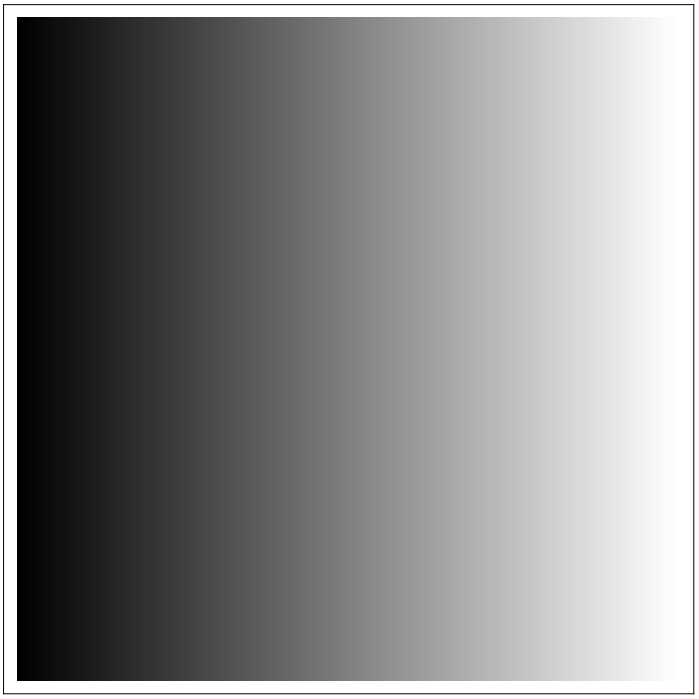
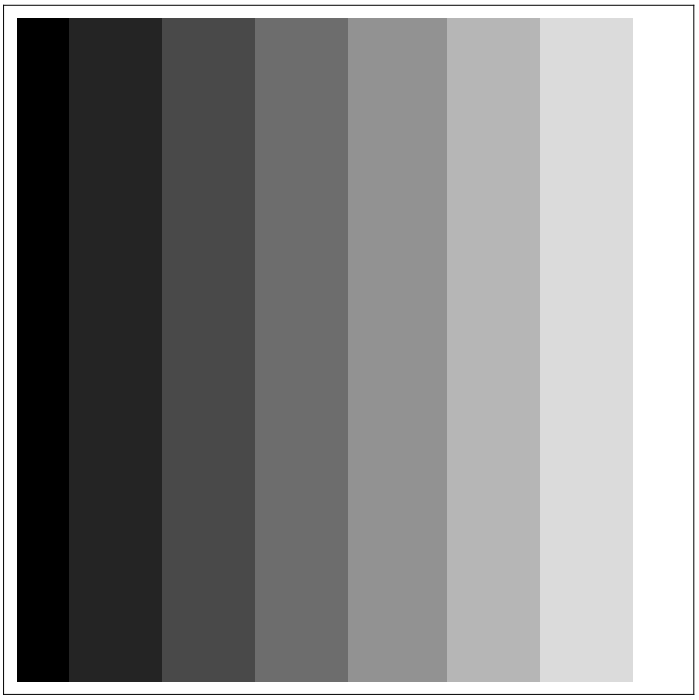
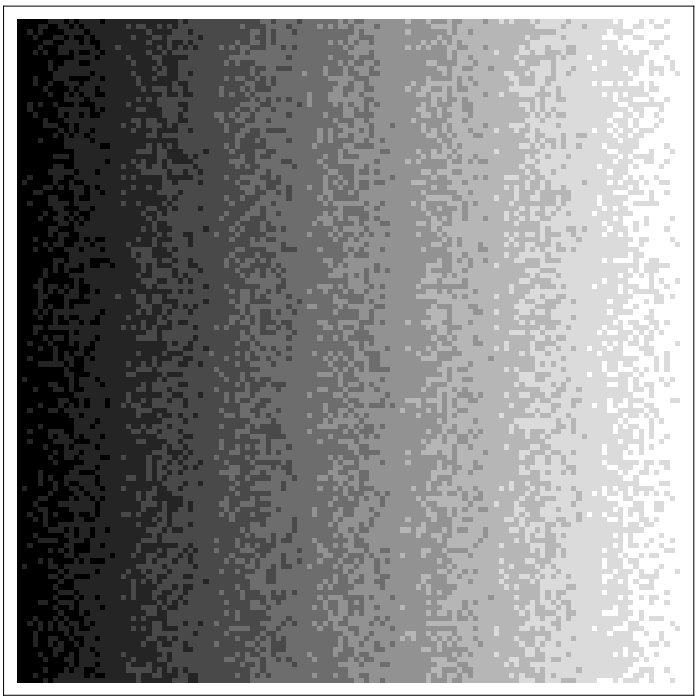
```
SetDirectory@NotebookDirectory[];
QuantizeBits[x_, bits_, dither_: 0] :=
 Clip[Round[x * (Power[2, bits] - 1) + dither] / (Power[2, bits] - 1)]
remaptri[v_] := (orig = FractionalPart[v + 0.5] * 2 - 1;
  If[Abs[orig] == 0, -1, orig / Sqrt[Abs[orig]]] - Sign[orig])

SampleArrayWithWrap[arr_, x_, y_] :=
 Module[{dims = Dimensions[arr]}, arr[[Mod[x, dims[[1]], 1], Mod[y, dims[[2]], 1]]]]
origSignal = Table[x / 128, {y, 0, 127}, {x, 0, 127}];
ArrayPlot[origSignal, ColorFunction → GrayLevel]
```



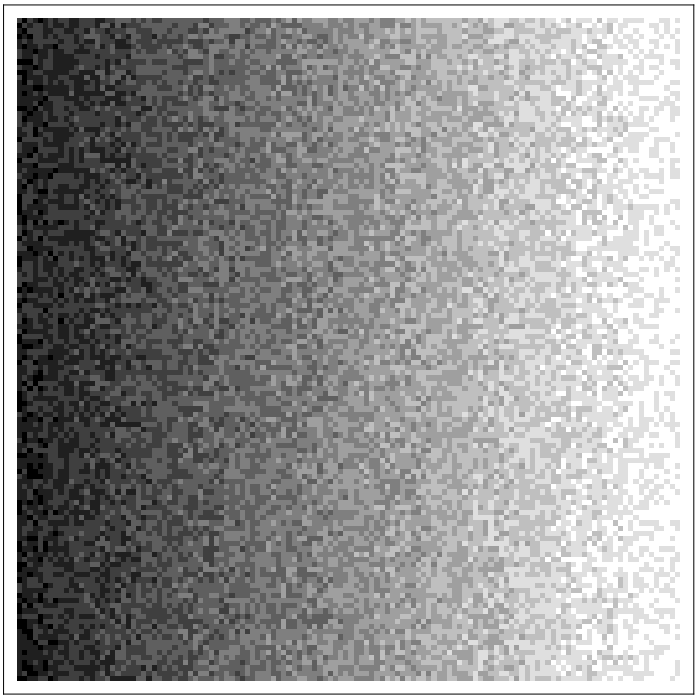
```
ArrayPlot[QuantizeBits[#1, 3] & /@ origSignal, ColorFunction → GrayLevel]
```



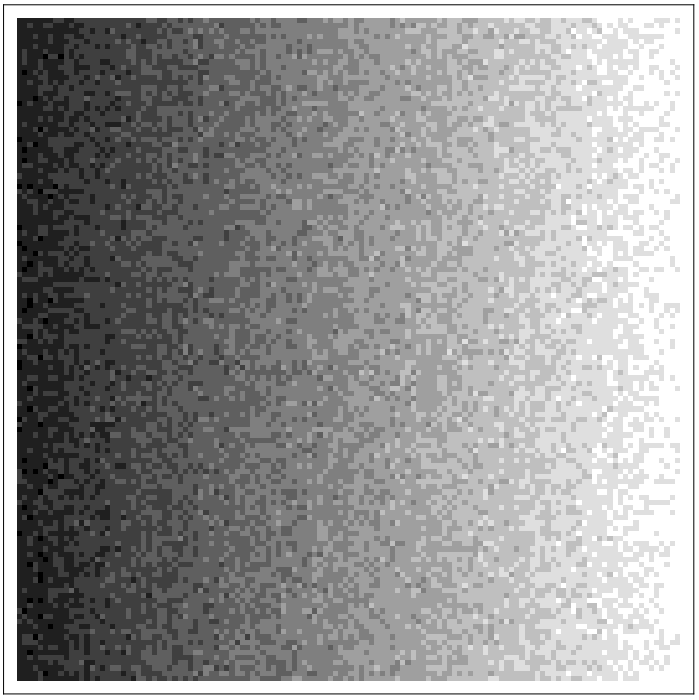
```
ditheredWhite = Map[QuantizeBits[#1, 3, RandomReal[] - 0.5] &, origSignal, {2}];
ArrayPlot[ditheredWhite, ColorFunction → GrayLevel]
```



```
ArrayPlot[Map[QuantizeBits[#1, 3, (RandomReal[] - 0.5) * 2] &, origSignal, {2}],
 ColorFunction → GrayLevel]
```

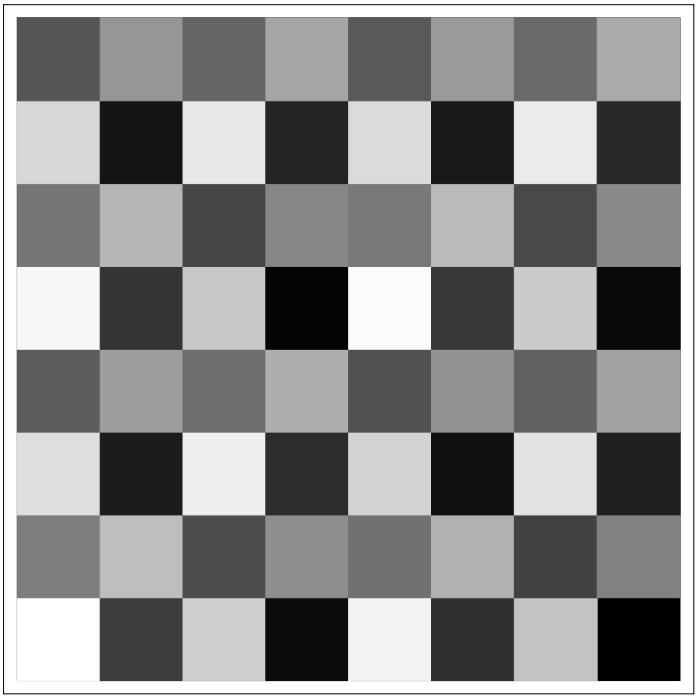


```
ArrayPlot[Map[QuantizeBits[#1, 3, (RandomReal[] + RandomReal[] - 1)] &,
  origSignal, {2}], ColorFunction → GrayLevel]
```

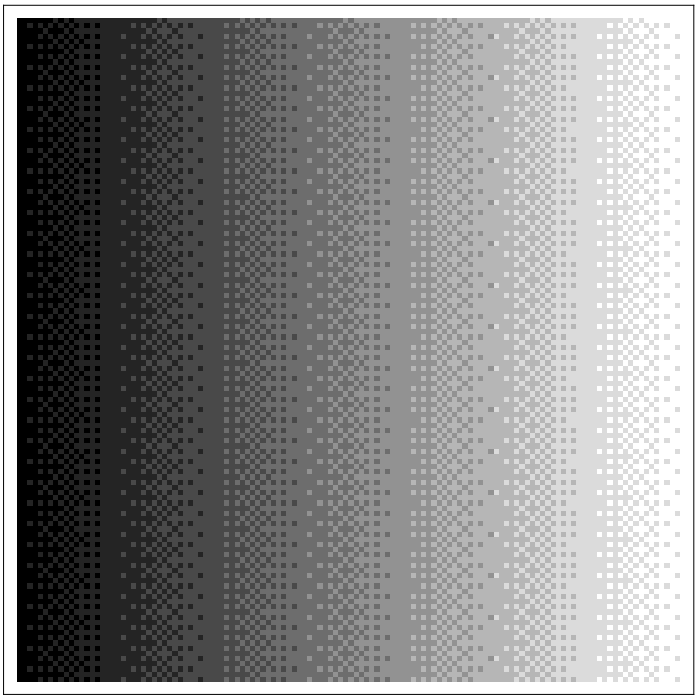


```
Bayer[x_, y_, level_] :=
 Mod[Mod[BitShiftRight[x, level], 2] + 1 + 2 * Mod[BitShiftRight[y, level], 2], 4] +
  If[level == 0, 0, 4 * Bayer[x, y, level - 1]]
bayer3 = Table[Bayer[x, y, 2], {y, 0, 7}, {x, 0, 7}] / 63;
bayer6 = Table[Bayer[x, y, 5], {y, 0, 63}, {x, 0, 63}] / (64 * 64 - 1);

ditheredBayer = MapIndexed[
   QuantizeBits[#1, 3, SampleArrayWithWrap[bayer3, #2[[1]], #2[[2]]] - 0.5] &,
   origSignal, {2}];
```

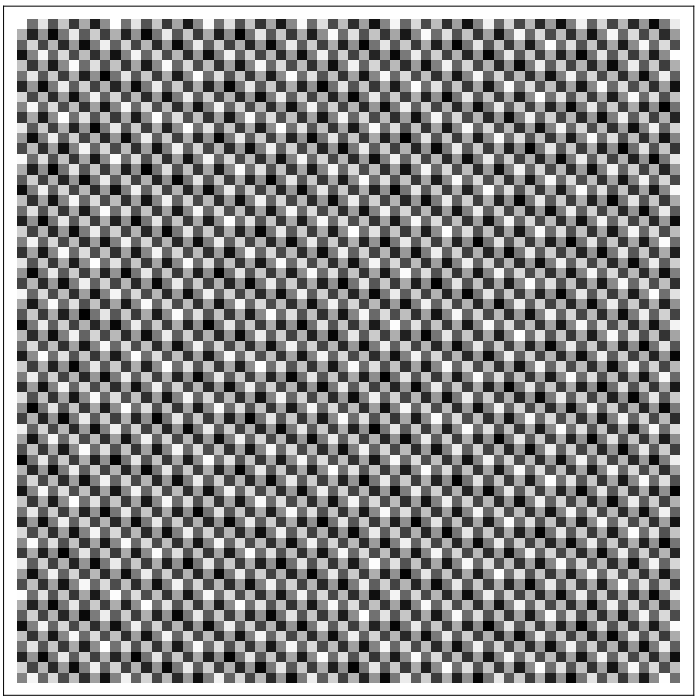
```
ArrayPlot[bayer3, ColorFunction → GrayLevel]
```



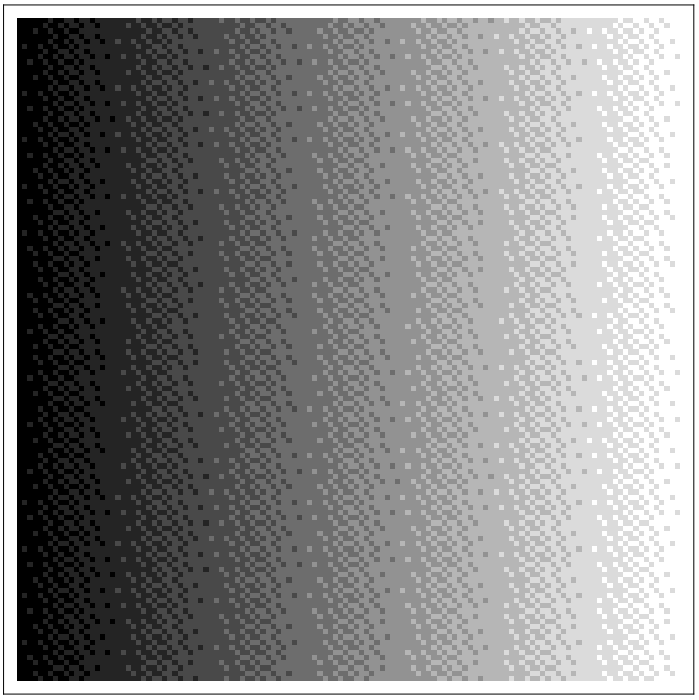
```
ArrayPlot[ditheredBayer, ColorFunction → GrayLevel]
```



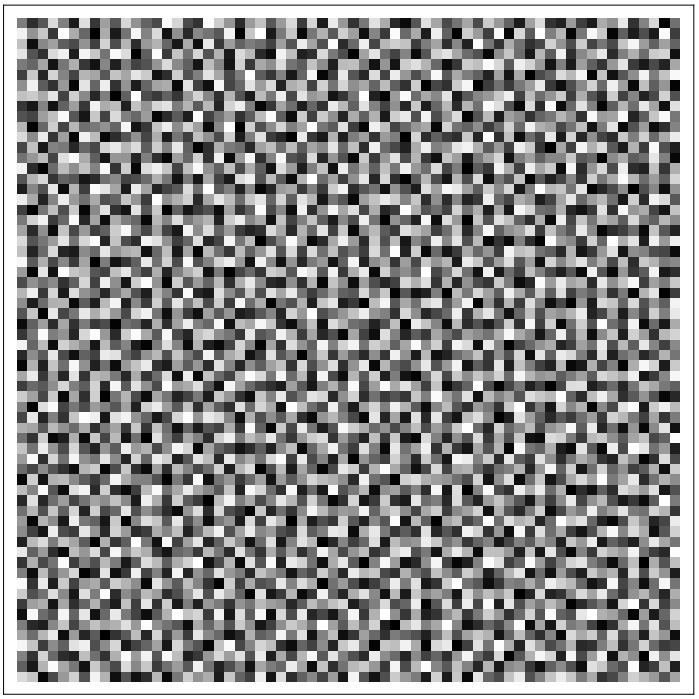
```
InterleavedGradientNoise[x_, y_] :=
 FractionalPart[52.9829189 * FractionalPart[0.06711056 * x + 0.00583715 * y]]

ArrayPlot[Table[InterleavedGradientNoise[x, y], {y, 0, 63}, {x, 0, 63}]]
```



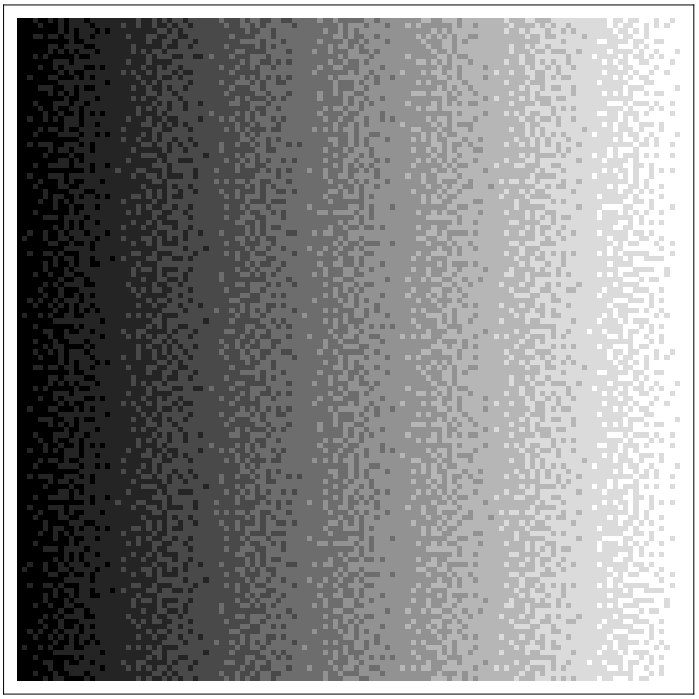
```
ditheredInterleaved =
  MapIndexed[QuantizeBits[#1, 3, InterleavedGradientNoise[#2[[1]], #2[[2]]] - 0.5] &,
   origSignal, {2}];
ArrayPlot[ditheredInterleaved, ColorFunction → GrayLevel]
```



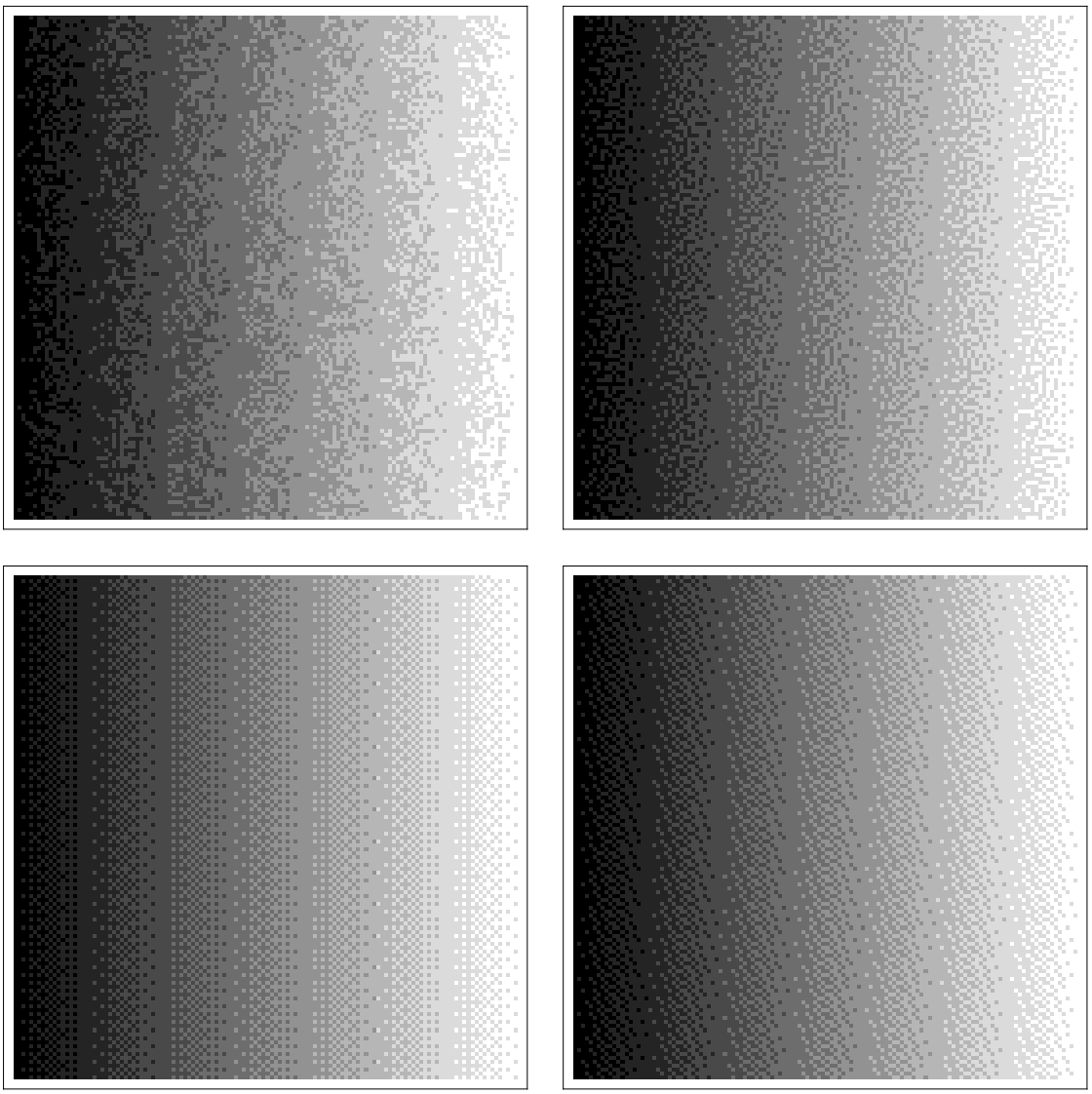
```
ArrayPlot[blueNoise64, ColorFunction → GrayLevel]
```



```
ditheredBlue = MapIndexed[
   QuantizeBits[#1, 3, SampleArrayWithWrap[blueNoise64, #2[[1]], #2[[2]]] - 0.5] &,
   origSignal, {2}];
ArrayPlot[ditheredBlue, ColorFunction → GrayLevel]
```

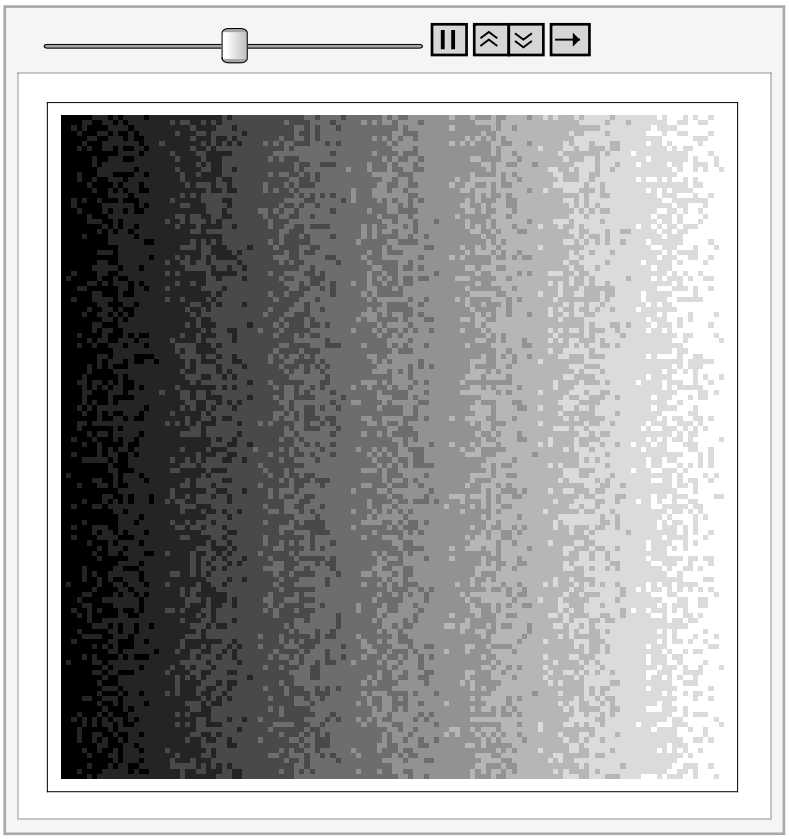


```
GraphicsGrid[{{ArrayPlot[ditheredWhite, ColorFunction → GrayLevel],
    ArrayPlot[ditheredBlue, ColorFunction → GrayLevel]},
   {ArrayPlot[ditheredBayer, ColorFunction → GrayLevel],
    ArrayPlot[ditheredInterleaved, ColorFunction → GrayLevel]}}]
Export["2dditheringalltechniquescompared.gif",
   {ArrayPlot[ditheredWhite, ColorFunction → GrayLevel],
    ArrayPlot[ditheredBlue, ColorFunction → GrayLevel],
    ArrayPlot[ditheredBayer, ColorFunction → GrayLevel], ArrayPlot[
     ditheredInterleaved, ColorFunction → GrayLevel]}, "DisplayDurations" → 2];
```



```
Export["2dditheringquandizecomparisongrid.png",
  GraphicsGrid[{{ArrayPlot[ditheredWhite, ColorFunction → GrayLevel],
     ArrayPlot[ditheredBlue, ColorFunction → GrayLevel]},
    {ArrayPlot[ditheredBayer, ColorFunction → GrayLevel],
     ArrayPlot[ditheredInterleaved, ColorFunction → GrayLevel]}}]];
```

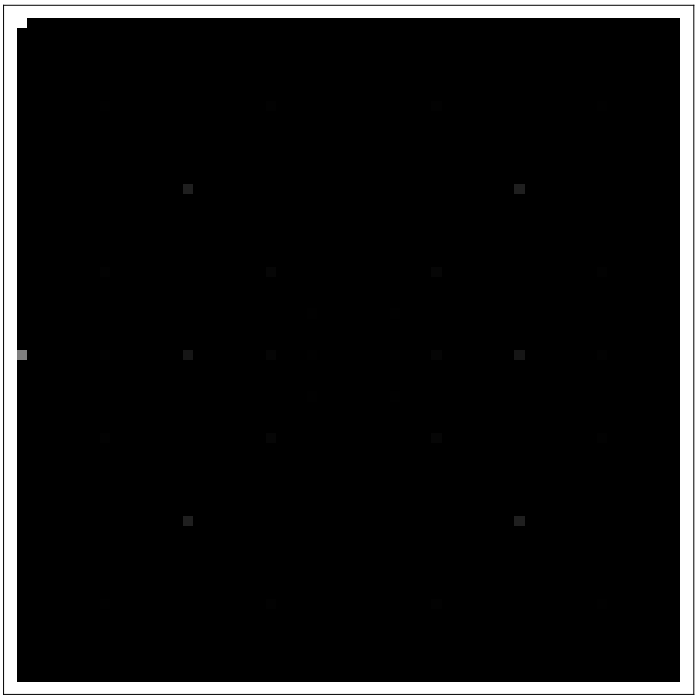
```
ListAnimate[{ArrayPlot[Abs[ditheredWhite - origSignal], ColorFunction → GrayLevel],
  ArrayPlot[ditheredWhite, ColorFunction → GrayLevel],
  ArrayPlot[origSignal, ColorFunction → GrayLevel]}]
```



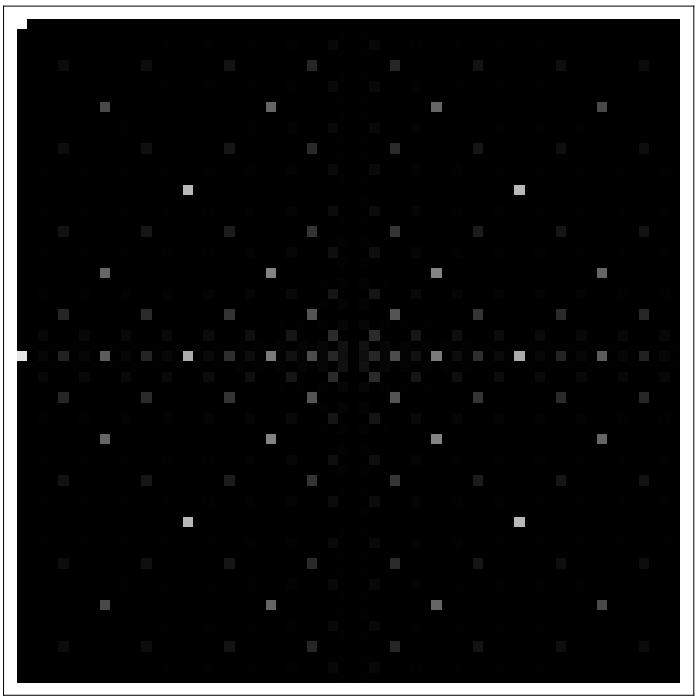
```
Export["2dditheringwhitebands.gif",
  {ArrayPlot[Abs[ditheredWhite - origSignal], ColorFunction → GrayLevel],
   ArrayPlot[ditheredWhite, ColorFunction → GrayLevel],
   ArrayPlot[origSignal, ColorFunction → GrayLevel]}, "DisplayDurations" → 2];
```

```
MyTwoDimPeriodogram[arr_] := ArrayPlot[
  Abs[RotateLeft[Fourier[arr], Dimensions[arr] / 2]], ColorFunction → GrayLevel]
```

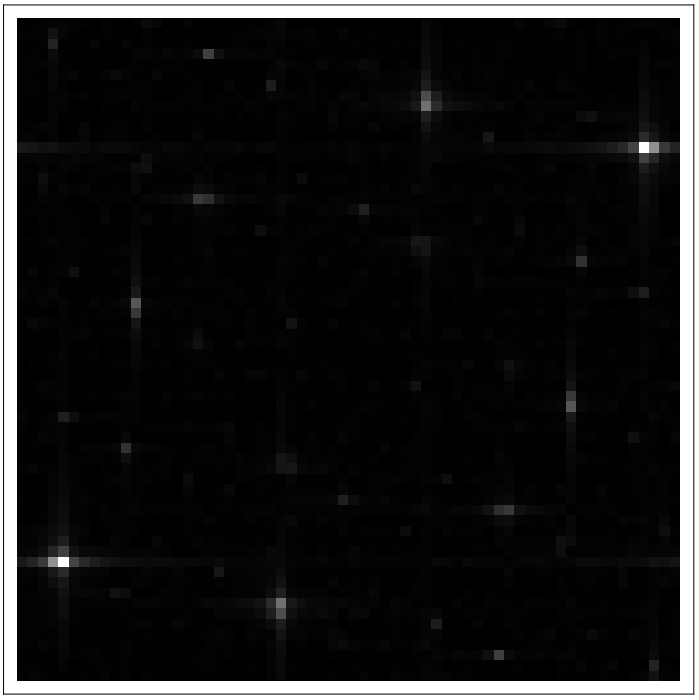
```
MyTwoDimPeriodogram[bayer6 - 0.5]
```

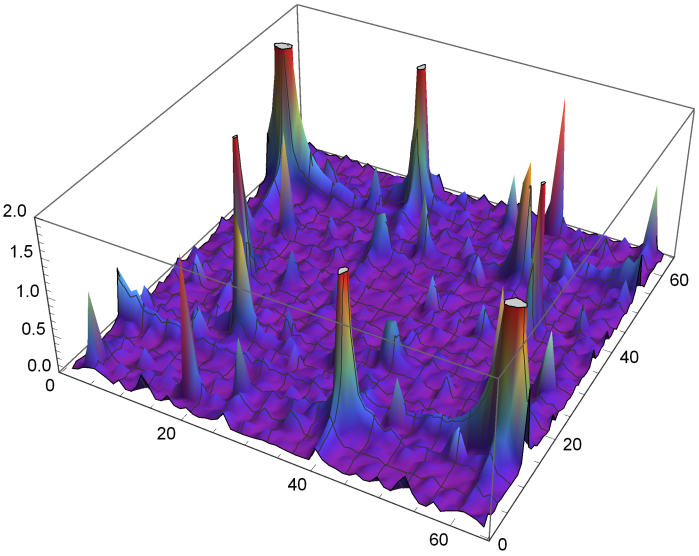


```
ArrayPlot[Log[0.01 + Abs[RotateLeft[Fourier[bayer6 - 0.5], Dimensions[bayer6] / 2]]],
 ColorFunction → GrayLevel]
```

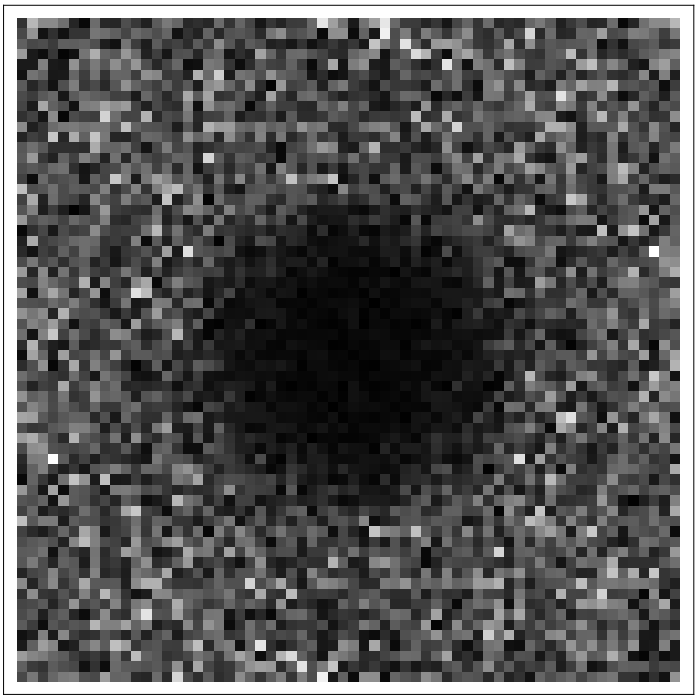


```
MyTwoDimPeriodogram[
 Table[InterleavedGradientNoise[x, y], {y, 0, 63}, {x, 0, 63}] - 0.5]
ListPlot3D[Abs[RotateLeft[Fourier[
    Table[InterleavedGradientNoise[x, y], {y, 0, 63}, {x, 0, 63}] - 0.5], {32, 32}]],
 PlotRange → {0, 2}, ColorFunction → "Rainbow"]
```

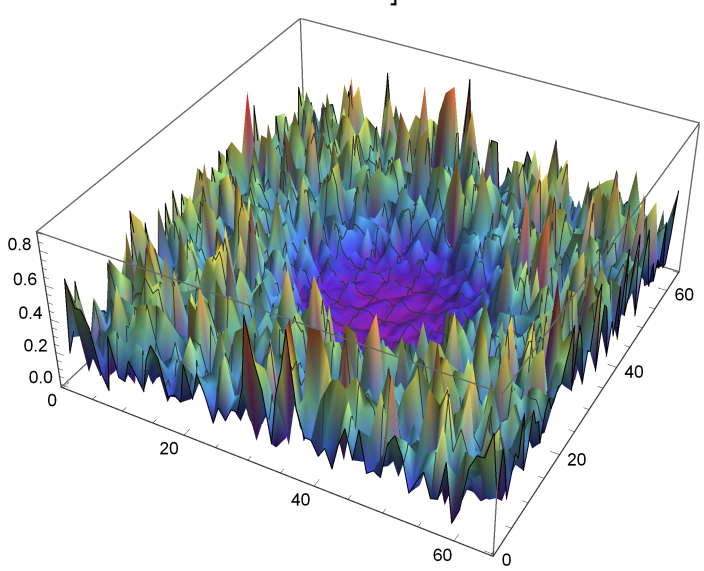




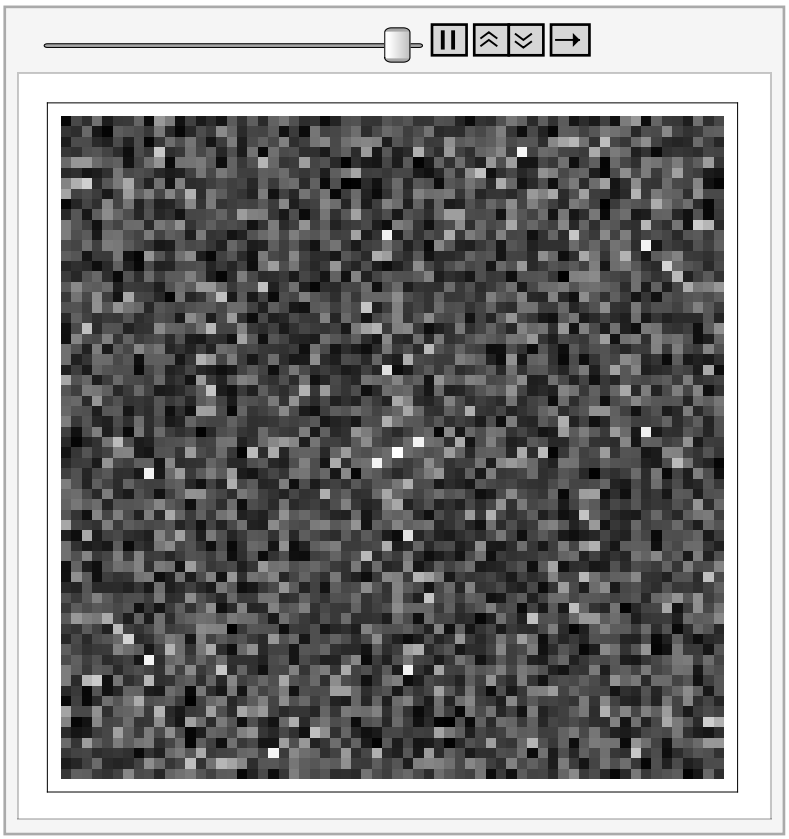
```
MyTwoDimPeriodogram[blueNoise64 - 0.5]
```



```
ListPlot3D[Abs[RotateLeft[Fourier[blueNoise64 - 0.5], Dimensions[bayer6] / 2]],
 ColorFunction → "Rainbow"]
```

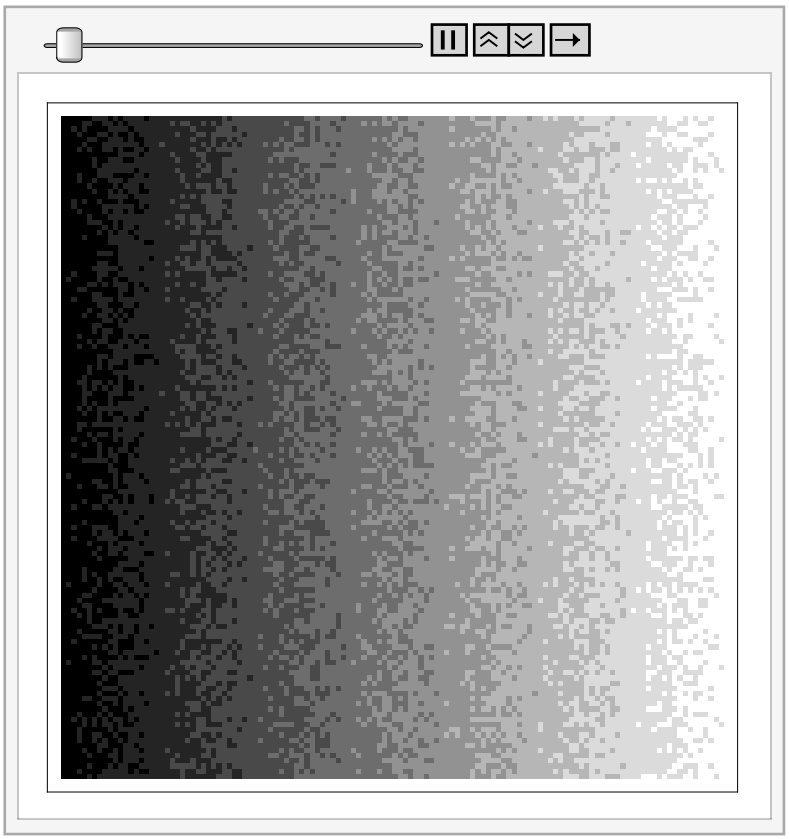


```
ListAnimate[{MyTwoDimPeriodogram[blueNoise64 - 0.5],
  MyTwoDimPeriodogram[Table[RandomReal[], {x, 0, 63}, {y, 0, 63}] - 0.5]}]
```



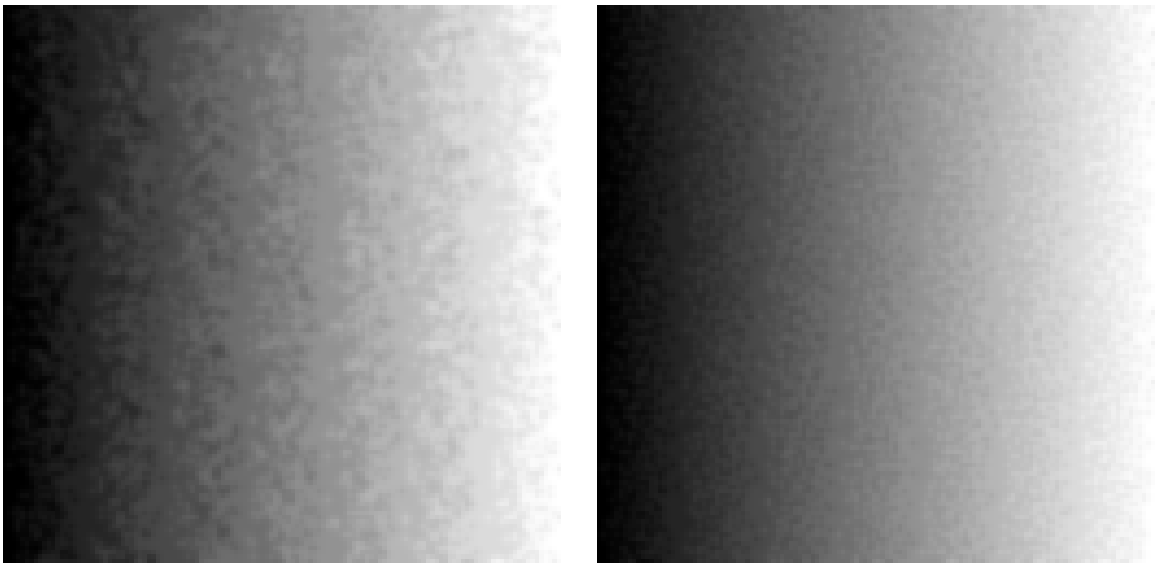
```
Export["2dwhitenoisevsbluenoise.gif",
  {MyTwoDimPeriodogram[blueNoise64 - 0.5], MyTwoDimPeriodogram[
    Table[RandomReal[], {x, 0, 63}, {y, 0, 63}] - 0.5]}, "DisplayDurations" → 2];
```

```
ListAnimate[{ArrayPlot[ditheredWhite, ColorFunction → GrayLevel],
  ArrayPlot[ditheredBlue, ColorFunction → GrayLevel]}]
```

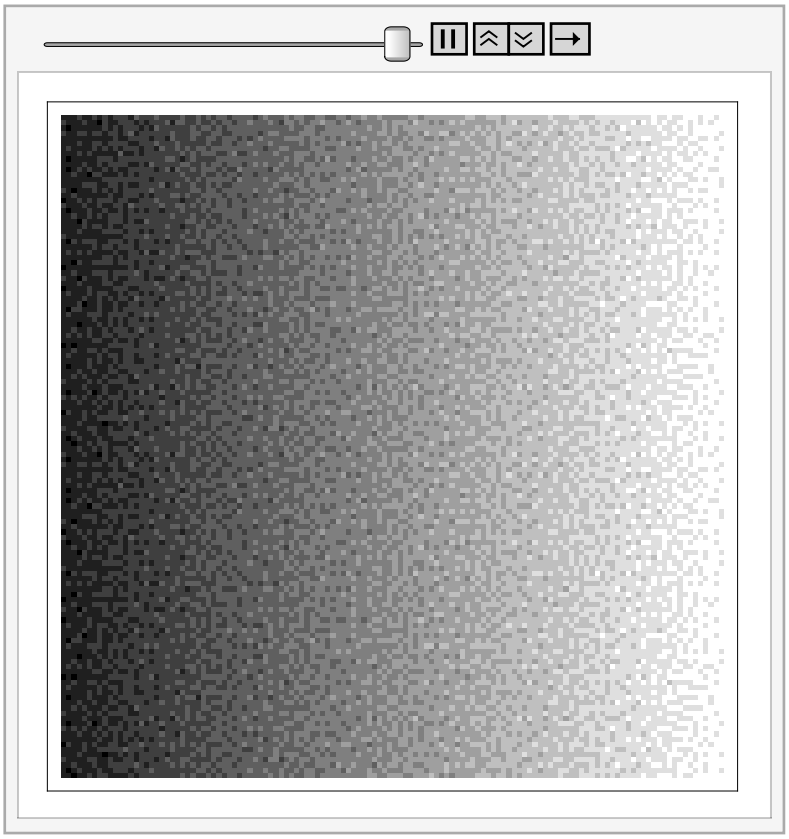


```
Export["2dwhitenoisevsbluenoise.gif",
  {ArrayPlot[ditheredWhite, ColorFunction → GrayLevel],
   ArrayPlot[ditheredBlue, ColorFunction → GrayLevel]}, "DisplayDurations" → 2];
```

```
GraphicsRow[
 {GaussianFilter[Image[ditheredWhite], 2], GaussianFilter[Image[ditheredBlue], 2]}]
```



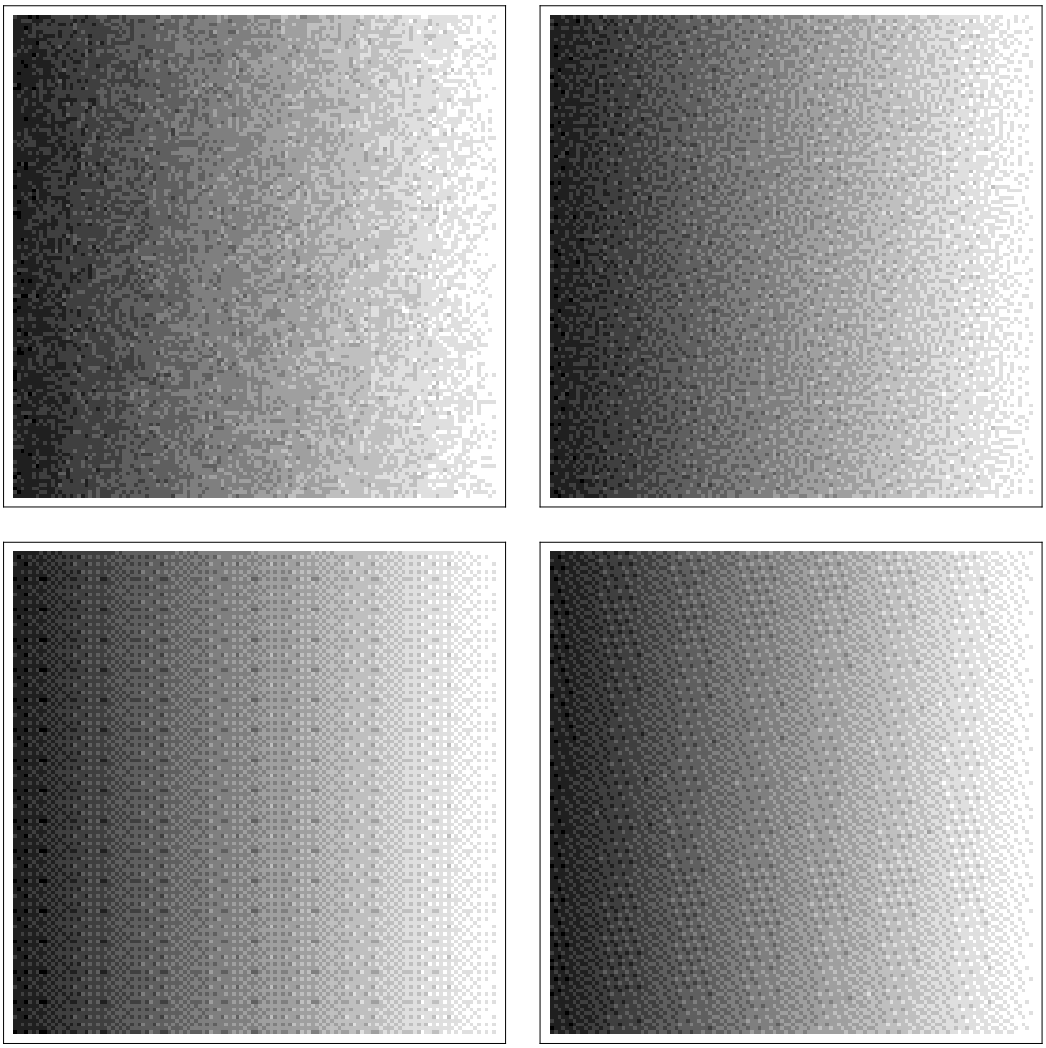
```
ListAnimate[
 {ArrayPlot[Map[QuantizeBits[#1, 3, (RandomReal[] + RandomReal[] - 1)] &, origSignal,
    {2}], ColorFunction → GrayLevel], ArrayPlot[MapIndexed[QuantizeBits[#1,
      3, remaptri[SampleArrayWithWrap[blueNoise64, #2[[1]], #2[[2]]]]] &,
    origSignal, {2}], ColorFunction → GrayLevel]}]
```



```
Export["2dwhitenoisevsbluenoisetriangle.gif",
  {ArrayPlot[Map[QuantizeBits[#1, 3, (RandomReal[] + RandomReal[] - 1)] &, origSignal,
     {2}], ColorFunction → GrayLevel], ArrayPlot[MapIndexed[QuantizeBits[#1,
       3, remaptri[SampleArrayWithWrap[blueNoise64, #2[[1]], #2[[2]]]]] &,
     origSignal, {2}], ColorFunction → GrayLevel]}, "DisplayDurations" → 2];
```

```
ditheredInterleavedRemap = MapIndexed[
   QuantizeBits[#1, 3, remaptri[InterleavedGradientNoise[#2[[1]], #2[[2]]]]] &,
   origSignal, {2}];
```

```
ditheredBayerRemap = MapIndexed[
   QuantizeBits[#1, 3, remaptri[SampleArrayWithWrap[bayer3, #2[[1]], #2[[2]]]]] &,
   origSignal, {2}];
ditheredWhiteRemap = Map[QuantizeBits[#1, 3, remaptri[RandomReal[]]] &,
   origSignal, {2}];

ditheredBlueRemap = MapIndexed[QuantizeBits[#1, 3, remaptri[
      SampleArrayWithWrap[blueNoise64, #2[[1]], #2[[2]]]]] &, origSignal, {2}];

GraphicsGrid[{{ArrayPlot[ditheredWhiteRemap, ColorFunction → GrayLevel],
   ArrayPlot[ditheredBlueRemap, ColorFunction → GrayLevel]},
  {ArrayPlot[ditheredBayerRemap, ColorFunction → GrayLevel],
   ArrayPlot[ditheredInterleavedRemap, ColorFunction → GrayLevel]}}]
```



```
Export["2ddithering allfourcomparedtriangular.gif",
  {ArrayPlot[ditheredWhiteRemap, ColorFunction → GrayLevel],
   ArrayPlot[ditheredBlueRemap, ColorFunction → GrayLevel],
   ArrayPlot[ditheredBayerRemap, ColorFunction → GrayLevel], ArrayPlot[
    ditheredInterleavedRemap, ColorFunction → GrayLevel]}, "DisplayDurations" → 2];
```