Computational Vision

Signal-in-noise psychophysics demo

U. Minn. Psy 5036

Daniel Kersten

Lecture 5

GaborSKEDetection.nb

```
stdnoise = 1.0;
ndist =
  NormalDistribution[0,
   stdnoise];
size = 64;
(* image size *)
i = 0;
pc = 0;
```

Signal-in-noise psychophysics demo

Initialize

In[696]:=

```
Off[General::spell1]
```

In[698]:=

```
z[p_] :=
 Sqrt[2]
  InverseErf[1 - 2 p];
dprime[x_] :=
 N[-Sqrt[2] z[x], 2];
```

Various frequencies, vertical orientations, and fixed width

```
stdnoise = 1.0;
ndist =
  NormalDistribution[0,
   stdnoise];
size = 64;
(* image size *)
i = 0;
pc = 0;
```

```
In[705]:= vtheta = Table[i1 Pi/4, {i1,4}]
vf = {1,1,2,4};
swidth = {.25,1,4};
```

Define test images

Basis set: Cartesian representation of Gabor functions:

```
In[703]:= cgabor[x_,y_, fx_, fy_,s_] :=
Exp[-(x^2 +  y^2)/s^2] Cos[2 Pi
sgabor[x_,y_, fx_, fy_, s_] :=
Exp[-(x^2 +  y^2)/s^2] Sin[2 Pi
```

Various frequencies , vertical orientations, and fixed width

```
In[705]:= vtheta = Table[i1 Pi/4, {i1,4}]
vf = {1,1,2,4};
swidth = {.25,1,4};
```

```
In[708]:= signal =
  Table[
    N[cgabor[x, y,
        vf[[1]] Cos[vtheta[[1]]],
        vf[[1]] Sin[vtheta[[1]]],
        swidth[[2]]]],
    {x, -2, 2, 4/(size - 1)},
    {y, -2, 2, 4/(size - 1)}];
Print[Max[signal], "  ",
  Min[signal], "  ",
  Dimensions[signal]];
noise :=
  Table[RandomReal[ndist],
   {size}, {size}];
```

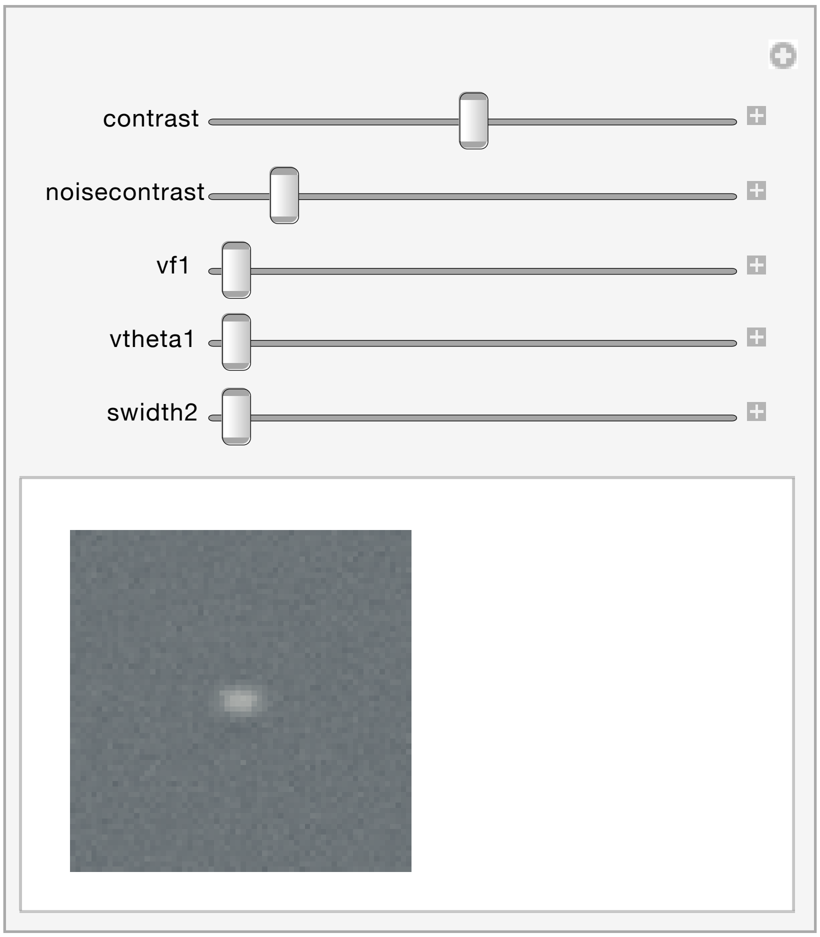
```
0.997986
 -0.782989  {64, 64}
```

```
In[708]:= signal =
    Table[
     N[cgabor[x, y,
       vf⟦1⟧ Cos[vtheta⟦1⟧],
       vf⟦1⟧ Sin[vtheta⟦1⟧],
       swidth⟦2⟧]],
     {x, -2, 2, 4/(size - 1)},
     {y, -2, 2, 4/(size - 1)}];
   Print[Max[signal], "   ",
    Min[signal], "   ",
    Dimensions[signal]];
   noise :=
    Table[RandomReal[ndist],
     {size}, {size}];

   0.997986
    -0.782989   {64, 64}
```

```
signal2 =
  Table[contrast *
    N[cgabor[x, y,
      vf1 Cos[vtheta1],
      vf1 Sin[vtheta1],
      swidth2]],
   {x, -2, 2, 4/(size - 1)},
   {y, -2, 2, 4/(size - 1)}];
noises = noisecontrast *
   noise;

ArrayPlot[signal2 + noises,
  Mesh → False,
  PlotRange → {-1, 1},
  ColorFunction →
   "GrayTones"],
{{contrast, .5}, 0, 1},
{{noisecontrast, .03},
```

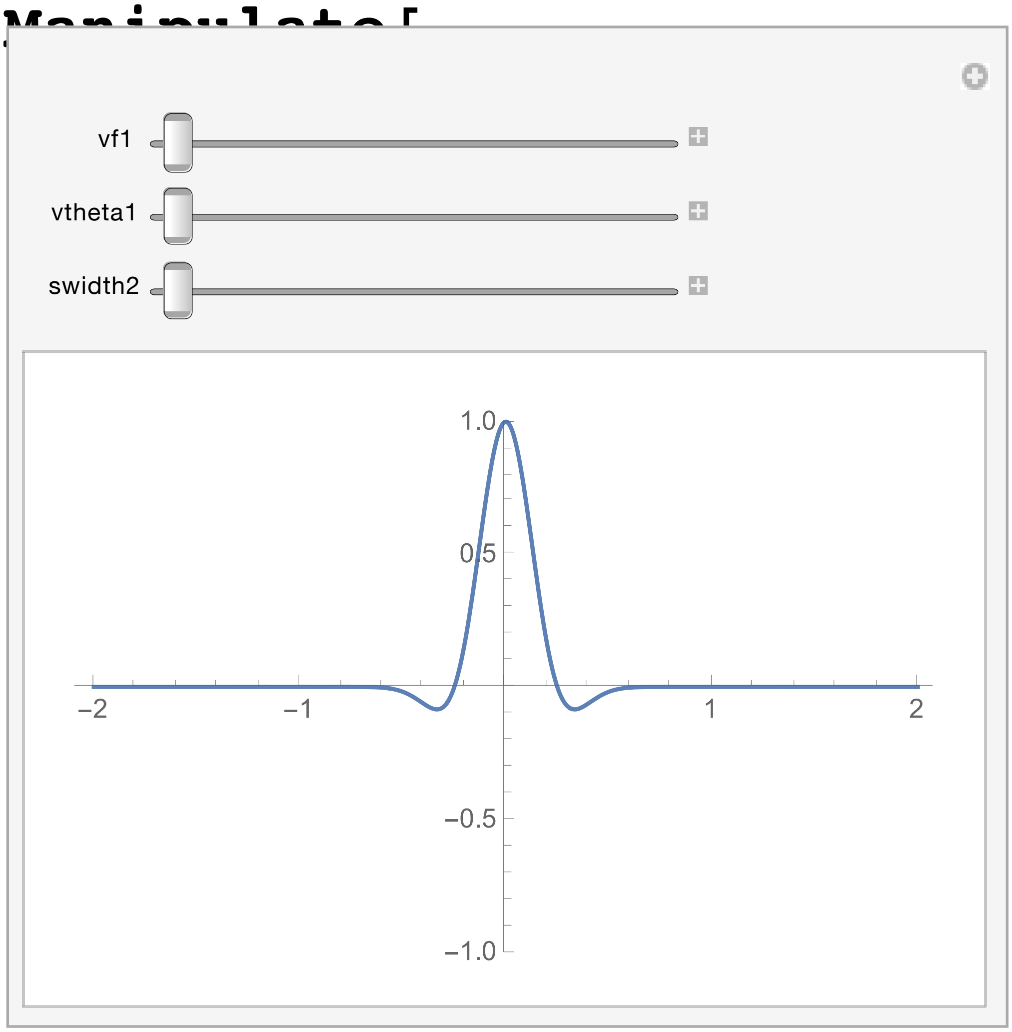
```
0, .3}, {vf1, 1, 4},
{vtheta1, 0, Pi},
{swidth2, .25, 4}]
```

Out[711]=

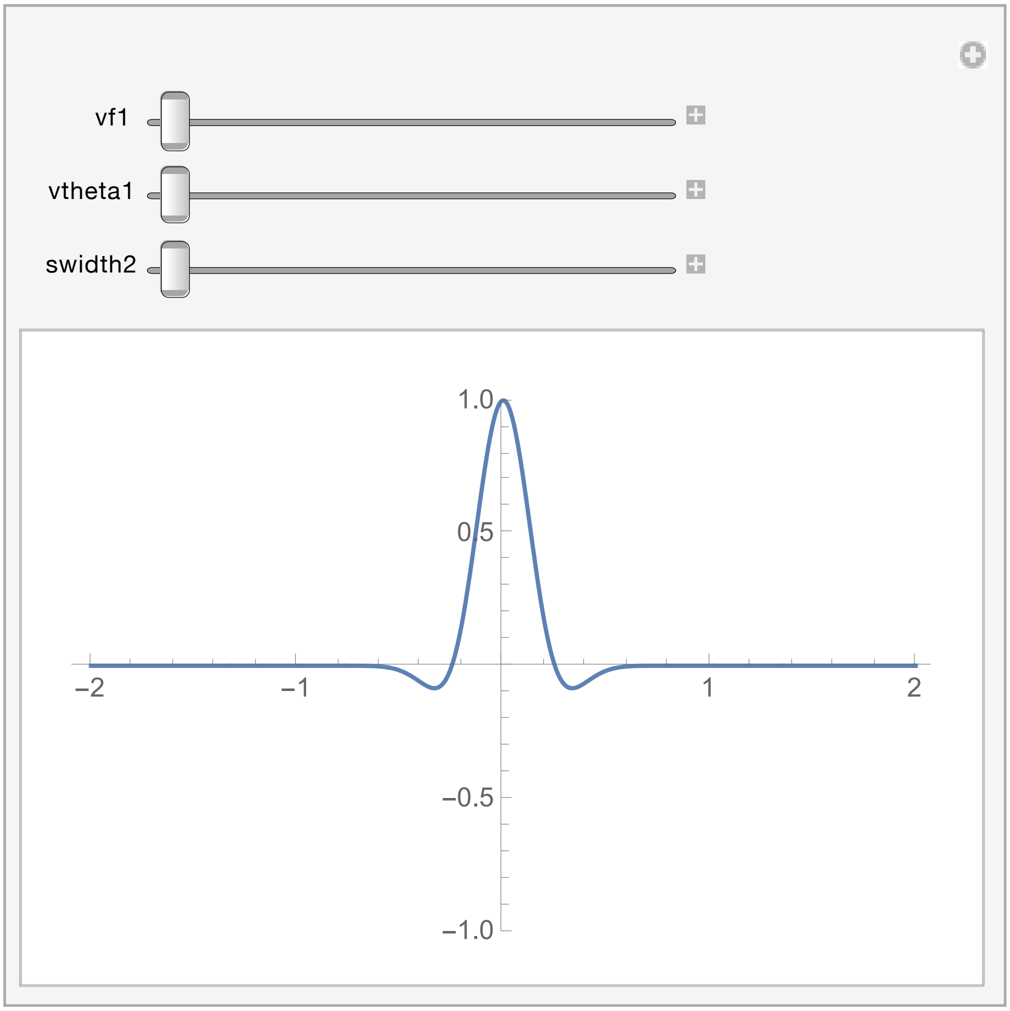


In[720]:= Manipulate[

Out[720]=



Out[720]=



```
In[724]:= blank = Table[0.0,
    {{x, -2, 2, 4 / (size - 1)},
     {y, -2, 2,
      4 / (size - 1)}}];
blank =
   ArrayPlot[blank,
    Mesh → False,
    Frame → False,
    PlotRange → {-1, 1},
    ColorFunction →
     "GrayTones"];
flash = blank;
```

```
In[721]:= data =
    {"Was I Correct?",
     "Was Ideal
      Correct?"};
g[illegible]als = 5;
(*Signal and noise
  contrasts*)
scon = 0.1;
(*scon =
  0.01 is closer to
  threshold*)
ncon = .15;
```

```
In[724]:= data =
    {{"Was I Correct?",
      "Was Ideal
        Correct?"}};
  numtrials = 5;

  (*Signal and noise
   contrasts:*)
  scon = 0.1;
  (*scon =
   0.01 is closer to
    threshold.*)
  ncon = .15;
```

Put up stimulus window

In[728]:=
```
nb = CreateDocument[
    Dynamic[flash],
    ShowCellBracket → False,
    WindowSize → {300, 300},
    WindowMargins →
     {{Automatic, 0},
      {Automatic, 0}},
    WindowElements → {},
    Background → Black,
    NotebookFileName →
     "2AFC Pattern
       Detection"];
```

Define a trial

In[729]:=
```
twoflashes :=
  Module[{tempmean},
   Table[whichflash =
      RandomInteger[{0, 1}];
     If[whichflash == 1,
```

```
leftnumsample =
 ArrayPlot[
  leftx =
   scon * signal +
    ncon * noise,
  Mesh → False,
  PlotRange → {-1, 1},
  ColorFunction →
   "GrayTones"];
rightnumsample =
 ArrayPlot[
  rightx = ncon * noise,
  Mesh → False,
  PlotRange → {-1, 1},
  ColorFunction →
   "GrayTones"],
leftnumsample =
 ArrayPlot[
  leftx = ncon * noise,
  Mesh → False,
  PlotRange → {-1, 1},
```

```
      ColorFunction →
       "GrayTones"];
  rightnumsample =
   ArrayPlot[
     rightx =
      scon * signal +
       ncon * noise,
     Mesh → False,
     PlotRange → {-1, 1},
     ColorFunction →
      "GrayTones"]];

flash = leftnumsample;
Pause[.25];
flash = blank;
Pause[.25];
flash = rightnumsample;
Pause[.25];
flash = blank;

myanswer =
```

```
ChoiceDialog[
  "Signal on",
  {"First" → 1,
   "Second" → 0},
  WindowSize →
   {300, 80},
  WindowMargins →
   {{Automatic, 0},
    {Automatic,
     330}}];

If[myanswer ==
   whichflash,
  WasICorrect = 1,
  WasICorrect = 0];

idealanswer =
  If[
   Flatten[leftx].
      Flatten[signal] >
    Flatten[rightx].
```

```
          Flatten[signal],
        1, 0];

      If[idealanswer ==
        whichflash,
       WasIdealCorrect = 1,
       WasIdealCorrect = 0];
      data = Append[data,
        {WasICorrect,
         WasIdealCorrect}],

      {numtrials}];
   ]
```

Run a block of trials

In[730]:=
```
twoflashes
NotebookClose[nb];
```

Take a look at the raw data

```
data // TableForm
```

Was I Correct?	Was Ideal
1	1
1	1
1	1
1	1
1	1

Analyze the data

Let's drop the table heading stored in row 1, and then transpose the matrix so that the columns become the rows:

```
In[733]:= data2 = Transpose[
            Drop[data, 1]]

Out[733]= {{1, 1, 1, 1, 1},
           {1, 1, 1, 1, 1}}
```

Let's use a combination of Map[ ]

and Count[ ] (used earlier to make histograms) to count up all occurrences of an event type. So the total for myhits is:

```
  dprime[
   myproportioncorrect];
idealdprime =
  dprime[
   idealproportioncorrect];
mystatisticalefficiency =
  Round[
   100 *
    (mydprime /
       idealdprime) ^ 2];
```

In[739]:=
```
Print[
  Style[
   Grid[
    {{"my prop correct",
      "ideal's prop correct",
      "my d'",
      "ideal's d'",
      "my efficiency (%)"},
     {myproportioncorrect,
      idealproportioncorrect, mydprime,
      idealdprime,
      mystatisticalefficiency}},
    Frame → All], 9]];
```

How can you measure absolute efficiency if the signal-to-noise ratio is so high the ideal observer doesn't make enough mistakes to get a reliable estimate of it's d'? Recall that we don't need to simulate this SKE observer. Its d' is: $\frac{\sqrt{s.s}}{\sigma}$

my prop correct	ideal's prop correct	my d'	ideal's d'	my efficiency (%)
1.	1.	∞	∞	Indeterminate

To get a reasonably reliable estimate, you need at least 100 or more trials, preferably more. And you and the ideal need to make mistakes!

How can you measure absolute efficiency if the signal-to-noise ratio is so high the ideal observer doesn't make enough mistakes to get a reliable estimate of it's d' ? Recall that we don’t need to simulate this SKE observer. Its d’ is: $\frac{\sqrt{s.s}}{\sigma}$

References

Burgess, A. E., Wagner, R. F., Jennings, R. J., & Barlow, H. B. (1981). Efficiency of human visual signal discrimination. Science, 214, 93-94.
De Valois, R. L., Albrecht, D. G., & Thorell, L. G. (1982). Spatial frequency selectivity of cells in macaque visual cortex. *Vision Research*, *22*(5), 545–559.

Kersten, D. (1984). Spatial summation in visual noise. Vision Research, 24,, 1977-1990.
Morgenstern, Y., & Elder, J. H. (2012). Local Visual Energy Mechanisms Revealed by Detection of Global Patterns. *Journal of Neuroscience*, *32*(11), 3679–3696.
Watson, A. B., Barlow, H. B., & Robson, J. G. (1983). What does the eye see best? Nature, 31,, 419-422.