

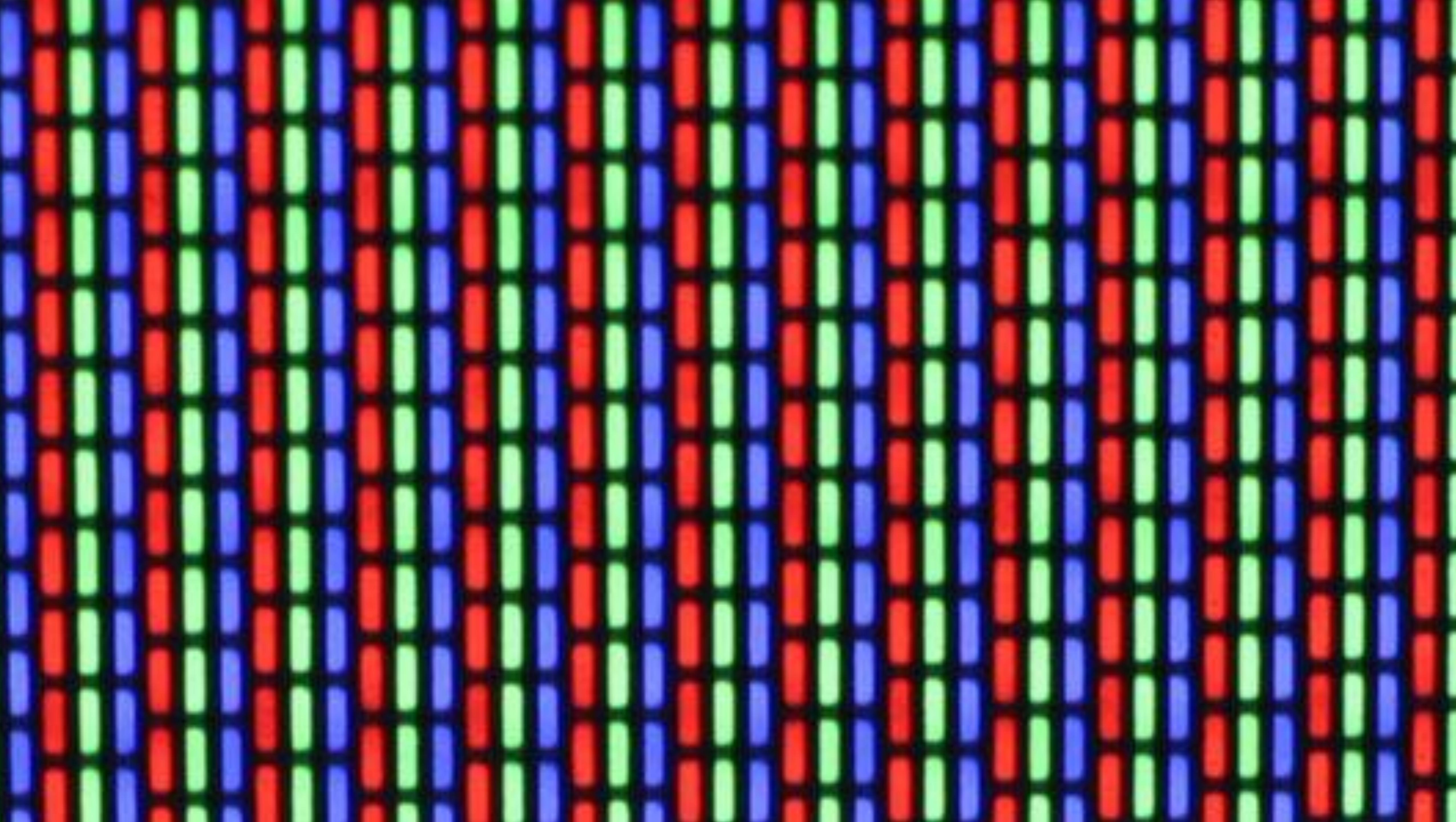
RASTER GRAPHICS

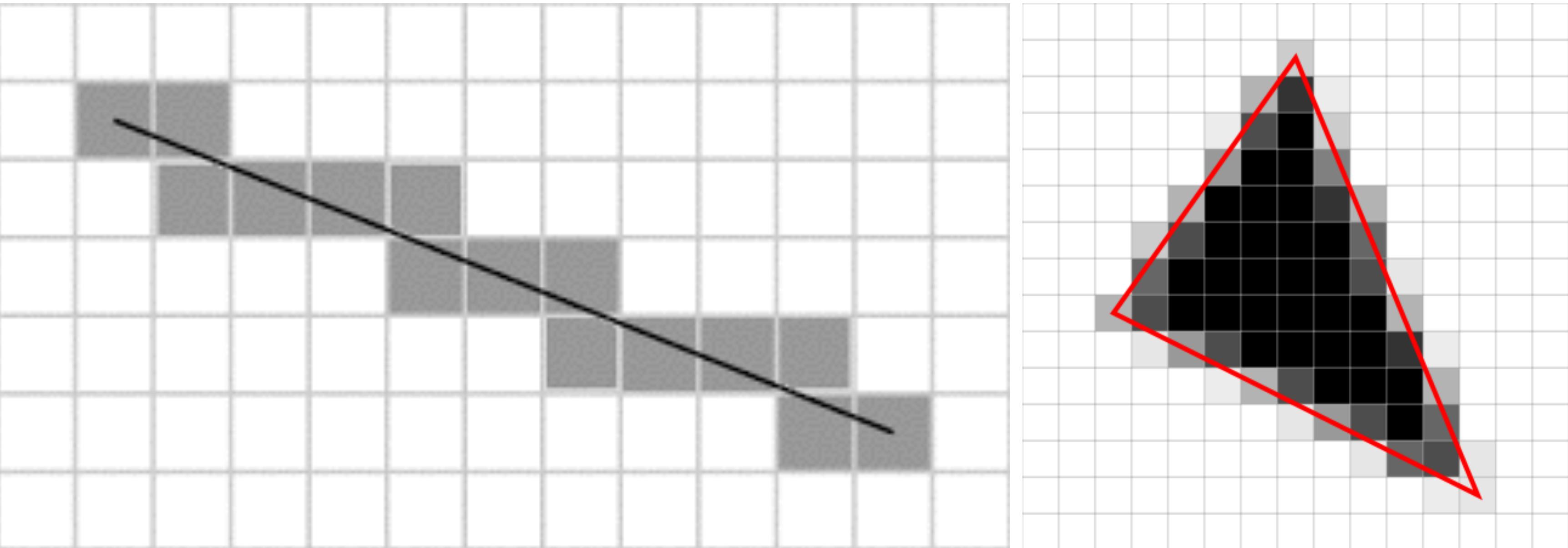
Lior Ben-Gai
January 2020

-  SoundsOfEarth IS71016B PfA
h (Lior) (...321 (1))
-  Screen Shot 2017-1...41.35 VOE_exports
-  Stereo
-  140524_001.W
AV.gpk
-  Introduction01
-  UM2_drawing
Machi....gcode
-  backpack2.fxg
-  Screen Shot
2017-1...40.55
-  140524_001.W
AV
-  outline_mk3
-  ImageArchive
-  Macintosh HD



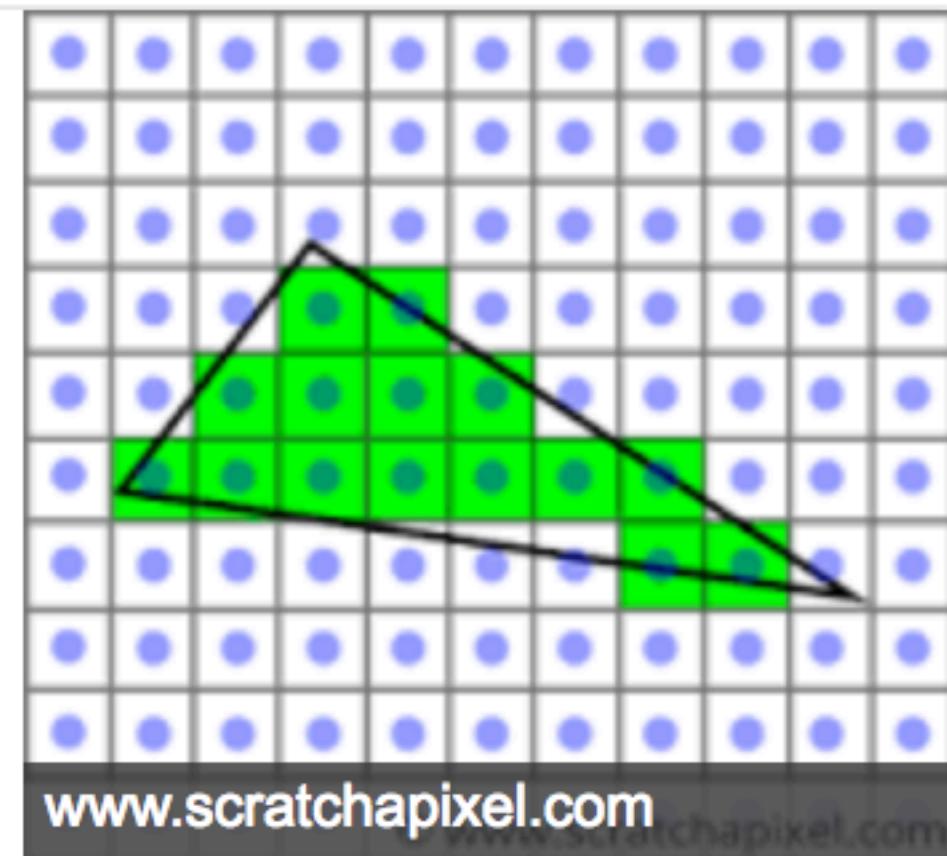




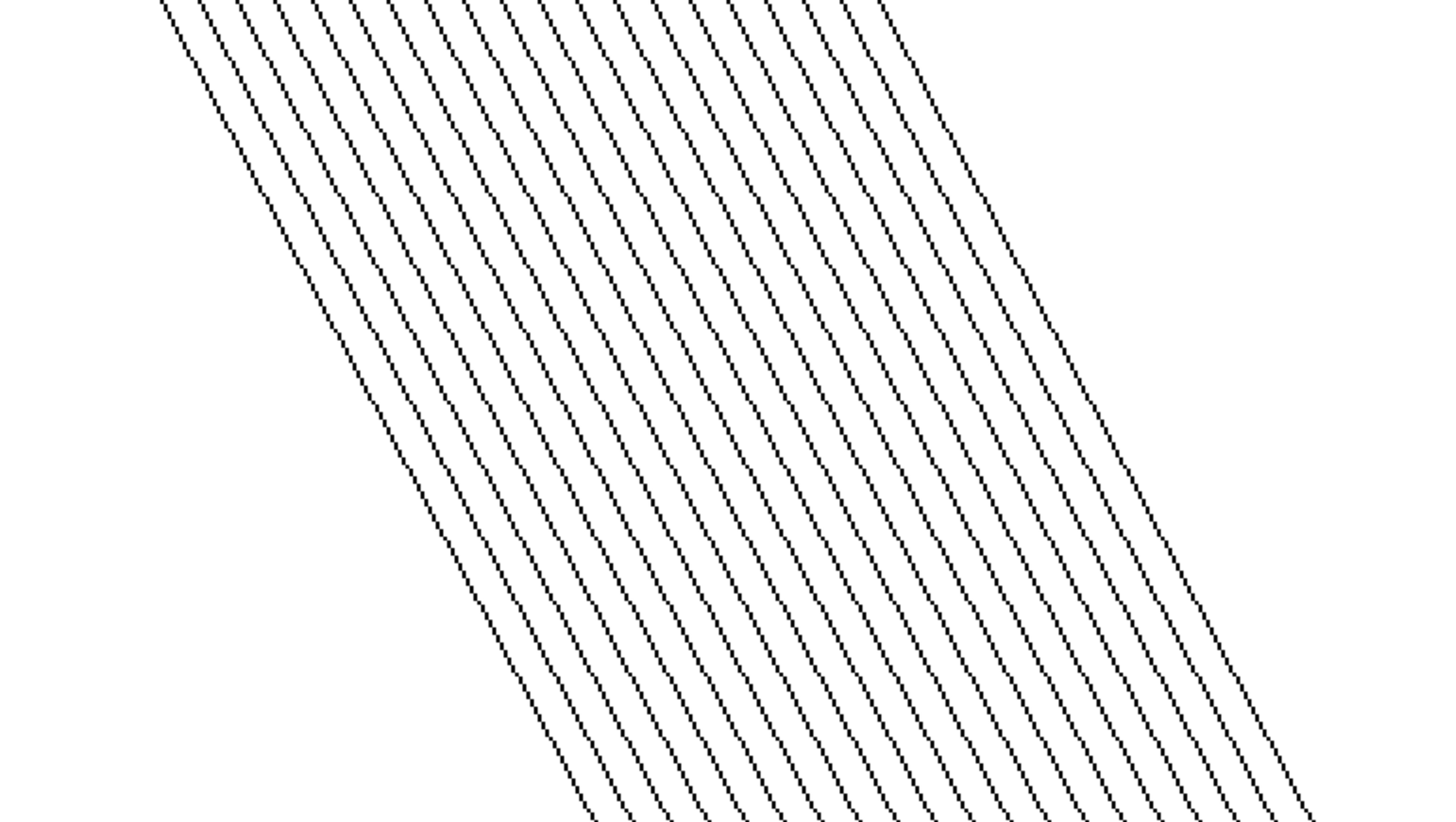


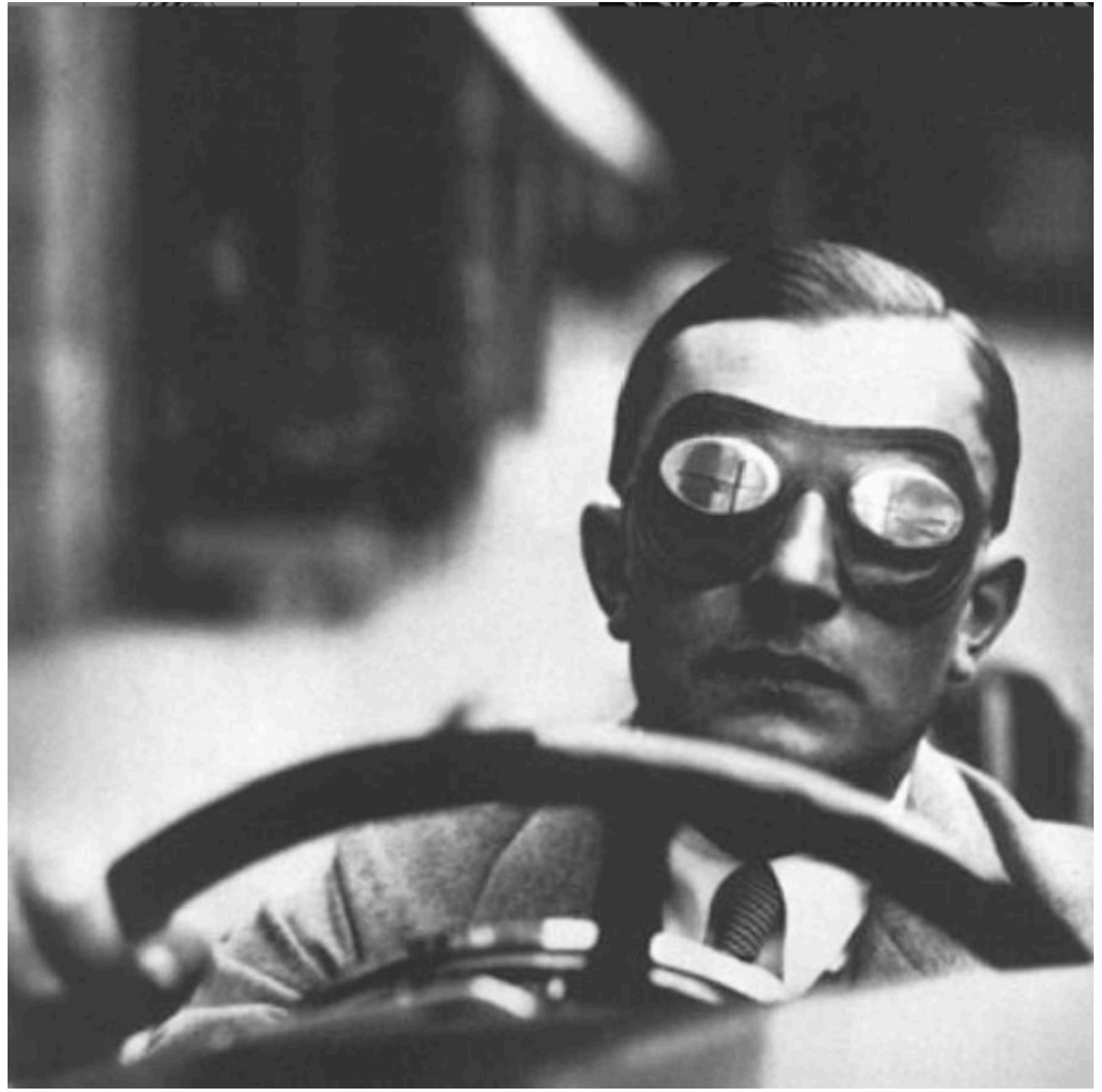
Rasterisation (or **rasterization**) is the task of taking an image described in a vector graphics format (shapes) and converting it into a raster image (pixels or dots) for output on a video display or printer, or for storage in a bitmap file format.

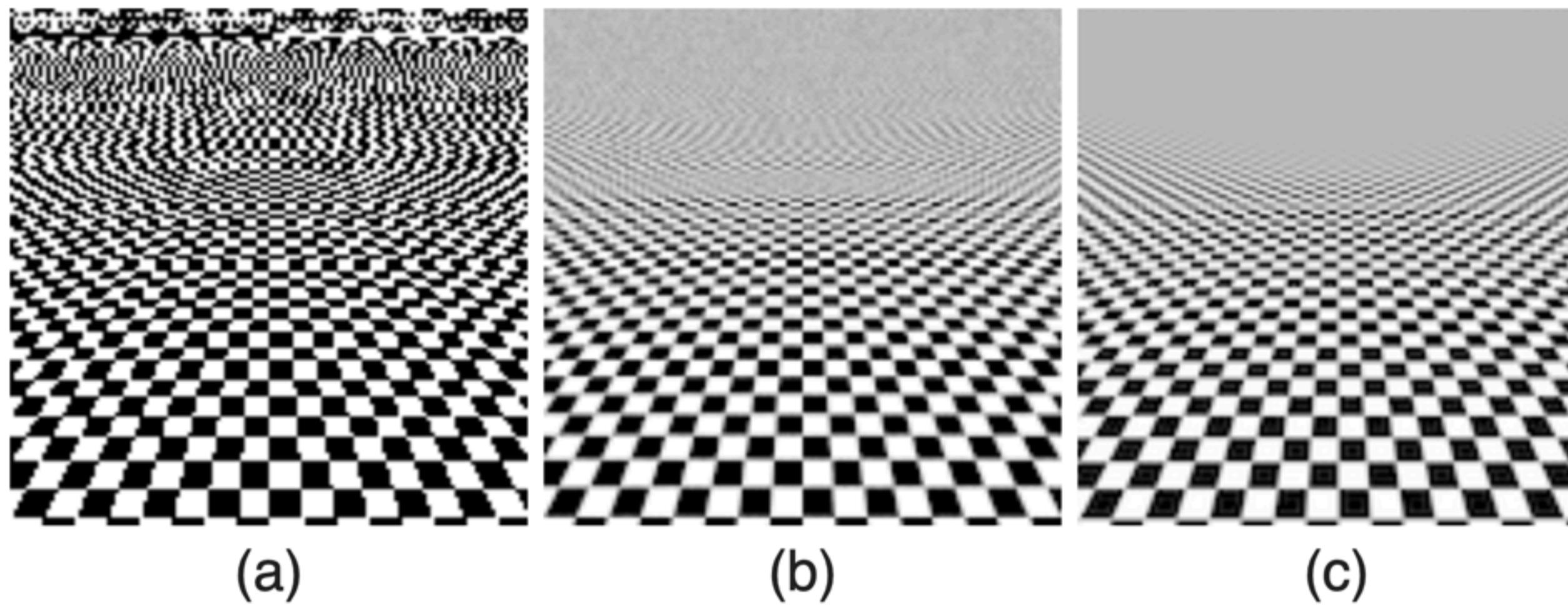
Rasterisation - Wikipedia
<https://en.wikipedia.org/wiki/Rasterisation>



<https://en.wikipedia.org/wiki/Rasterisation>







Spatial anti-aliasing

From Wikipedia, the free encyclopedia

In [digital signal processing](#), **spatial anti-aliasing** is a technique for minimizing the distortion artifacts known as [aliasing](#) when representing a high-resolution image at a lower resolution. [Anti-aliasing](#) is used in [digital photography](#), [computer graphics](#), [digital audio](#), and many other applications.



8.9M



68.34K

https://en.wikipedia.org/wiki/Image_compression

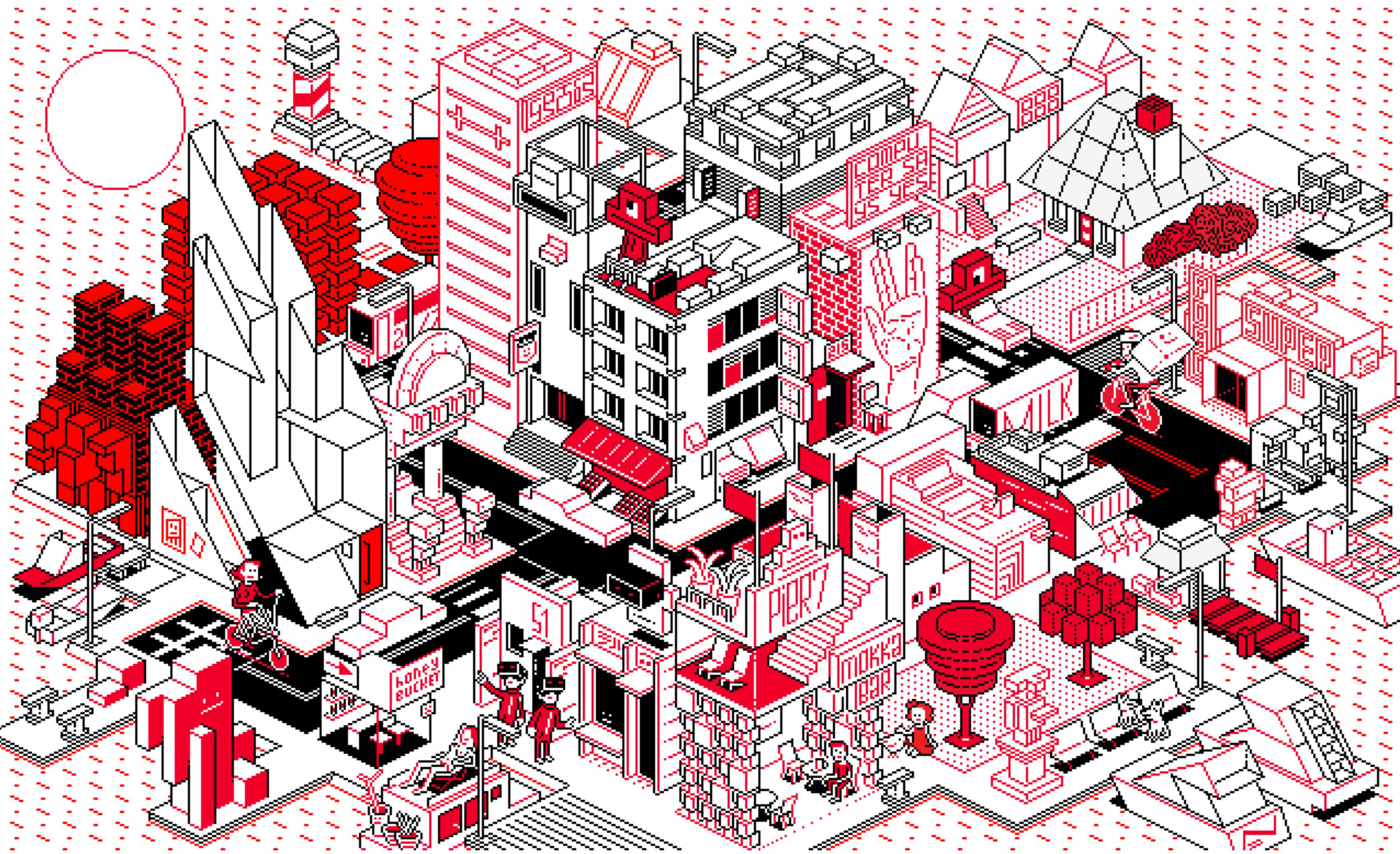
MARIO . . 4060 TOP . 12000



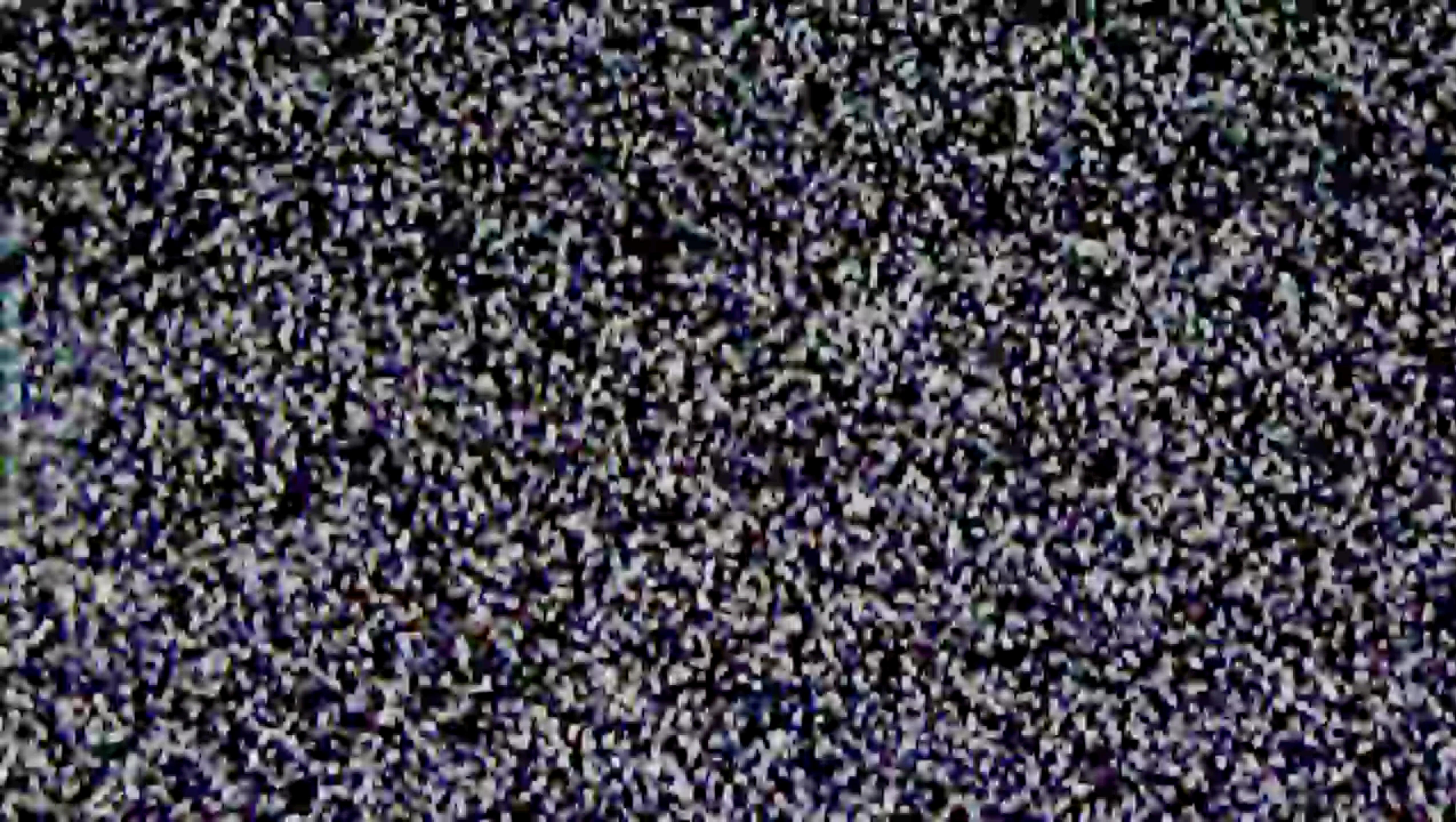
800

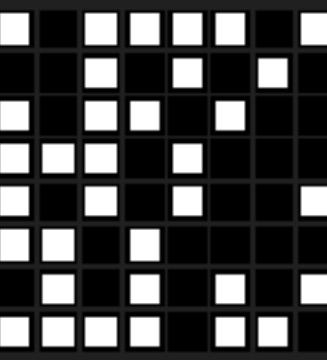
A Brief History of Computer Games





<http://hello.eboy.com/eboy/>





A GRID

```
var sz = 10;
for(var i = 0; i < cols; i ++){
    for(var j = 0; j < rows; j ++){
        fill( random(255) );
        rect(i*sz, j*sz , sz, sz);
    }
}
```

A DATA STRUCTURE FOR A GRID

```
var colorData = [];
```

```
for(var i = 0; i < rows; i ++){  
    for(var j = 0; j < cols; j ++){  
        colorData.push(random(255));  
    }  
}
```

1D ARRAY

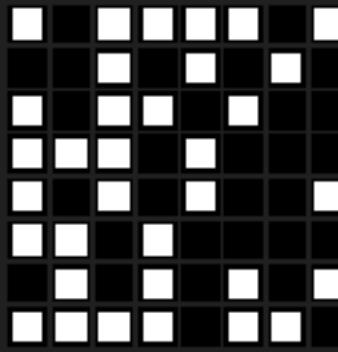
A DATA STRUCTURE FOR A GRID

```
var colorData = [];
```

```
for(var i = 0; i < rows; i ++){  
  
    var tempArr = [];  
    for(var j = 0; j < cols; j ++){  
        tempArr.push( random(255) );  
    }  
    colorData.push(tempArr);  
  
}
```

2D ARRAY

THE PIXELS[] ARRAY



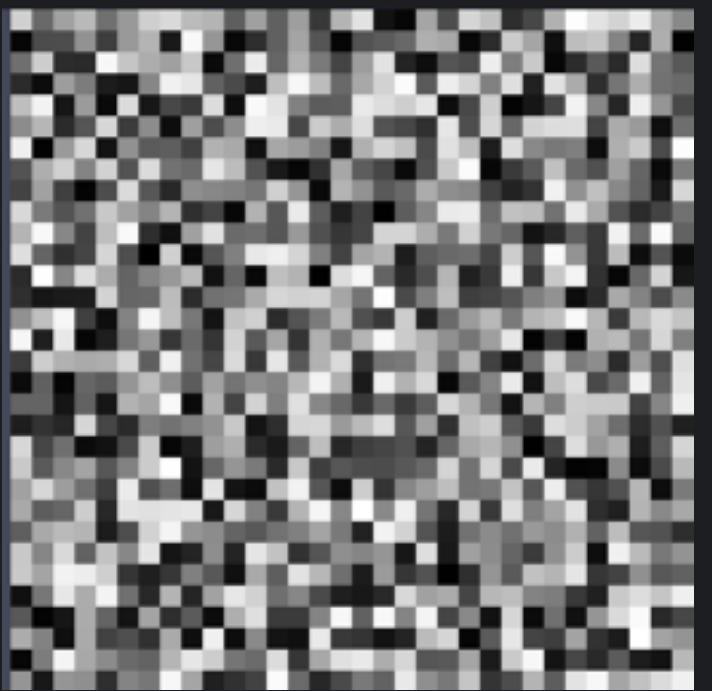
<https://p5js.org/reference/#/p5/pixels>

`pixels.length == width * height * 4`

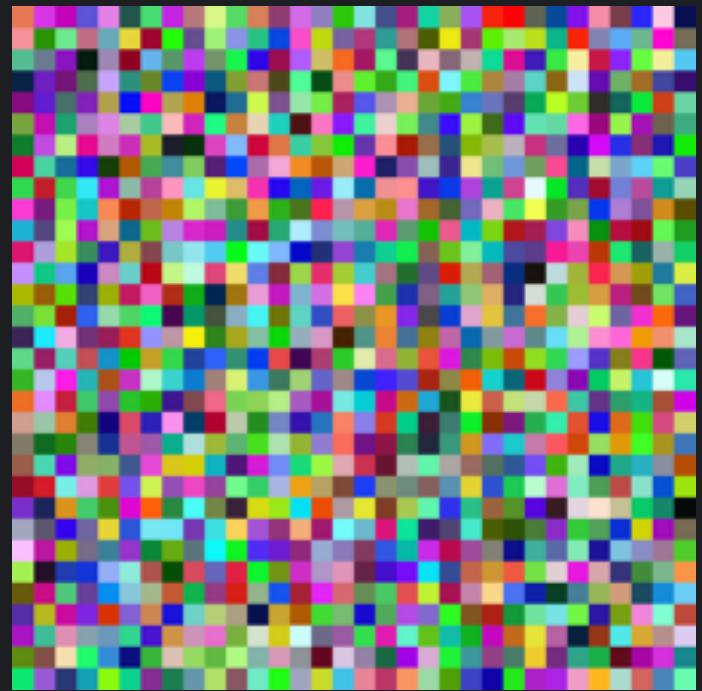


<https://p5js.org/reference/#/p5/pixels>

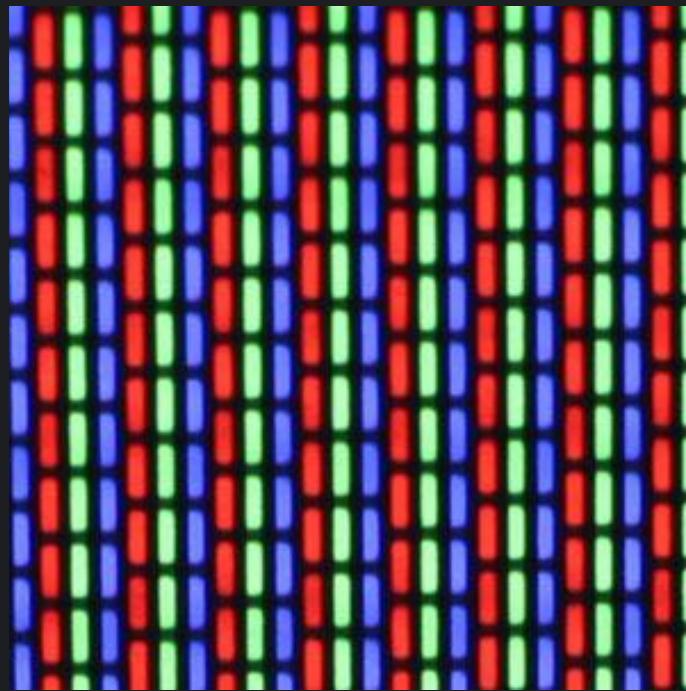
A



B



C



```
var A = [ 0, 0, 0, ... ]
```

```
var B = [ [ 0, 0, 0, 0 ], [ 0, 0, 0, 0 ], [ 0, 0, 0, 0 ], ... ]
```

```
var C = [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ... ]
```

<https://p5js.org/reference/#/p5/pixels>

1D LOOP - CONVERT TO 2D

```
for(var i = 0 ; i < pixels.length; i+= 4){  
    var y = (i/4) / width;  
    var x = (i/4) % width;  
}
```

2D LOOP - CONVERT TO 1D

```
for(var i = 0 ; i < width * 4; i+=4){  
    for(var j = 0 ; j < height * 4; j+=4){  
        var idx = (j * width) + i;  
    }  
}
```

DRAWING IMAGES

```
image(myImg, 0, 0);
```

<https://p5js.org/reference/#/p5.Image>

LOADING IMAGES

```
var myImg;  
  
function preload(){  
    myImg = loadImage('...')  
}  
  
function setup(){  
    image(myImg, 0, 0);  
}
```

<https://p5js.org/reference/#/p5/loadImage>

CREATING IMAGES



The screenshot shows a p5.js code editor interface. On the left, there is a preview window displaying a small teal square. The main area contains the following JavaScript code:

```
let img = createImage(66, 66);
img.loadPixels();
for (let i = 0; i < img.width; i++) {
  for (let j = 0; j < img.height; j++) {
    img.set(i, j, color(0, 90, 102));
  }
}
img.updatePixels();
image(img, 17, 17);
```

At the top right of the editor, there are three buttons: "edit", "reset", and "copy".

*the canvas itself is just an image

<https://p5js.org/reference/#/p5/createImage>

<https://p5js.org/reference/#/p5/createGraphics>

HOMEWORK #11



Glitch art uses the properties of the digital medium as an aesthetic

For next week - please 'GLITCHIFY' the study from this week in a way that makes sense conceptually and visually