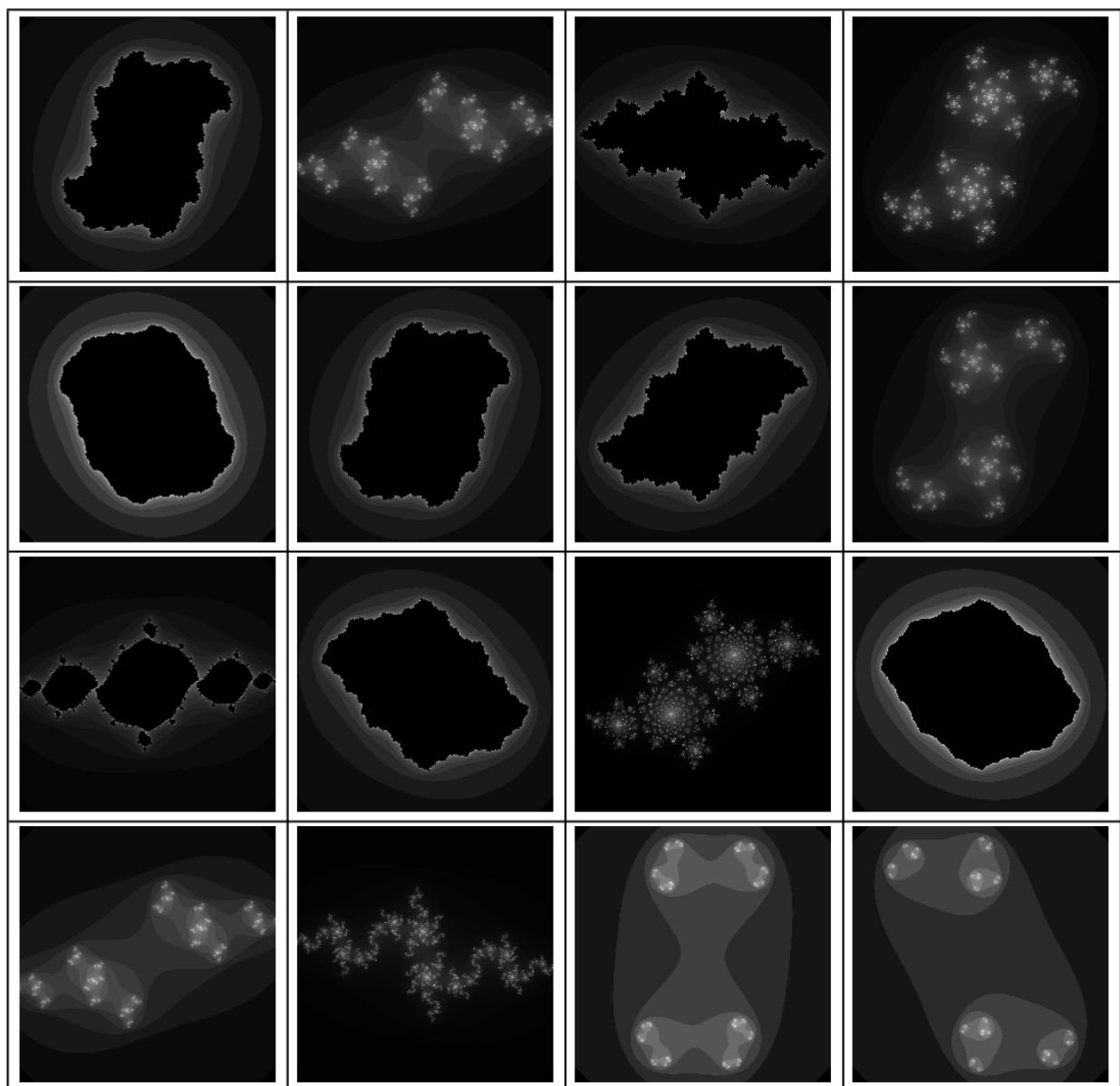


Centaur Beauty in Fractals: GPT-4 Interfaced by a Human to Mathematica

Douglas C. Youvan

doug@youvan.com

September 1, 2023



The concept of "Centaur Beauty" resides at the intersection of art and science, embodying the magical harmony that arises from the convergence of two seemingly divergent fields. This study is an exploration of Centaur Beauty through the lens of fractal geometry, specifically Julia set fractals. The project aims to blend the artistic allure and mathematical rigor of these fractals by employing Shannon entropy as a metric to quantify their aesthetic value.

A 4x4 grid of Julia set fractals, generated using Mathematica, serves as the canvas for our exploration. Each fractal is assessed for its visual complexity through Shannon entropy calculations derived from their grayscale representations. Preliminary findings indicate a positive correlation between higher entropy values and fractals subjectively considered more visually appealing. This not only validates the proposed method but also successfully identifies the "Centaur Beauty" residing within mathematical constructs.

The paper includes Mathematica code for computational reproducibility and a rich catalog of fractal visualizations, each annotated with its corresponding entropy value. By marrying art and science, this study illuminates the objective facets of subjective beauty, thereby contributing to the broader understanding of Centaur Beauty.

{2.356373445663676,
2.9209476677432216,
2.5042196140007253,
3.2725462941556613,
2.2363443197585435,
2.312768157277956,
2.375222695791603,

```

2.9583471042089142,
2.651943540624304,
2.294460838079769,
4.100566324248899,
2.235391381278046,
2.57338319740044,
3.3905750777563073,
2.195509357066584,
1.9767814596086128}

```

```

GenerateJulia[c_, maxIter_, res_] :=
  Image@
  Rescale@
  Table[Module[{z = x + I y, i = 0},
    While[Abs[z] < 2 && i < maxIter, z = z^2 + c; i++];
    If[i == maxIter, 0, i]], {y, -1.5, 1.5, 3. / res},
    {x, -1.5, 1.5, 3. / res}];

calcEntropy[img_Image] :=
  Module[{data, counts, prob},
    data = Flatten[ImageData[ColorConvert[img, "Grayscale"]]];
    counts = Last /@ Tally[data];
    prob = counts / Total[counts];
    N[-Total[prob Log2[prob]]]];

grid = Table[c = RandomComplex[{-1 - I, 1 + I}];
  img = GenerateJulia[c, 1000, 250];
  ent = calcEntropy[img];
  {img, ent}, {i, 4}, {j, 4}];

Grid[Map[First, grid, {2}], Frame → All]
Flatten[Map[Last, grid, {2}]]

```