A visual method for generating the separating isosurface of two classes of objects in two-dimensional space using Marching Squares and OpenCV

Shawn Halayka*

Friday 22^{nd} April, 2022 17:59

Abstract

With regard to the separating isosurface of two classes of objects in two-dimensional space, the transition from nonlinear to linear is documented.

1 Introduction

Iron out the wrinkles by downsizing. Can also convolve the images, like blurring, which can be used to melt away higher-frequency detail.

*Independent – sjhalayka@gmail.com

References

[1] James, et al. An Introduction to Statistical Learning with Applications in R. ISBN: 978-1-0716-1417-4

```
vector<float> opencv_downsize(const vector<float>& image, size_t res, size_t target_res)
{
    Mat m = Mat(static_cast<int>(res), static_cast<int>(res), CV_32FC1);
    memcpy(m.data, image.data(), image.size() * sizeof(float));

    float x = static_cast<float>(target_res) / res;

    resize(m, m, cv::Size(), x, x, INTER_LINEAR);
    vector<float> temp_image(target_res * target_res);
    memcpy(&temp_image[0], m.data, temp_image.size() * sizeof(float));

    return temp_image;
}
```

Figure 1: Code to downsize an image, using OpenCV.

```
vector<float> opencv_blur(const vector<float>& image, const size_t num_iterations)
{
    Mat m = Mat(marching_squares_resolution, marching_squares_resolution, CV_32FC1);
    memcpy(m.data, image.data(), image.size() * sizeof(float));

    for (size_t i = 0; i < num_iterations; i++)
        GaussianBlur(m, m, Size(25, 25), 1, 1);

    vector<float> temp_image = image;
    memcpy(&temp_image[0], m.data, image.size() * sizeof(float));

    return temp_image;
}
```

Figure 2: Code to blur an image, using OpenCV.

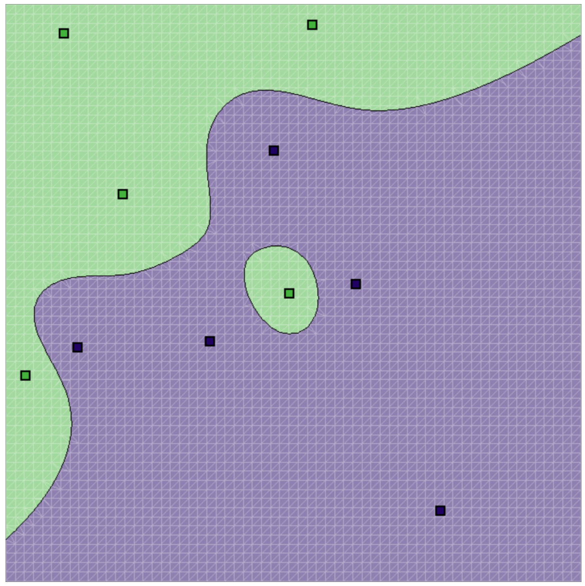

Figure 3: Nonlinear, radial, separation. Grid resolution is 64.

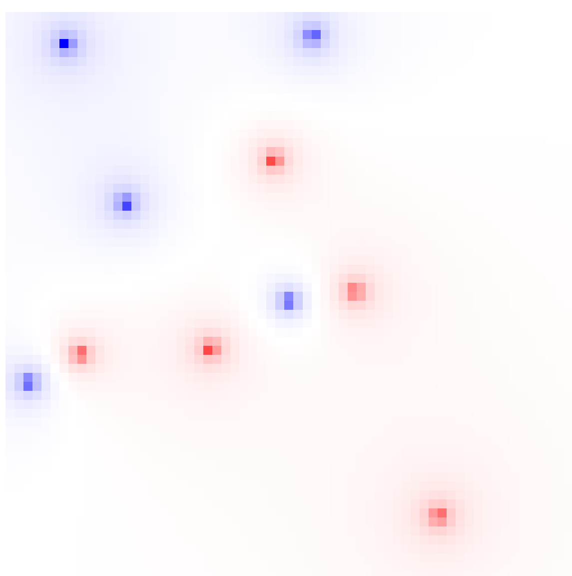

Figure 4: Bitmap image used as input to the Marching Squares algorithm. Image size is 64x64.

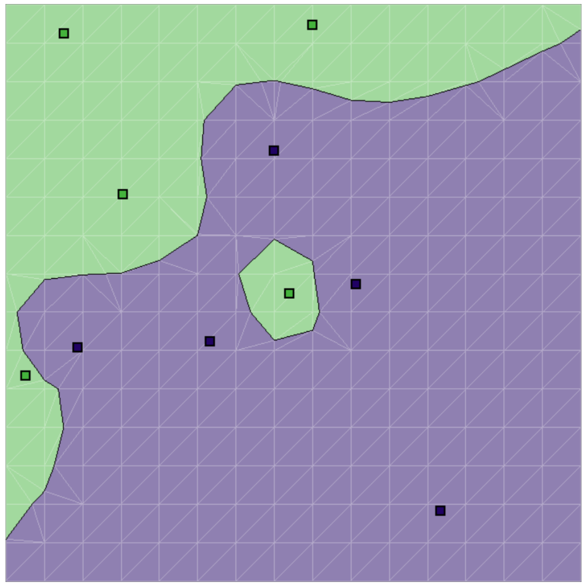

Figure 5: Nonlinear, radial, separation. Grid resolution is 16.

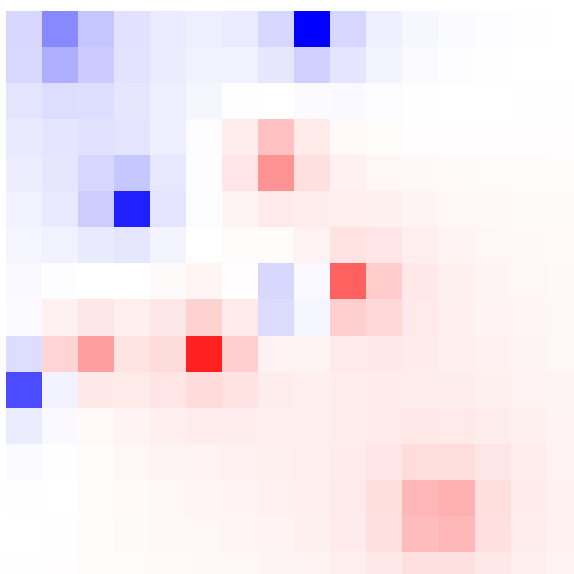

Figure 6: Bitmap image used as input to the Marching Squares algorithm. Image size is 16x16.

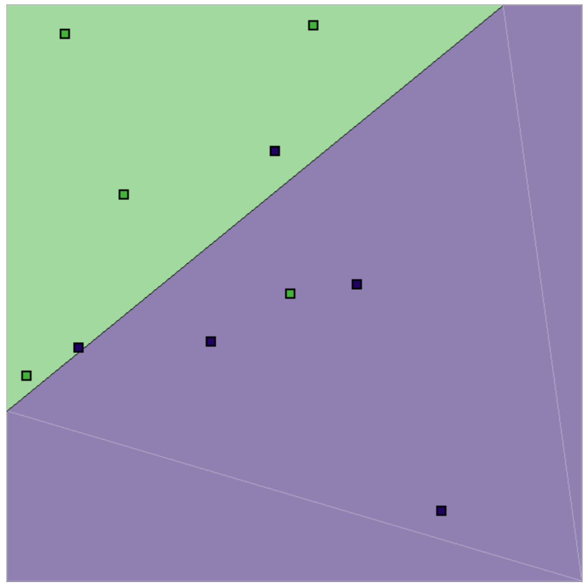

Figure 7: Linear separation. Grid resolution is 2.

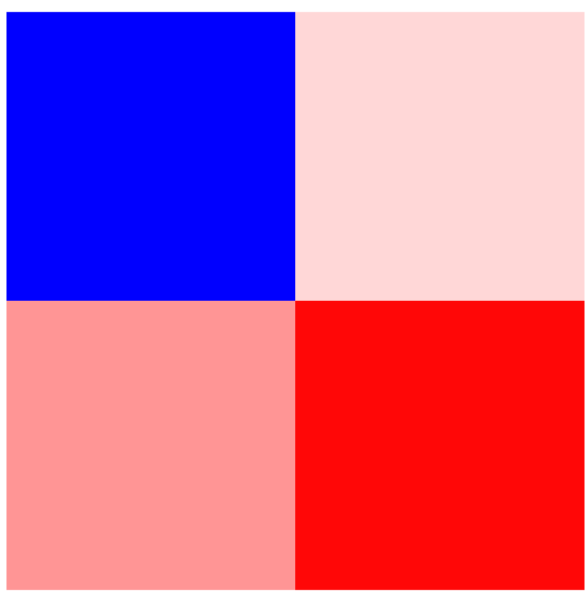

Figure 8: Bitmap image used as input to the Marching Squares algorithm. Image size is 2x2.

Figure 9: Bitmap image used as input to the Marching Squares algorithm.

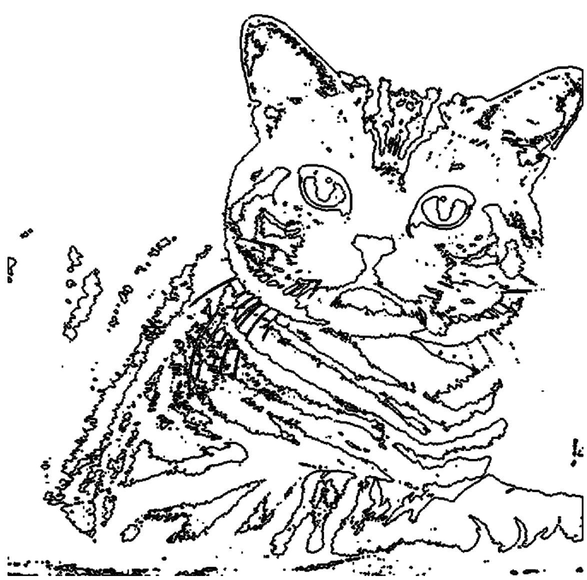

Figure 10: Nonlinear, radial, separation.

Figure 11: Blurred bitmap image used as input to the Marching Squares algorithm.

Figure 12: Nonlinear, radial, separation. Note that the blurring of the bitmap image melts away the higher-resolution detail.