a-random-number-in-r.html
x <- runif(n, 1, 100)         # n random x-values from 1 to 100
y <- runif(n, 1, 100)         # n random y-values from 1 to 100

z <- abs(rnorm(n))            # n random normal distribution values
z <- 100.0*z/max(z)           # Normalized to 100 max
zz <- sort(z)                 # Sorted in ascending order

#Apply color palette indices to the z values
colors <- sapply(z,function(x)which.min(abs(x - colorbreaks)))


# Plot points only
xyz <- cbind(x, y, z)
plot(xyz, col=colors, pch=19, cex=2.0, xlim=c(0,100), ylim=c(0,100))
points(x, y, pch=21, cex=2)
grid(lwd=3, col="black")   #Add a grid
# http://www.dummies.com/how-to/content/how-to-round-off-numbers-in-r.html
text(x+7,y,signif(z, digits=2), cex=1.5)      #Add text (point values) adjacent to the points


#Akima interpolation
library(akima)
akima <- interp(x,y,z, duplicate="mean", xo=seq(1, 100, 2), yo=seq(1, 100, 2))
image(akima, col=levelcolors, breaks=colorbreaks, xlim=c(0,100), ylim=c(0,100))
grid(lwd=3, col="black")
contour(akima, labcex=1.5, levels=colorbreaks, add=TRUE, col="white", lwd=2)
title(paste("Akima -  using ", nz, " data points"))
points(x, y, pch=21, bg=colors, lwd=2, cex=2)
```

```


# MBA interpolation
library(MBA)
nx <- 50
s <- data.frame(x, y, z)
mba <- mba.surf(s, nx, nx, h=ncolors)$xyz.est
image(mba, col=levelcolors, breaks=colorbreaks, xlim=c(0,100), ylim=c(0,100))
grid(lwd=3, col="black")
contour(mba, labcex=1.5, levels=colorbreaks, add=TRUE, col="white", lwd=2)
title(paste("MBA -  using ", nz, " data points"))
points(x, y, pch=21, bg=colors, lwd=2, cex=2)
```