

```

In[18]:= (* Similarity (SSD) Sorting of Signals Displayed in Contour Plots *)
(* Copyright April 26, 2006, Doug Youvan,
www.youvan.com & www.pseudocolor.com *)
(* Revised April 30, 2006 *)

(* mock data; z is spectrum number, y is channel , x is intensity;
Make 800 spectra in 9 channels with 10 heights *)

zmax = 800; (* Total number of spectra*)
xmax = 10; (*intensity levels*)
ymax = 10; (* channels *)

(* Initialize lists *)

spec = .
spec = Table[{z}, {z, 1, zmax}];
spec[[All]] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0};

specrand = .
specrand = Table[{z}, {z, 1, zmax}];
specrand[[All]] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0};

mat = .
mat = Table[{z}, {z, 1, zmax}];
mat[[All]] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0};

(* use channel 1 as the spectrum
number from the original input or mock generation *)
(* hardcoded: 5, because it is midrange in y; 1.11,
because of flooring; 1.1 base to broaden peaks *)

For[z = 1, z <= zmax, z++,
spec[[z, 1]] = z / zmax;
For[y = 2, y <= ymax - 2, y++,
spec[[z, (2.0 + Floor[1.11 * (y * (z / zmax))])]] = 1.1 ^ (-1 * ((5 - y) ^ 2))]];

(* make triplet values for RGB ;
N[] converts fractions to decimals for RGB inputs *)

For[z = 1, z <= zmax, z++,
For[y = 1, y <= ymax, y++,
spec[[z, y]] = {N[spec[[z, y]]], N[spec[[z, y]]], N[spec[[z, y]]]}
];

Show[Graphics[RasterArray[Apply[RGBColor, spec, {2}]], ImageSize -> {800, 800}]];

(* Randomize spectral rows: This code can pick the same row multiple times,
or miss rows, but it is simple method to scramble
mock data. Actual data would be input after this step
and below the next comment statement *)

For[z = 1, z <= zmax, z++,
tomove = Random[Integer, {1, zmax}];

```

```

specrand[[z]] = spec[[tomove]]
];

(* save non-triplet form of specrand for matrix sorting *)

For[z = 1, z <= zmax, z++, mat[[z]] = specrand[[z, All, 1]]];

Show[
  Graphics[RasterArray[Apply[RGBColor, specrand, {2}]], ImageSize -> {800, 800}]];

(* Similarity sort based on multiple scans using sum-
of-the-square-of-the-differences (SSD) over all channels,
except channel 1, where there is a grayscale-encoded signal number marker;
SSD comparison is with the progressively last row sorted;
the outer-most scan loop advances 'z' to avoid already sorted rows;
the 'goodness' of the sort is indicated by the re-
formation of the channel 1 grayscale column,
that serves here as a 'passenger' and thus an experimental control *)

mat[[1]] = {0, 0.04, 0, 0, 0, 0, 0, 0, 0, 0}; (* target for intializing sort *)
(* signal in row 1 is lost by this method,
and the last 2 signals remain unsorted *)

For[
  scan = 1, scan < zmax, scan++, ssd = 10;
For[
  z = scan + 1, z < zmax, z++,
  For [
    y = 1, y <= 10, y++,
    If[y == 1, ssdtemp = 0, ssdtemp = ssdtemp + (mat[[scan, y]] - mat[[z, y]])^2 ]
  ];
  If[ssdtemp <= ssd, ssd = ssdtemp; up = z; moveup = mat[[z]] ]
];
mat[[up]] = mat[[scan + 1]];
mat[[scan + 1]] = moveup
];

(* Mathematica plot *)

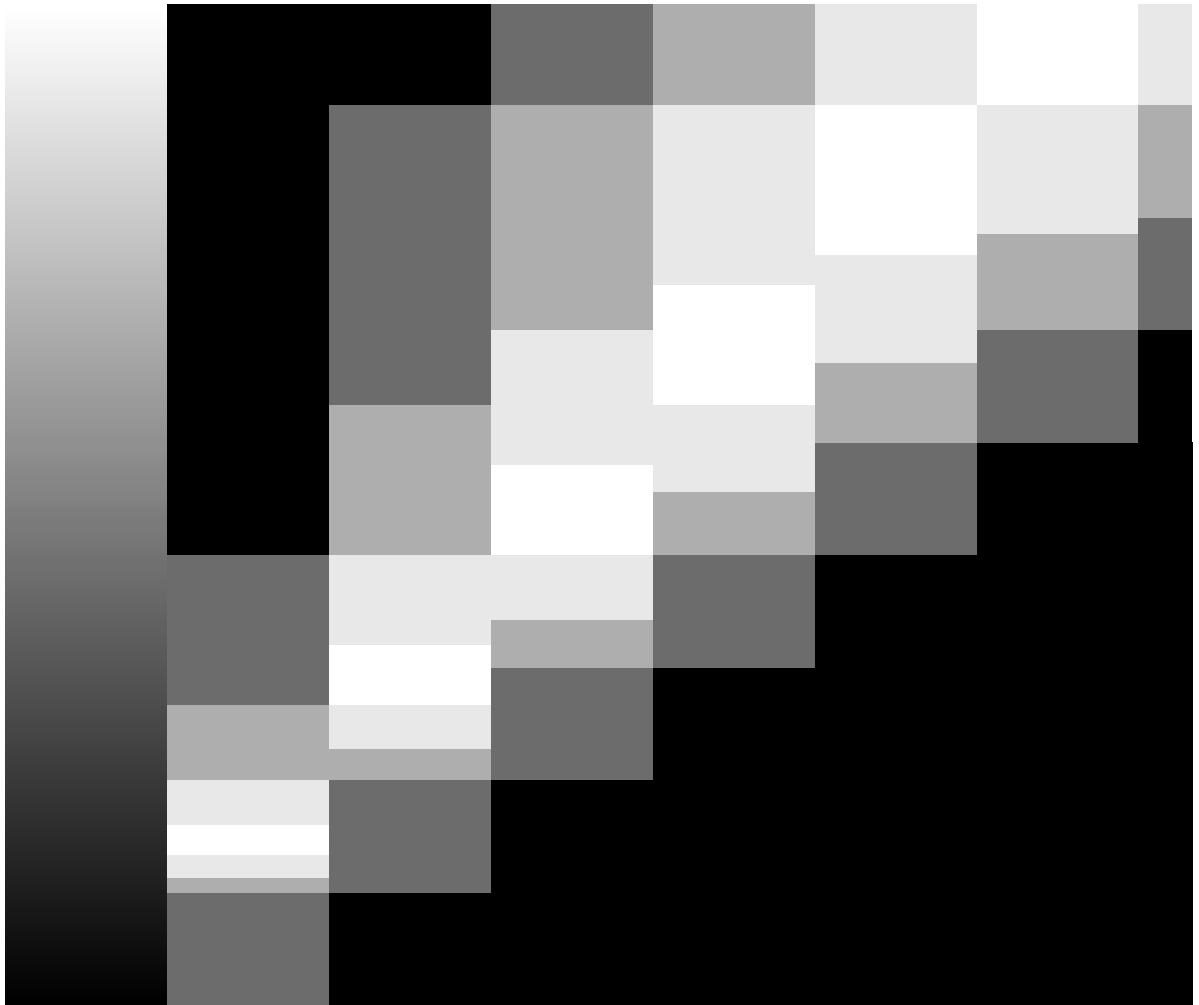
ListPlot3D[mat]

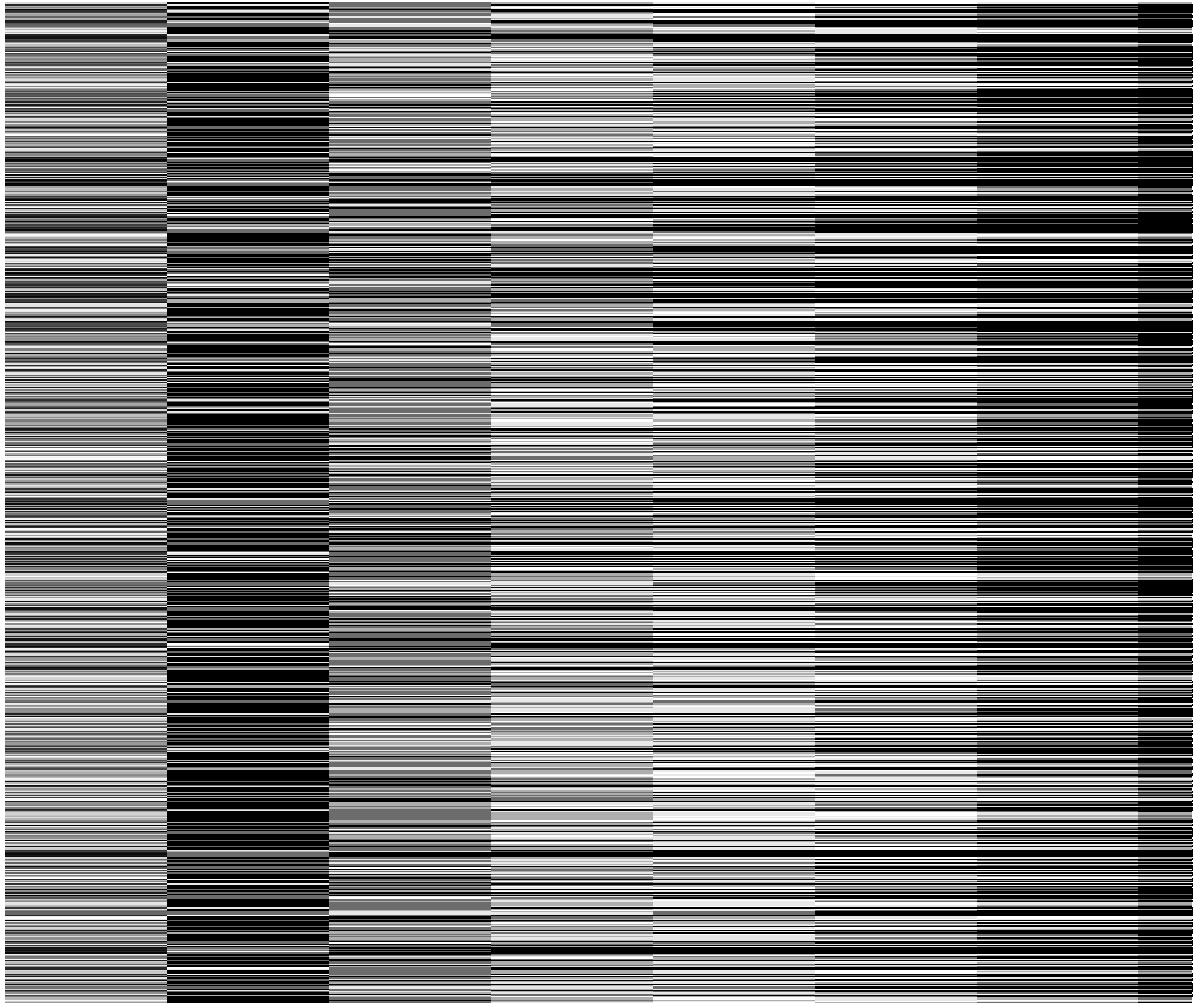
(* make triplet values for RGB *)

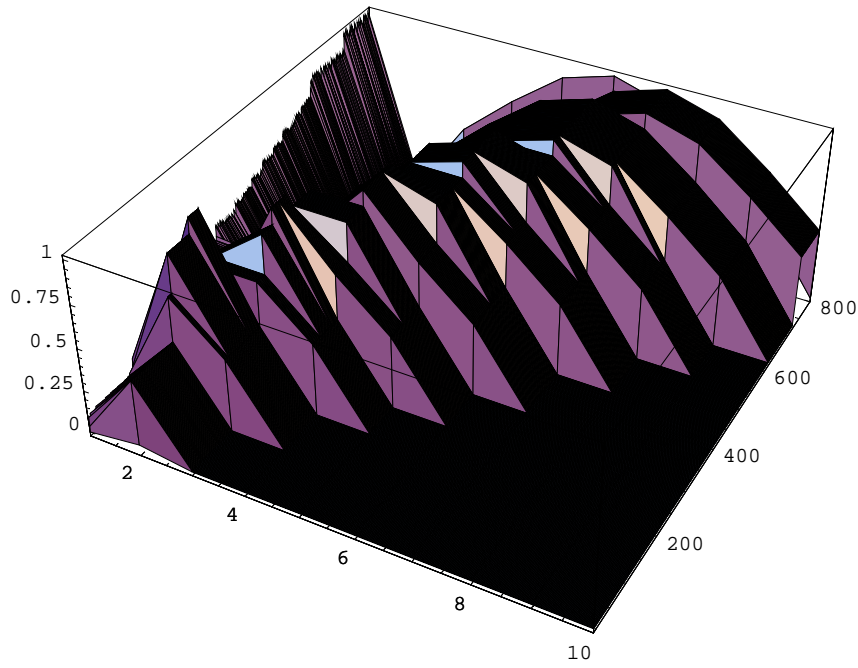
For[z = 1, z <= zmax, z++,
  For[y = 1, y <= ymax, y++,
    mat[[z, y]] = {N[mat[[z, y]]], N[mat[[z, y]]], N[mat[[z, y]]]}
  ];

Show[Graphics[RasterArray[Apply[RGBColor, mat, {2}]], ImageSize -> {800, 800}]];

```







Out[32]= - SurfaceGraphics -

