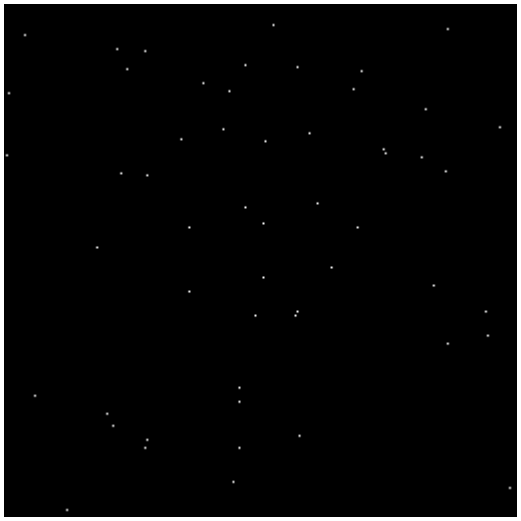
```
begin
    import Pkg;
    Pkg.add("StaticArrays")
    Pkg.add("FileIO")
    using Images, FileIO
end
```

```
  Updating registry at `C:\Users\ItsYe\.julia\registries\General.toml`
 Resolving package versions...
No Changes to `C:\Users\ItsYe\.julia\environments\v1.8\Project.toml`
No Changes to `C:\Users\ItsYe\.julia\environments\v1.8\Manifest.toml`
 Resolving package versions...
No Changes to `C:\Users\ItsYe\.julia\environments\v1.8\Project.toml`
No Changes to `C:\Users\ItsYe\.julia\environments\v1.8\Manifest.toml`
```

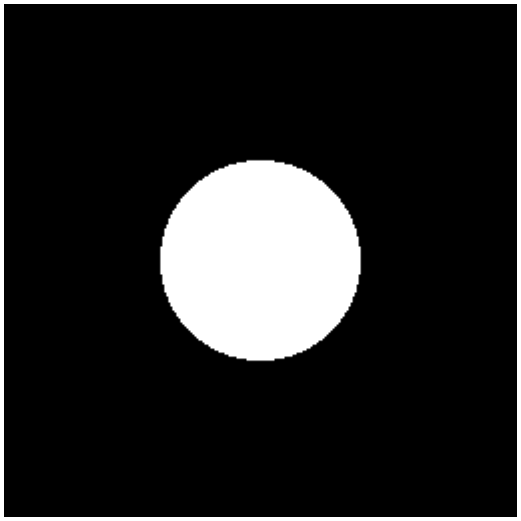
```
begin
    # my code
    include("../SDF/SerialSDF.jl");
    include("../SDF/SDFVis.jl");
    using .SerialSDF
    using .SDFVis
end
```

```
begin
    # Generated Images
    # create just a couple of random
    dots = rand(Float64,256,256) .< 0.001
    # circle shape
    circle = zeros(Bool, 256,256)

    for x in 1:256
        for y in 1:256
            dist = sqrt((x-128.5)^2 + (y-128.5)^2)
            circle[x,y] = dist ≤ 50
        end
    end
end
```



```
begin
    using Colors
    Gray.(dots)
end
```



```
Gray.(circle)
```

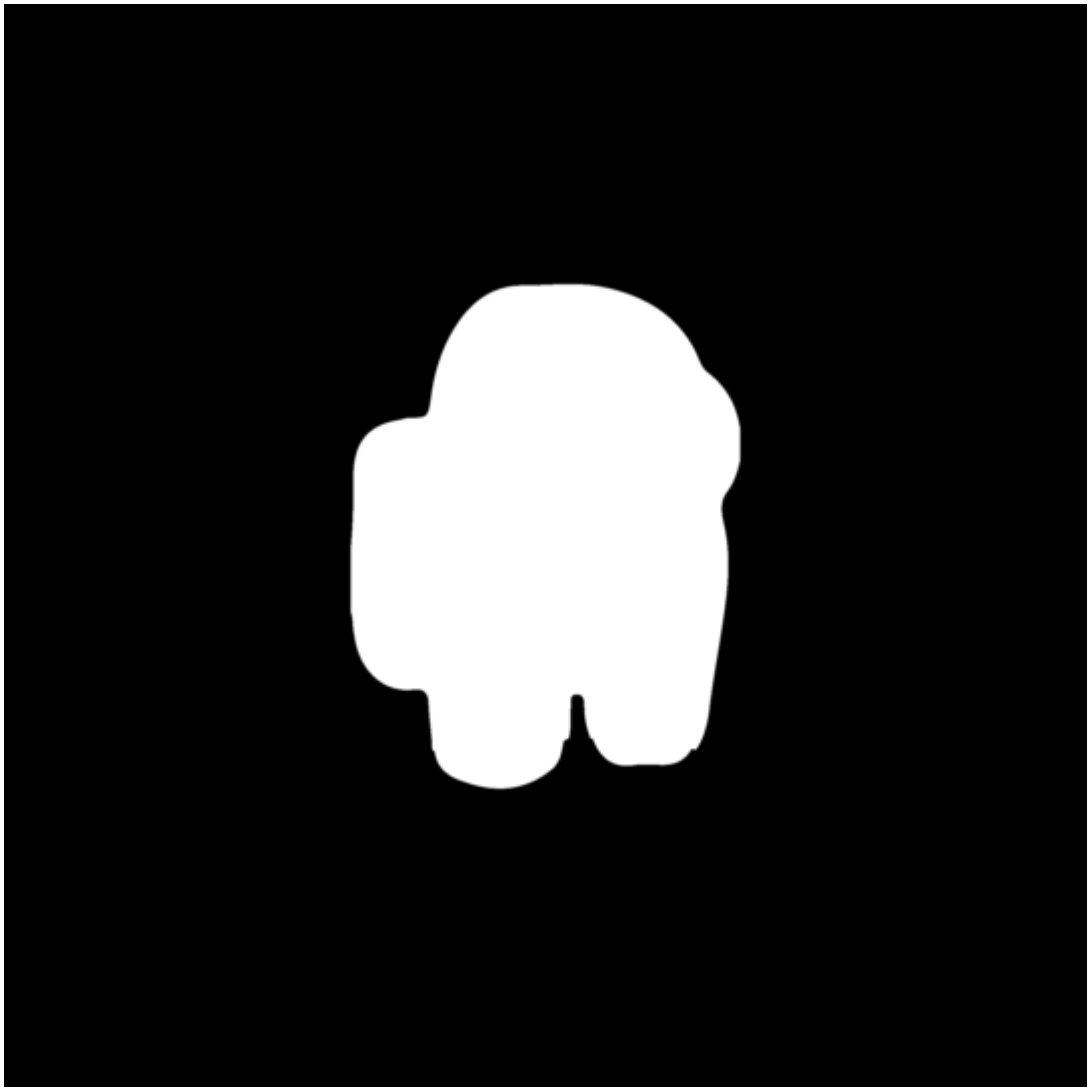
petty =


```
petty = load("./Images/43.png")
```

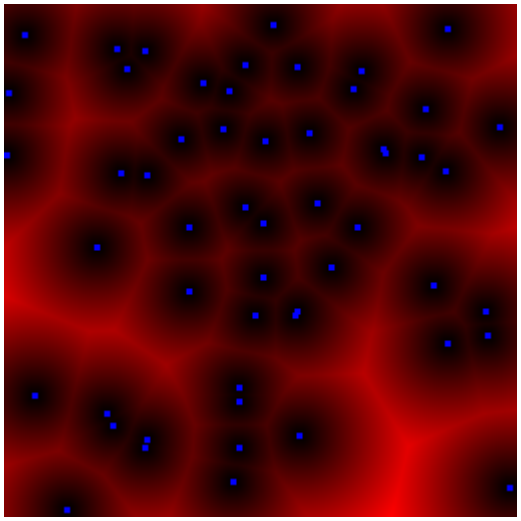
amogus =

```
amogus = load("./Images/amogus.png")
```

```
begin
    petty_threshold = SerialSDF.threshold(petty)
    Gray.(petty_threshold)
end
```

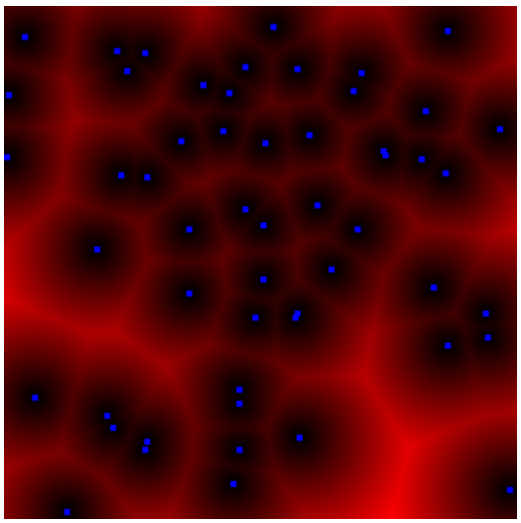


```
begin
    amogus_threshold = SerialSDF.threshold(amogus, 0.5, ColorTypes.alpha)
    Gray.(amogus_threshold)
end
```



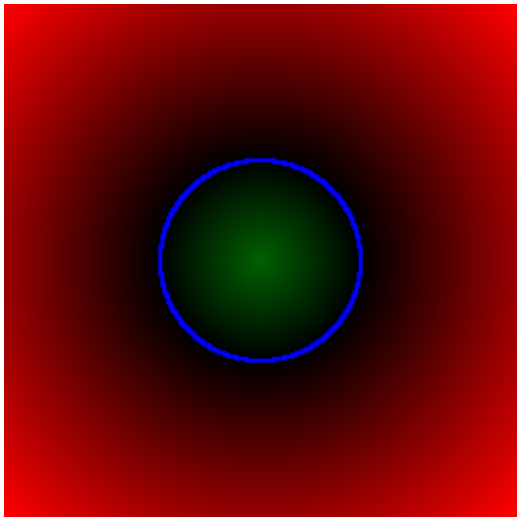
```
begin
    @time dotsSDFBrute = SerialSDF.bruteSDF2D(dots)
    SDFVis.toImageSDF(dotsSDFBrute)
end
```

```
  9.466325 seconds (57.33 k allocations: 3.276 MiB, 0.63% compilation time)
```



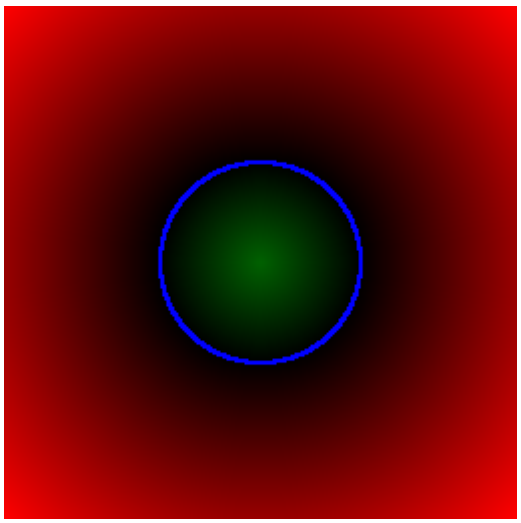
```
begin
    @time dotsSDFDijkstra = SerialSDF.dijkstraSDF2D(dots)
    SDFVis.toImageSDF(dotsSDFDijkstra)
end
```

```
  1.123197 seconds (1.66 M allocations: 87.368 MiB, 4.50% gc time, 87.92% co
mpilation time)
```



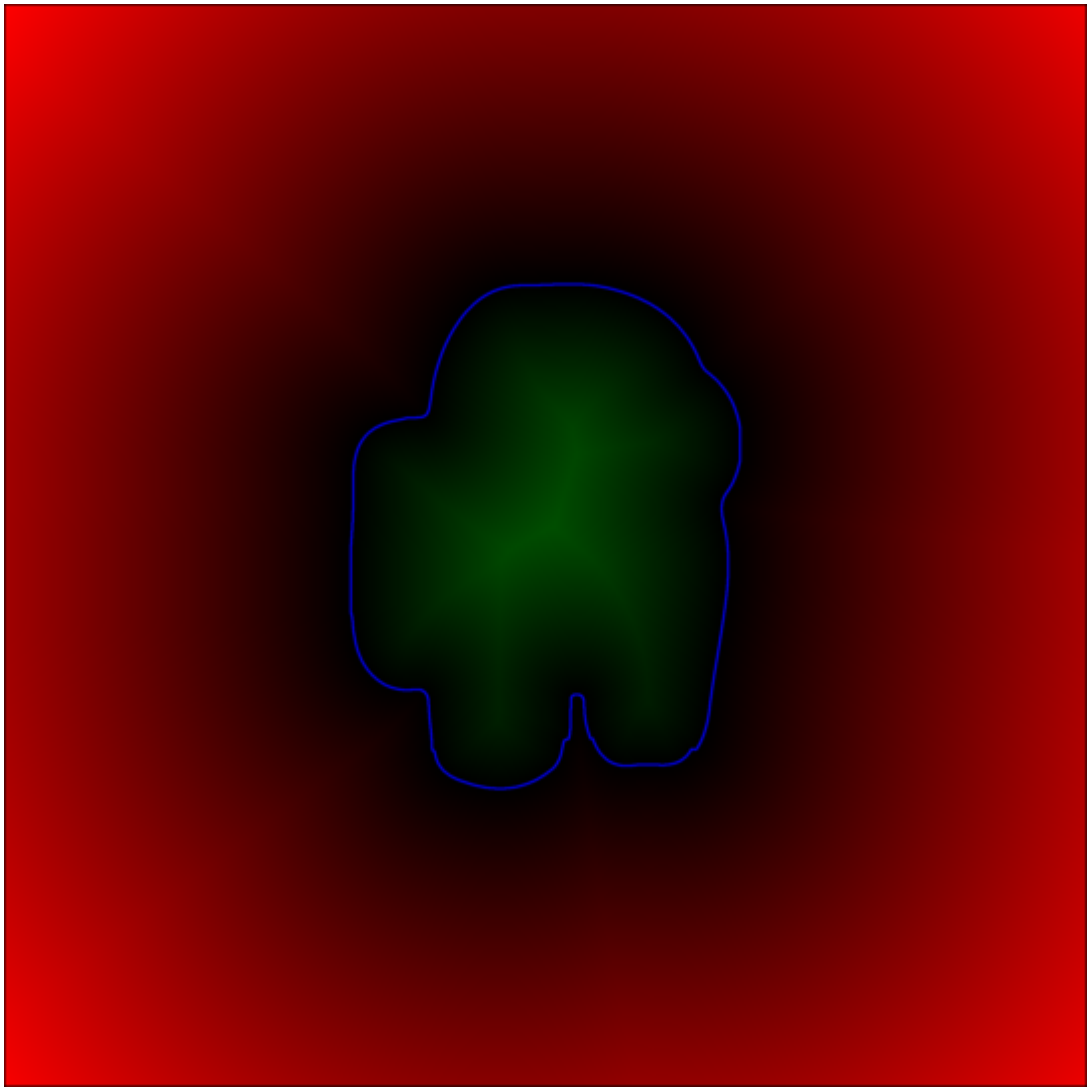
```
begin
    @time circleSDFBrute = SerialSDF.bruteSDF2D(circle)
    SDFVis.toImageSDF(circleSDFBrute)
end
```

```
  5.635181 seconds (39.24 k allocations: 2.540 MiB, 0.77% compilation time)
```



```
begin
    @time circleSDFDijkstra = SerialSDF.dijkstraSDF2D(circle)
    SDFVis.toImageSDF(circleSDFDijkstra)
end
```

```
  0.431620 seconds (602.56 k allocations: 35.754 MiB, 71.18% compilation tim
e)
```

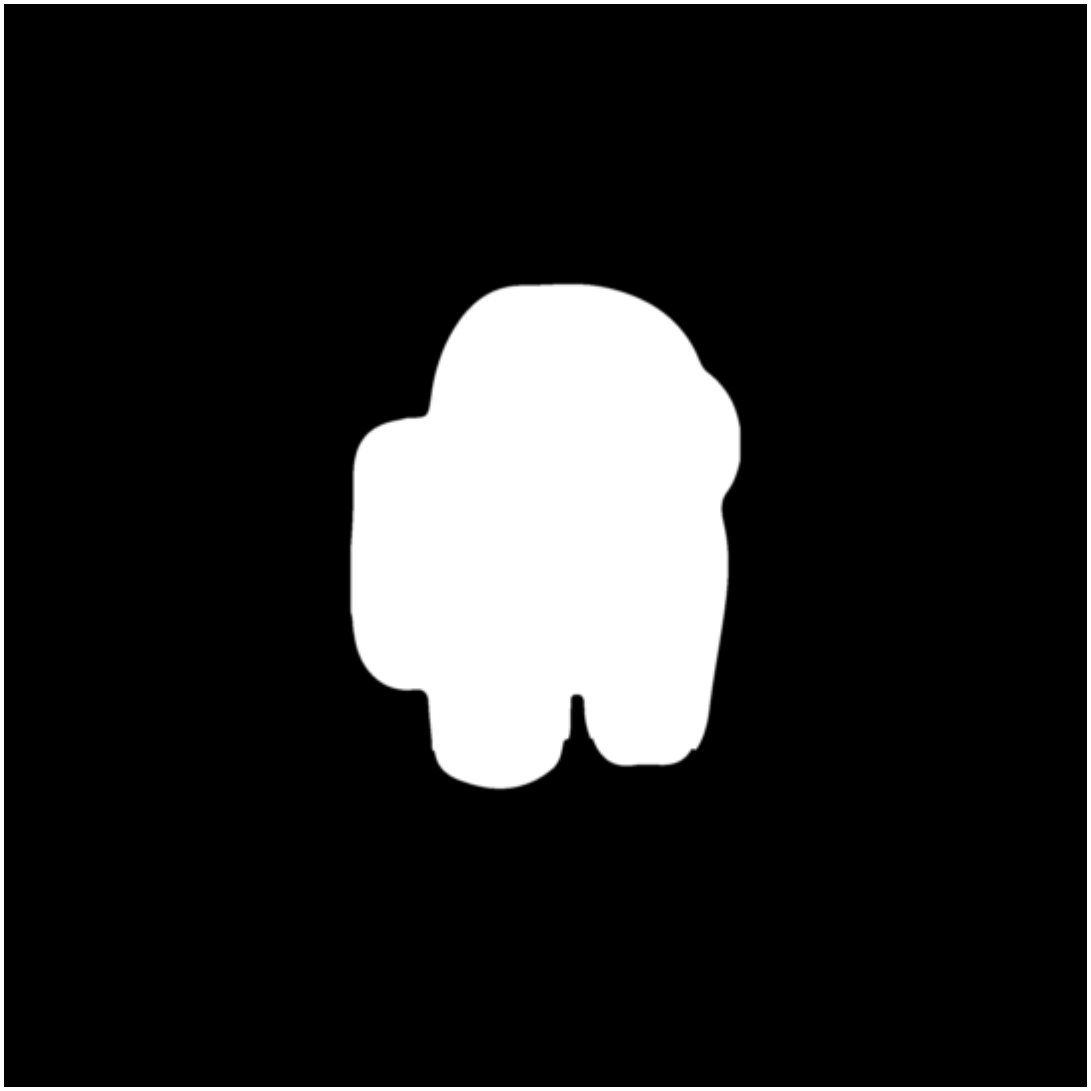


```
begin
    @time amogusSDF = SerialSDF.dijkstraSDF2D(amogus_threshold)
    SDFVis.toImageSDF(amogusSDF, 2.0)
end
```

```
12.791171 seconds (1.75 k allocations: 449.692 MiB, 3.16% gc time)
```

Threshold:

```
begin
    amogus_thres_slider = @bind amogus_thres_val html"<input type=range min=-500 max=500>"
    md"Threshold: $amogus_thres_slider "
end
```



```
Gray.(amogusSDF .< amogus_thres_val)
```


```
begin
    @time pettySDF = SerialSDF.dijkstraSDF2D(petty_threshold)
    SDFVis.toImageSDF(pettySDF)
end
```

```
  0.829416 seconds (326 allocations: 35.718 MiB)
```

Threshold:

```
begin
    petty_slider = @bind petty_slider_val html"<input type=range value=0 min=0
    max=50>"
    md"Threshold: $petty_slider "
end
```


```
begin
    petty_outline = copy(petty)
    for x in 1:size(petty,1)
        for y in 1:size(petty,2)
            if abs(pettySDF[x,y]) < petty_slider_val
                petty_outline[x,y] = RGB(0.1,0.1,0.1)
            end
        end
    end
petty_outline
end
```

```
begin
    # Parallel SDF dependencies
    Pkg.add("MPI")
    using MPI
end
```

```
  Resolving package versions...
 No Changes to `C:\Users\ItsYe\.julia\environments\v1.8\Project.toml`
 No Changes to `C:\Users\ItsYe\.julia\environments\v1.8\Manifest.toml`
```

```
begin
    include("../SDF/ParallelSDF.jl")
    using .ParallelSDF
end
```

```
512×512 Matrix{Float64}:
 218.866  218.203  217.542  216.884  216.228  …  157.151  157.691  158.236  158.784
 218.12   217.455  216.791  216.131  215.473     156.309  156.852  157.399  157.951
 217.376  216.708  216.043  215.38   214.72      155.469  156.014  156.565  157.119
 216.634  215.964  215.296  214.631  213.968     154.63   155.179  155.732  156.29
 215.895  215.222  214.552  213.884  213.219     153.794  154.345  154.902  155.462
 215.157  214.482  213.809  213.14   212.472  …  152.959  153.514  154.073  154.63
 214.421  213.744  213.069  212.397  211.727     152.127  152.684  153.247  153.794
   ⋮                                    ⋱                   ⋮
 192.49   191.814  191.14   190.469  189.801     203.918  204.53   205.145  205.763
 193.227  192.553  191.881  191.213  190.548     204.711  205.32   205.933  206.549
 193.965  193.294  192.625  191.96   191.297     205.505  206.113  206.723  207.337
 194.706  194.037  193.371  192.708  192.048     206.302  206.907  207.515  208.126
 195.449  194.783  194.12   193.459  192.802  …  207.1    207.703  208.309  208.917
 196.195  195.531  194.87   194.213  193.557     207.9    208.501  209.104  209.711
```

```
@time SerialSDF.dijkstraSDF2D(petty_threshold)
```

```
  1.018748 seconds (326 allocations: 35.718 MiB)
```

```
512×512 Matrix{Float64}:
 218.866  218.203  217.542  216.884  216.228  …  157.151  157.691  158.236  158.784
 218.12   217.455  216.791  216.131  215.473     156.309  156.852  157.399  157.951
 217.376  216.708  216.043  215.38   214.72      155.469  156.014  156.565  157.119
 216.634  215.964  215.296  214.631  213.968     154.63   155.179  155.732  156.29
 215.895  215.222  214.552  213.884  213.219     153.794  154.345  154.902  155.462
 215.157  214.482  213.809  213.14   212.472  …  152.959  153.514  154.073  154.63
 214.421  213.744  213.069  212.397  211.727     152.127  152.684  153.247  153.794
   ⋮                                    ⋱                   ⋮
 192.49   191.814  191.14   190.469  189.801     203.918  204.53   205.145  205.763
 193.227  192.553  191.881  191.213  190.548     204.711  205.32   205.933  206.549
 193.965  193.294  192.625  191.96   191.297     205.505  206.113  206.723  207.337
 194.706  194.037  193.371  192.708  192.048     206.302  206.907  207.515  208.126
 195.449  194.783  194.12   193.459  192.802  …  207.1    207.703  208.309  208.917
 196.195  195.531  194.87   194.213  193.557     207.9    208.501  209.104  209.711
```

```
@time ParallelSDF.dijkstraSDF2DSerialUDF(petty_threshold)
```

```
  1.349283 seconds (540.56 k allocations: 62.089 MiB, 50.60% compilation tim
e)
```

```
512×512 Matrix{Float64}:
 218.866  218.203  217.542  216.884  216.228  …  157.151  157.691  158.236  158.784
 218.12   217.455  216.791  216.131  215.473     156.309  156.852  157.399  157.951
 217.376  216.708  216.043  215.38   214.72      155.469  156.014  156.565  157.119
 216.634  215.964  215.296  214.631  213.968     154.63   155.179  155.732  156.29
 215.895  215.222  214.552  213.884  213.219     153.794  154.345  154.902  155.462
 215.157  214.482  213.809  213.14   212.472  …  152.959  153.514  154.073  154.63
 214.421  213.744  213.069  212.397  211.727     152.127  152.684  153.247  153.794
   ⋮                                  ⋱                    ⋮
 192.49   191.814  191.14   190.469  189.801     203.918  204.53   205.145  205.763
 193.227  192.553  191.881  191.213  190.548     204.711  205.32   205.933  206.549
 193.965  193.294  192.625  191.96   191.297     205.505  206.113  206.723  207.337
 194.706  194.037  193.371  192.708  192.048     206.302  206.907  207.515  208.126
 195.449  194.783  194.12   193.459  192.802  …  207.1    207.703  208.309  208.917
 196.195  195.531  194.87   194.213  193.557     207.9    208.501  209.104  209.711
```

```
1 @time ParallelSDF.dijkstraSDF2DParallelUDF(petty_threshold)
```

```
  5.222824 seconds (33.10 M allocations: 1.018 GiB, 6.06% gc time, 19.01% co
mpilation time)
```

```
2160×2160 Matrix{Float64}:
 1104.14  1103.32  1102.5   1101.67  1100.85  …  1033.18  1033.96  1034.74  1035.52
 1103.57  1102.75  1101.93  1101.11  1100.28     1032.55  1033.33  1034.11  1034.89
 1103.01  1102.18  1101.36  1100.54  1099.71     1031.92  1032.7   1033.48  1034.26
 1102.44  1101.62  1100.79  1099.97  1099.15     1031.3   1032.08  1032.86  1033.64
 1101.87  1101.05  1100.22  1099.4   1098.58     1030.67  1031.45  1032.23  1033.01
 1101.31  1100.48  1099.66  1098.83  1098.01  …  1030.04  1030.83  1031.61  1032.39
 1100.74  1099.92  1099.09  1098.27  1097.44     1029.42  1030.2   1030.98  1031.76
   ⋮                                  ⋱
 1074.63  1073.82  1073.01  1072.2   1071.39     1022.35  1023.1   1023.86  1024.62
 1075.21  1074.41  1073.6   1072.79  1071.98  …  1023.0   1023.76  1024.51  1025.27
 1075.8   1075.0   1074.19  1073.38  1072.57     1023.66  1024.41  1025.17  1025.93
 1076.39  1075.58  1074.78  1073.97  1073.16     1024.31  1025.07  1025.82  1026.58
 1076.98  1076.17  1075.37  1074.56  1073.76     1024.97  1025.72  1026.48  1027.23
 1077.57  1076.77  1075.96  1075.15  1074.35     1025.62  1026.38  1027.13  1027.89
```

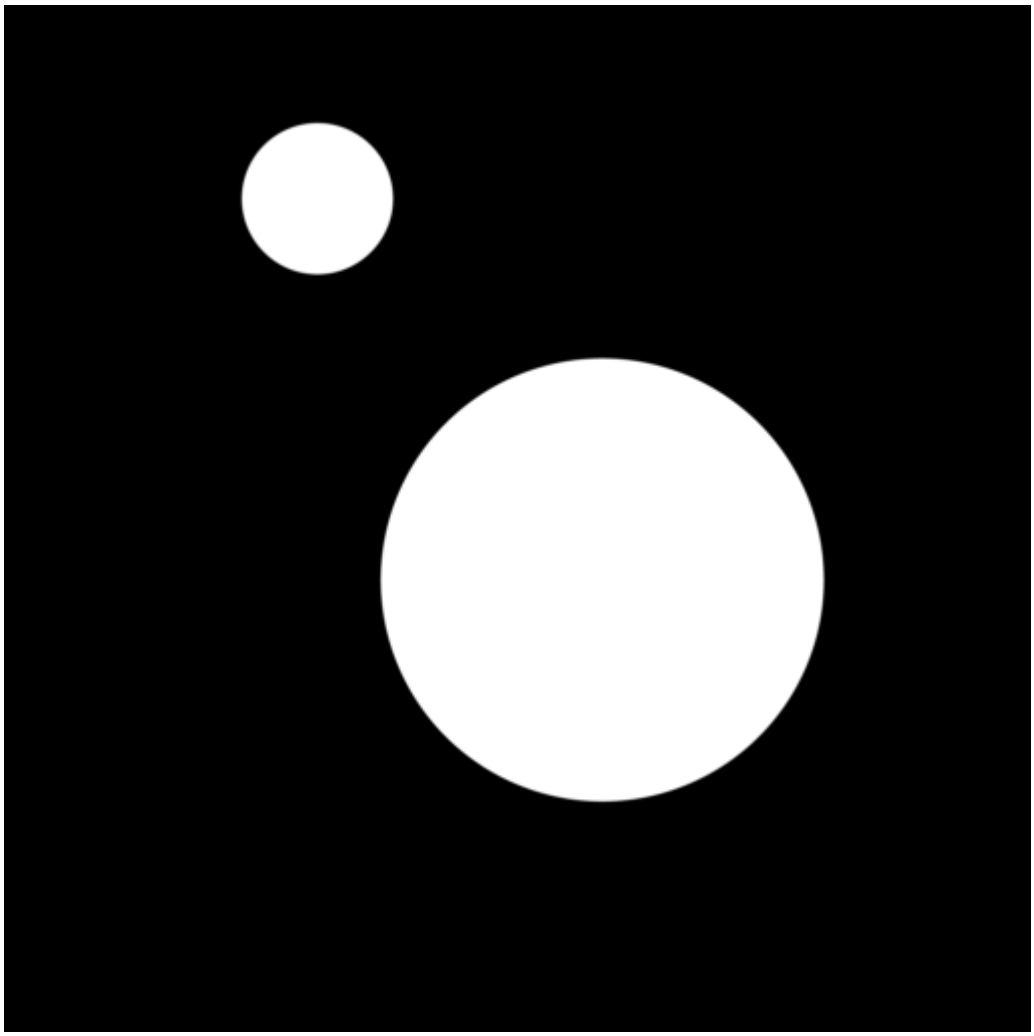
```
1 @time SerialSDF.dijkstraSDF2D(amogus_threshold)
```

```
 12.986069 seconds (1.75 k allocations: 449.692 MiB, 0.39% gc time)
```

```
2160×2160 Matrix{Float64}:
 1104.14  1103.32  1102.5   1101.67  1100.85  …  1033.18  1033.96  1034.74  1035.52
 1103.57  1102.75  1101.93  1101.11  1100.28     1032.55  1033.33  1034.11  1034.89
 1103.01  1102.18  1101.36  1100.54  1099.71     1031.92  1032.7   1033.48  1034.26
 1102.44  1101.62  1100.79  1099.97  1099.15     1031.3   1032.08  1032.86  1033.64
 1101.87  1101.05  1100.22  1099.4   1098.58     1030.67  1031.45  1032.23  1033.01
 1101.31  1100.48  1099.66  1098.83  1098.01  …  1030.04  1030.83  1031.61  1032.39
 1100.74  1099.92  1099.09  1098.27  1097.44     1029.42  1030.2   1030.98  1031.76
    ⋮                                  ⋱
 1074.63  1073.82  1073.01  1072.2   1071.39     1022.35  1023.1   1023.86  1024.62
 1075.21  1074.41  1073.6   1072.79  1071.98  …  1023.0   1023.76  1024.51  1025.27
 1075.8   1075.0   1074.19  1073.38  1072.57     1023.66  1024.41  1025.17  1025.93
 1076.39  1075.58  1074.78  1073.97  1073.16     1024.31  1025.07  1025.82  1026.58
 1076.98  1076.17  1075.37  1074.56  1073.76     1024.97  1025.72  1026.48  1027.23
 1077.57  1076.77  1075.96  1075.15  1074.35     1025.62  1026.38  1027.13  1027.89
```

```
1 @time ParallelSDF.dijkstraSDF2DSerialUDF(amogus_threshold)
```

```
11.509789 seconds (1.78 k allocations: 449.694 MiB, 0.48% gc time)
```

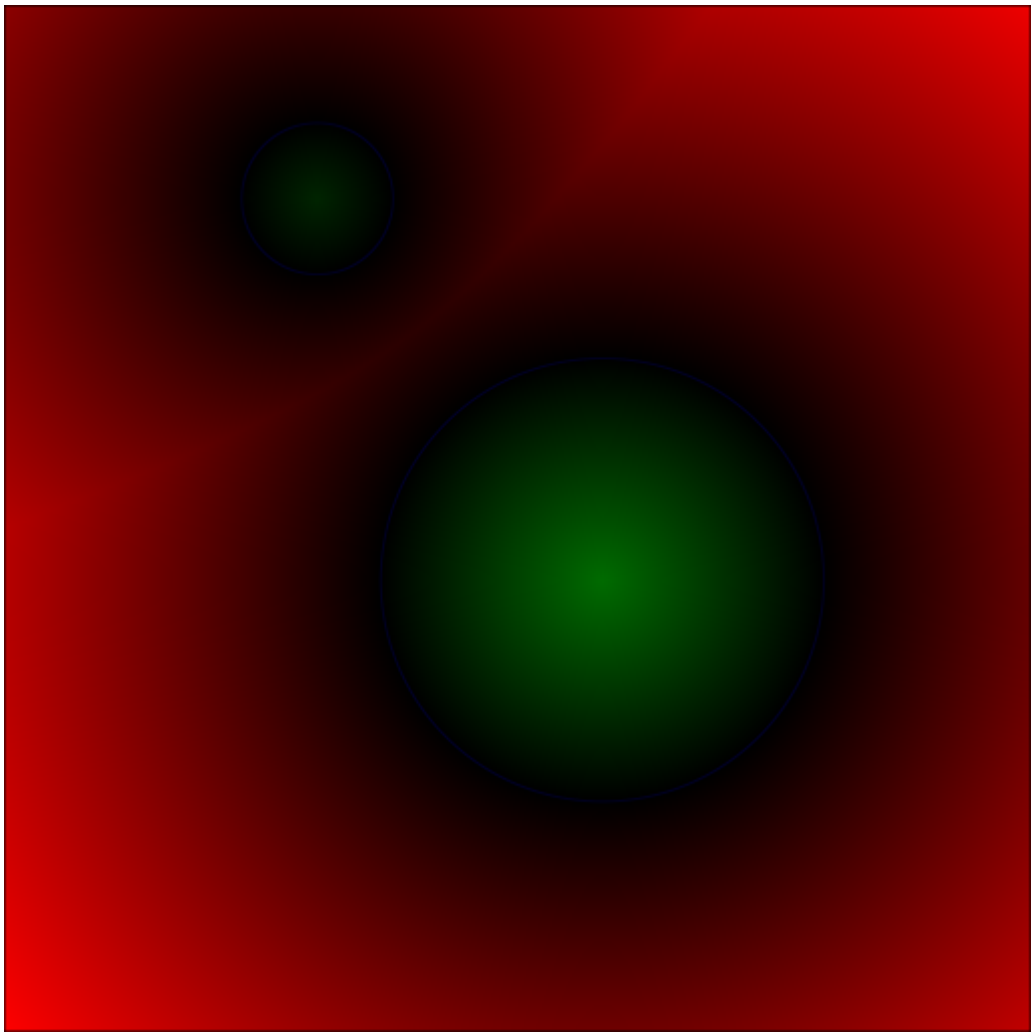


```
1 begin
2     huge = SerialSDF.threshold(load("./images/huge.png"))
3     Gray.(huge)
4 end
```

```
@time hugeParallel =
8192×8192 Matrix{Float64}:
 2327.75  2326.9   2326.04  2325.19  2324.34  …  3946.7   3947.3   3947.9   3948.5
 2327.22  2326.37  2325.52  2324.67  2323.82     3945.9   3946.5   3947.1   3947.7
 2326.7   2325.85  2324.99  2324.14  2323.29     3945.1   3945.7   3946.3   3946.9
 2326.17  2325.32  2324.47  2323.62  2322.77     3944.3   3944.9   3945.5   3946.1
 2325.65  2324.8   2323.95  2323.09  2322.24     3943.5   3944.1   3944.7   3945.3
 2325.13  2324.27  2323.42  2322.57  2321.72  …  3942.7   3943.3   3943.9   3944.5
 2324.6   2323.75  2322.9   2322.05  2321.2      3941.9   3942.5   3943.1   3943.7
    ⋮                                  ⋱           ⋮
 4207.3   4206.5   4205.7   4204.9   4204.1      3190.09  3190.78  3191.47  3192.16
 4207.9   4207.1   4206.3   4205.5   4204.7      3190.81  3191.5   3192.19  3192.88
 4208.5   4207.7   4206.9   4206.1   4205.3      3191.54  3192.23  3192.92  3193.6
 4209.1   4208.3   4207.5   4206.7   4205.9      3192.26  3192.95  3193.64  3194.33
 4209.7   4208.9   4208.1   4207.3   4206.5  …   3192.99  3193.68  3194.37  3195.05
 4210.3   4209.5   4208.7   4207.9   4207.1      3193.71  3194.4   3195.09  3195.78
```

```
1 @time hugeParallel = ParallelSDF.dijkstraSDF2DSerialUDF(huge)
```

```
238.639466 seconds (7.78 k allocations: 5.947 GiB, 0.58% gc time)
```

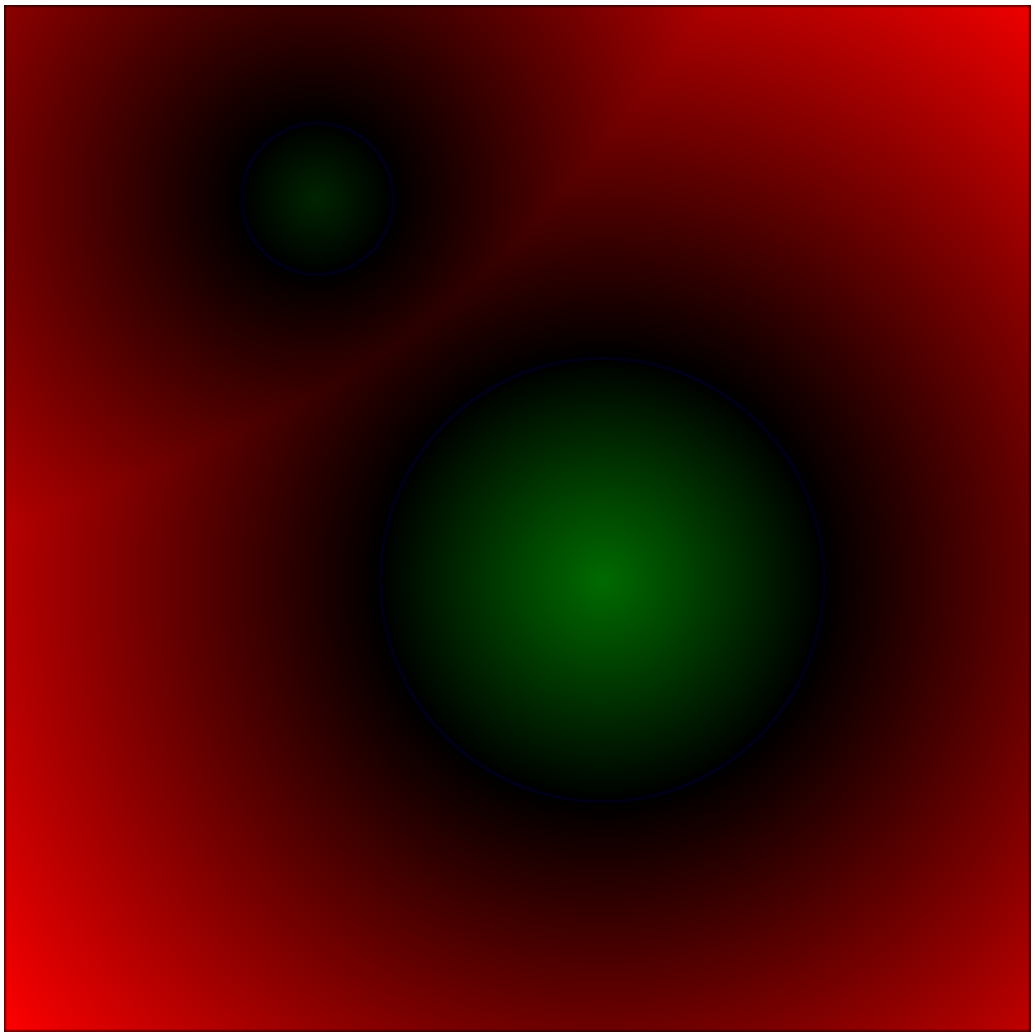


```
1 SDFVis.toImageSDF(hugeParallel)
```

```
@time hugeSerial =
8192×8192 Matrix{Float64}:
 2327.75  2326.9   2326.04  2325.19  2324.34  …  3946.7   3947.3   3947.9   3948.5
 2327.22  2326.37  2325.52  2324.67  2323.82     3945.9   3946.5   3947.1   3947.7
 2326.7   2325.85  2324.99  2324.14  2323.29     3945.1   3945.7   3946.3   3946.9
 2326.17  2325.32  2324.47  2323.62  2322.77     3944.3   3944.9   3945.5   3946.1
 2325.65  2324.8   2323.95  2323.09  2322.24     3943.5   3944.1   3944.7   3945.3
 2325.13  2324.27  2323.42  2322.57  2321.72  …  3942.7   3943.3   3943.9   3944.5
 2324.6   2323.75  2322.9   2322.05  2321.2      3941.9   3942.5   3943.1   3943.7
    ⋮                                  ⋱           ⋮
 4207.3   4206.5   4205.7   4204.9   4204.1      3190.09  3190.78  3191.47  3192.16
 4207.9   4207.1   4206.3   4205.5   4204.7      3190.81  3191.5   3192.19  3192.88
 4208.5   4207.7   4206.9   4206.1   4205.3      3191.54  3192.23  3192.92  3193.6
 4209.1   4208.3   4207.5   4206.7   4205.9      3192.26  3192.95  3193.64  3194.33
 4209.7   4208.9   4208.1   4207.3   4206.5   …  3192.99  3193.68  3194.37  3195.05
 4210.3   4209.5   4208.7   4207.9   4207.1      3193.71  3194.4   3195.09  3195.78
```

```
1 @time hugeSerial = SerialSDF.dijkstraSDF2D(huge)
```

```
276.805300 seconds (7.75 k allocations: 5.947 GiB, 0.21% gc time)
```



```
1 SDFVis.toImageSDF(hugeSerial)
```