

```
/**
  This is the most out there example of what we can do with images.

  For this example, I don't ever actually display the image, instead,
  I use the color values to drive the drawing process. I split the
  canvas into a grid and for each cell, I calculate an equivalent
  pixel on the image. I grab the pixel there and use the brightness
  to map to the size of an ellipse and the color for the fill. Changing
  the size of the grid cells changes how abstract the final image looks.
```

```
C. Andrews
2014-01-16
**/
```

```
PImage img;
int cellSize = 6; // this determines how big the individual cells are
void setup() {
  img = loadImage("cole2.jpg");
  size(img.width * 2, img.height*2);
  background(255);
  noLoop();
  noStroke();
}

void draw() {
  // load the image pixels into the pixels array
  img.loadPixels();

  // iterate over the canvas -- x and y are real coordinates on the canvas
  for (int y = cellSize / 2; y < height - cellSize / 2; y+=cellSize) {
    for (int x = cellSize / 2; x < width - cellSize / 2; x+=cellSize) {
      // compute an equivalent i,j on the image
      // x/width gives us the position of x on the canvas as a percentage
      // multiplying this by img.width converts it back to a position, but in a different
      space
      int i = x*img.width / width;
      // do the same for y -> j
      int j = y*img.height / height;
      // compute the index of the point (i,j)
      int srcIndex = i + j * img.width;
      color c = img.pixels[srcIndex];
      // map the brightness to a value between cellSize and 3
      // brighter values will be smaller
      float diameter = map(brightness(c), 0, 255, cellSize, 3);

      fill(c);
      // draw the ellipse
      ellipse(x, y, diameter, diameter);
    }
  }
}
```