

```

In[1]:= (* Synergy and Antagonism for Colors in Diffusion *)
(* Copyright May 2,
2006 Doug Youvan www.youvan.com www.pseudocolor.com *)

name = Table[i, {i, 1, 100}];

bckrow = .
bckcol = .
newsum = .
kerlsum = .
gsum = .

bckrow = Table[{x, y}, {x, 1, 512}, {y, 1, 512}];
bckcol = Table[{x, y}, {x, 1, 512}, {y, 1, 512}];
newsum = Table[{x, y}, {x, 1, 512}, {y, 1, 512}];
kerlsum = Table[{x, y}, {x, 1, 512}, {y, 1, 512}];

bckrow[[All, All]] = {0., 0., 0.};
bckcol[[All, All]] = {0., 0., 0.};

For[j = 126, j ≤ 130, j++, bckcol[[All, j]] = {1.0, 0.0, 0.0}];
For[j = 254, j ≤ 258, j++, bckcol[[All, j]] = {1.0, 0.0, 0.0}];
For[j = 382, j ≤ 386, j++, bckcol[[All, j]] = {1.0, 0.0, 0.0}];

For[i = 126, i ≤ 130, i++, bckrow[[i, All]] = {0.0, 1.0, 0.0}];
For[i = 254, i ≤ 258, i++, bckrow[[i, All]] = {0.0, 1.0, 0.0}];
For[i = 382, i ≤ 386, i++, bckrow[[i, All]] = {0.0, 1.0, 0.0}];

newsum = bckrow + bckcol;
kerlsum = bckrow + bckcol;

gsum = Graphics[RasterArray[Apply[RGBColor, newsum, {2}]],
  AspectRatio → error, ImageSize → {512, 512}];

Show[gsum];

Export["D:\\Mathematica\\Data\\0.gif", gsum, "GIF", ImageSize → {512, 512}]

kern = {{1/100, 1/100, 1/100, 1/100, 1/100, 1/100, 1/100, 1/100, 1/100, 1/100},
  {1/100, 1/100, 1/100, 1/100, 1/100, 1/100, 1/100, 1/100, 1/100, 1/100},
  {1/100, 1/100, 1/100, 1/100, 1/100, 1/100, 1/100, 1/100, 1/100, 1/100},
  {1/100, 1/100, 1/100, 1/100, 1/100, 1/100, 1/100, 1/100, 1/100, 1/100},
  {1/100, 1/100, 1/100, 1/100, 1/100, 1/100, 1/100, 1/100, 1/100, 1/100},
  {1/100, 1/100, 1/100, 1/100, 1/100, 1/100, 1/100, 1/100, 1/100, 1/100},
  {1/100, 1/100, 1/100, 1/100, 1/100, 1/100, 1/100, 1/100, 1/100, 1/100},
  {1/100, 1/100, 1/100, 1/100, 1/100, 1/100, 1/100, 1/100, 1/100, 1/100},
  {1/100, 1/100, 1/100, 1/100, 1/100, 1/100, 1/100, 1/100, 1/100, 1/100},
  {1/100, 1/100, 1/100, 1/100, 1/100, 1/100, 1/100, 1/100, 1/100, 1/100}};

(* centers at 128, 256, 384; range 64 - 448 ; borders at 192, 320 *)

(* 64,192

```

```

192,320
320,448 *)

(* i=1 bottom row ; j=1 left column *)

For[loop = 1, loop ≤ 15, loop++,

  (* independent by OR - top left*) For[i = 320, i ≤ 448, i++,
  For[j = 50, j ≤ 204, j++, If[(newsum[[i, j, 1]] > 0.3 || newsum[[i, j, 2]] > 0.3),
    (newsum[[i, j, 1]] = newsum[[i, j, 2]] = 0.; newsum[[i, j, 3]] = 1.)]]];

  (* combined by addition - top center*) For[i = 320, i ≤ 448, i++,
  For[j = 192, j ≤ 320, j++, If[(newsum[[i, j, 1]] + newsum[[i, j, 2]] > 0.30),
    (newsum[[i, j, 1]] = newsum[[i, j, 2]] = 0.; newsum[[i, j, 3]] = 1.)]]];

  (* independent by OR; assymetric threshold - top right*) For[i = 320, i ≤ 448, i++,
  For[j = 320, j ≤ 448, j++, If[(newsum[[i, j, 1]] > 0.4 || newsum[[i, j, 2]] > 0.2),
    (newsum[[i, j, 1]] = newsum[[i, j, 2]] = 0.; newsum[[i, j, 3]] = 1.)]]];

  (*antagonism - center left*) For[i = 192, i ≤ 320, i++,
  For[j = 50, j ≤ 204, j++, If[(((newsum[[i, j, 1]] > 0.2) && (newsum[[i, j, 2]] < 0.1)) |
    ((newsum[[i, j, 2]] > 0.2) && (newsum[[i, j, 1]] < 0.1))),
    (newsum[[i, j, 1]] = newsum[[i, j, 2]] = 0.; newsum[[i, j, 3]] = 1.)]]];

  (* more antagonism - center center*)
  For[i = 192, i ≤ 320, i++, For[j = 192, j ≤ 320, j++,
    If[(((newsum[[i, j, 1]] > 0.2) && (newsum[[i, j, 2]] < 0.01)) ||
      ((newsum[[i, j, 2]] > 0.2) && (newsum[[i, j, 1]] < 0.01))),
      (newsum[[i, j, 1]] = newsum[[i, j, 2]] = 0.; newsum[[i, j, 3]] = 1.)]]];

  (* asymmetric antagonism - right center*)
  For[i = 192, i ≤ 320, i++, For[j = 320, j ≤ 448, j++,
    If[(((newsum[[i, j, 1]] > 0.2) && (newsum[[i, j, 2]] < 0.001)) ||
      ((newsum[[i, j, 2]] > 0.2) && (newsum[[i, j, 1]] < 0.1))),
      (newsum[[i, j, 1]] = newsum[[i, j, 2]] = 0.; newsum[[i, j, 3]] = 1.)]]];

  (* synergy - bottom left *) For[i = 64, i ≤ 192, i++,
  For[j = 64, j ≤ 192, j++,
    If[(newsum[[i, j, 1]] * newsum[[i, j, 2]] > 0.01),
      (newsum[[i, j, 1]] = newsum[[i, j, 2]] = 0.; newsum[[i, j, 3]] = 1.)]]];

  (*synergy with threshold - bottom center *) For[i = 64, i ≤ 192, i++,
  For[j = 192, j ≤ 320, j++, If[(0.01 > newsum[[i, j, 1]] * newsum[[i, j, 2]] > 0.0001),
    (newsum[[i, j, 1]] = newsum[[i, j, 2]] = 0.; newsum[[i, j, 3]] = 1.)]]];

  (* asymmetric synergy - bottom right *) For[i = 64, i ≤ 192, i++,
  For[j = 320, j ≤ 448, j++,
    If[(newsum[[i, j, 1]] * (newsum[[i, j, 2]])^2 > 0.001),
      (newsum[[i, j, 1]] = newsum[[i, j, 2]] = 0.; newsum[[i, j, 3]] = 1.)]]];

  For[i = 1, i ≤ 10, i++, newsum[[i, All]] = {0.1, 0.1, 0.1}];
  For[i = 503, i ≤ 512, i++, newsum[[i, All]] = {0.1, 0.1, 0.1}];
  For[j = 1, j ≤ 10, j++, newsum[[All, j]] = {0.1, 0.1, 0.1}];
  For[j = 503, j ≤ 512, j++, newsum[[All, j]] = {0.1, 0.1, 0.1}];

```

```
gsum = Graphics[RasterArray[Apply[RGBColor, newsum, {2}]],
  AspectRatio -> error, ImageSize -> {512, 512}];
Show[gsum];

Export["D:\\Mathematica\\Data\\" <> ToString[name[[loop]]] <> ".gif",
  gsum, "GIF", ImageSize -> {512, 512}];

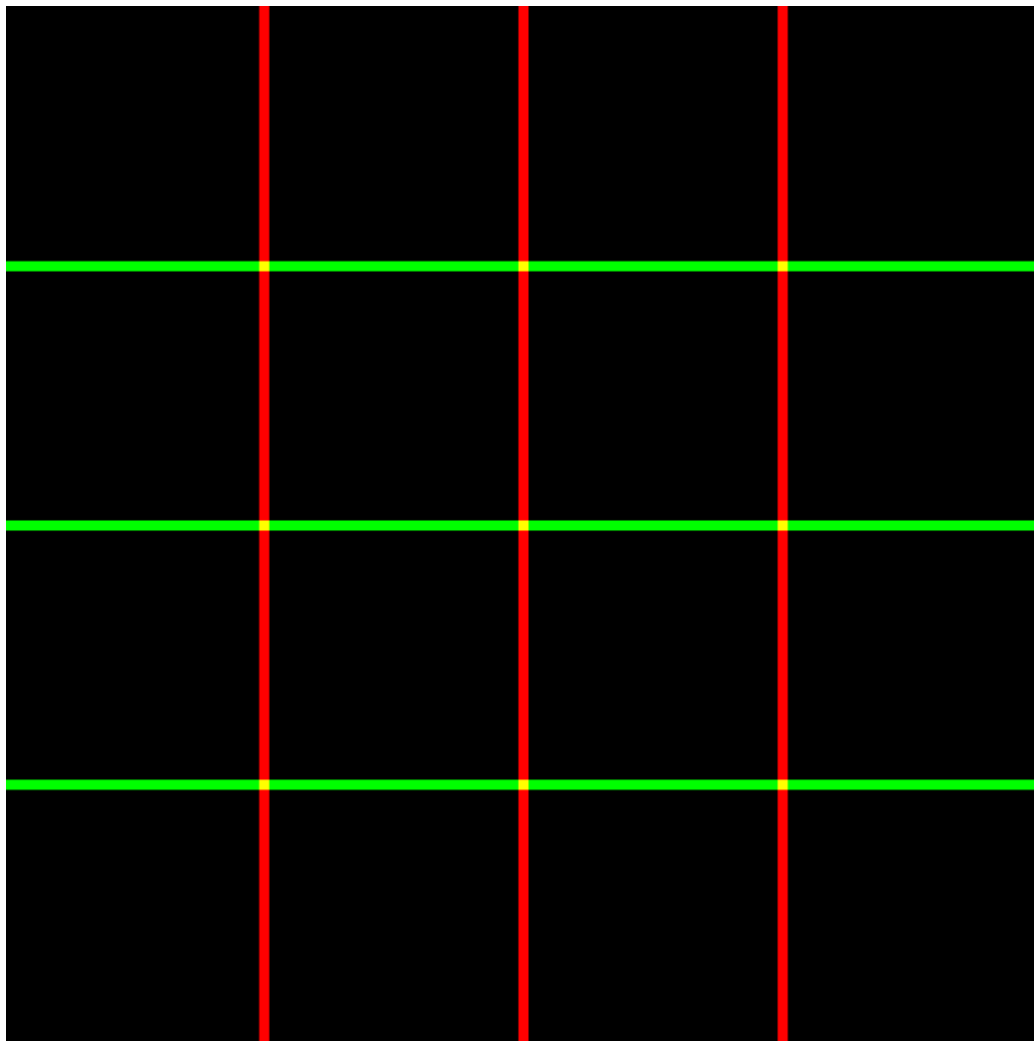
newsum = ker1sum;

sumr = 0.;
sumg = 0.;
sumb = 0.;
sumr = ListConvolve[kern, newsum[[All, All, 1]]];
sumg = ListConvolve[kern, newsum[[All, All, 2]]];
sumb = ListConvolve[kern, newsum[[All, All, 3]]]; (* place holder *)
For[i = 1, i <= 503, i++, For[j = 1, j <= 503, j++,
  newsum[[i, j]] = {sumr[[i, j]], sumg[[i, j]], sumb[[i, j]]}];

For[i = 1, i <= 503, i++,
  For[j = 1, j <= 503, j++, ker1sum[[i + 4, j + 4]] = newsum[[i, j]]];

newsum = ker1sum;
]

AspectRatio::aspr :
Value of option AspectRatio -> error is not a finite positive number or Automatic. More...
```



```
AspectRatio::aspr :  
  Value of option AspectRatio -> error is not a finite positive number or Automatic. More...
```

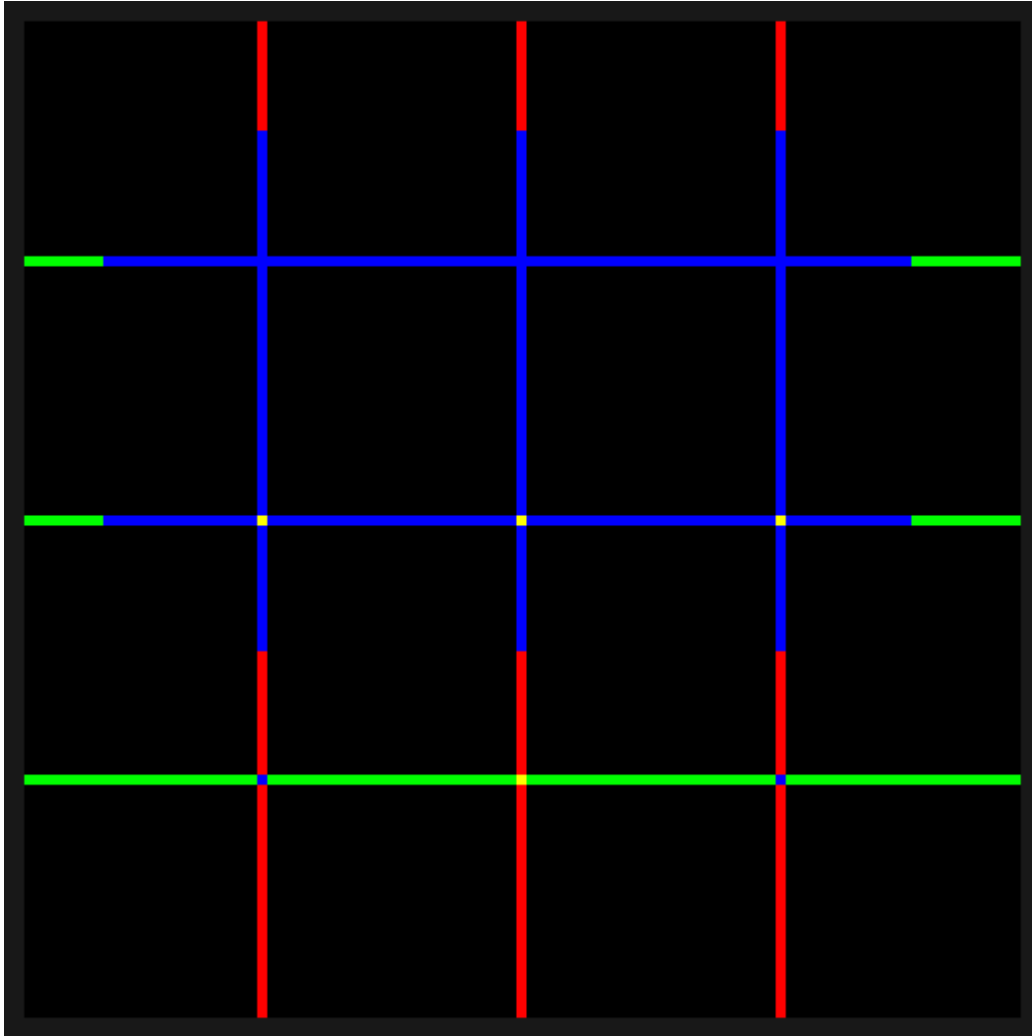
```
AspectRatio::aspr :  
  Value of option AspectRatio -> error is not a finite positive number or Automatic. More...
```

```
Out[23]= D:\Mathematica\Data\0.gif
```

```
General::spell1 :  
  Possible spelling error: new symbol name "sumg" is similar to existing symbol "sumr". More...
```

```
General::spell :  
  Possible spelling error: new symbol name "sumb" is similar to existing symbols {sumg, sumr}. More...
```

```
AspectRatio::aspr :  
  Value of option AspectRatio -> error is not a finite positive number or Automatic. More...
```



```
AspectRatio::aspr :  
  Value of option AspectRatio -> error is not a finite positive number or Automatic. More...
```

```
AspectRatio::aspr :  
  Value of option AspectRatio -> error is not a finite positive number or Automatic. More...
```

```
General::stop : Further output of AspectRatio::aspr will be suppressed during this calculation. More...
```

