

Cantor shadows (using Iterated Function Systems and Regions)

GONZALO GARCÍA ALARCÓN ESTRADA, *****

Dec / 01 / 2017

University of Toronto, Shameless Mathematica

Shadows of the Cantor square:

Draw an approximation of the Cantor square C_2 , rotate it by an angle θ and project the rotation vertically, draw the projection, and compute its measure. Manipulate.

DECLARATION OF FUNCTIONS

```
(* initial square *)
(* R=ImplicitRegion[0<= x<= 1^&theta<= y<= 1,{x,y}]; *)
r = Rectangle[];
```

```
(* transformations *)
t1 = AffineTransform[IdentityMatrix[2] * (1/3)];
t2 = AffineTransform[{IdentityMatrix[2] * (1/3), {0, 2/3}}];
t3 = AffineTransform[{IdentityMatrix[2] * (1/3), {2/3, 2/3}}];
t4 = AffineTransform[{IdentityMatrix[2] * (1/3), {2/3, 0}}];
T[r_] := RegionUnion[TransformedRegion[r, t1],
  TransformedRegion[r, t2], TransformedRegion[r, t3], TransformedRegion[r, t4]];
rota[r_, theta_] := TransformedRegion[r, RotationTransform[theta]];
(* rotarad[r_,theta_]:=TransformedRegion[r,RotationTransform[theta]]; *)
cantorN[r_, n_Integer?NonNegative] := Nest[T, r, n];
```

```
(* projections *)
Proj[r_] := Resolve[Exists[{x, y} in r, Reals] // Reduce; (* project *)
r2interval[r_] := Module[{p = Proj[r]}, Switch[Head[p],
  Or,
  Module[{aux}, aux = List@@@List@@p /. {a_, LessEqual, x, LessEqual, b_} => {a, b};
  Return[Interval@@aux]],
  Inequality,
  Module[{aux},
  aux = List@@p /. {a_, LessEqual, x, LessEqual, b_} => {a, b};
  Return[Interval@aux]]
]];
(* transform into an interval that Mathematica understands *)
```

```

(* plotting *)
Disp[r_] := RegionPlot[r, PlotRange → RegionBounds[r],
  BoundaryStyle → None, PlotStyle → Black, Frame → False];
Disp[r_Interval] := NumberLinePlot[r, {x, Min[r], Max[r]},
  PlotStyle → {PointSize[0], Red, Thick}, PlotRange → {{Min[r], Max[r]}, Automatic}];
pos[r_] := TransformedRegion[r, TranslationTransform[{0, 0.5}]];

Shadow[n_Integer?NonNegative,  $\theta$ ] (* Takes in DEGREES!!!*) := Module[{set, proj},
  If[(n + 1) > Length[C2], Return["n is too big"],
  set = rota[C2[[n + 1]],  $\theta$  Degree];
  proj = set // r2interval;
  Return[
    Show[set // pos // Disp, proj // Disp, lab[set] // Graphics, lab[proj] // Graphics]]]
]

(* text in figures *)
lab[r_] := Text[
  Style["Measure of set: " <> formatlab[r], Gray, FontSize → 13], Scaled[{0.17, .6}]];
lab[r_Interval] := Text[Style["Measure of projection: " <> formatlab[r],
  Red, FontSize → 13], Scaled[{0.22, 0.55}]];
formatlab[r_] := If[Head[RegionMeasure[r]] === Rational,
  RegionMeasure[r] /. Rational[p_, q_] => ToString[p] <> "/" <> ToString[q],
  ToString[RegionMeasure[r] // N]
]

```

```

(* pre-generate cantor squares, as REGIONS *)
i = 4; (*number of iterations*)
C2 = NestList[T, r, i];

```

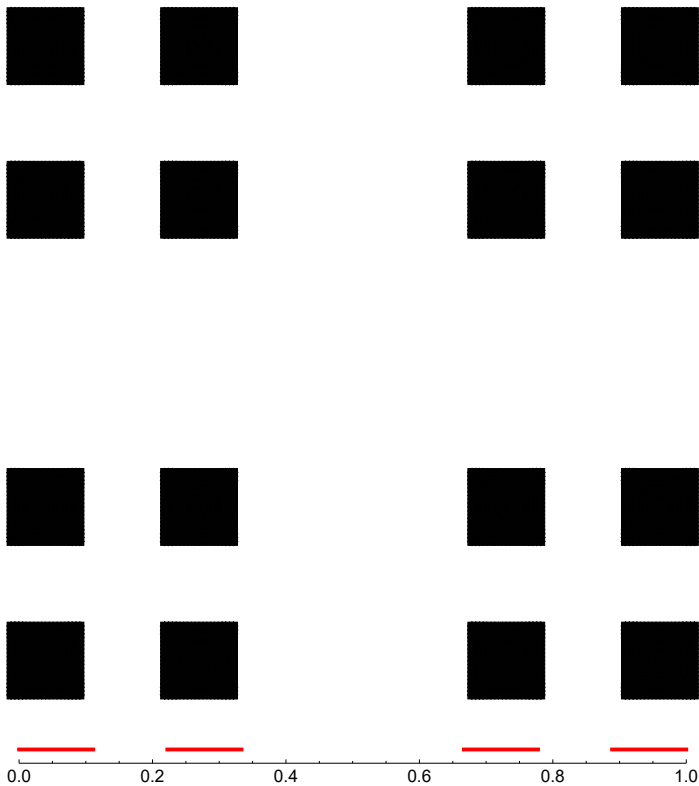
FIRST : AN EXAMPLE (directly generated, not pre-generated)

```

C2N = cantorN[r, 2]; C2Nint = C2N // r2interval;

a = C2N // pos // Disp
b = C2Nint // Disp
C2Nint
C2Nint // RegionMeasure
Show[lab[C2N] // Graphics, lab[C2Nint] // Graphics]

```



Interval $\left[\left\{ 0, \frac{1}{9} \right\}, \left\{ \frac{2}{9}, \frac{1}{3} \right\}, \left\{ \frac{2}{3}, \frac{7}{9} \right\}, \left\{ \frac{8}{9}, 1 \right\} \right]$

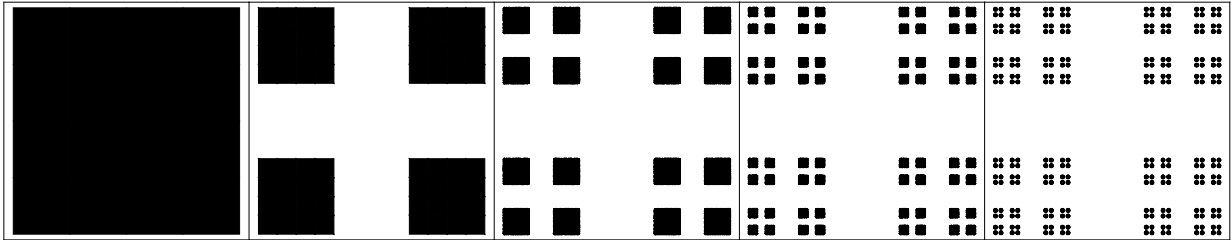
$$\frac{4}{9}$$

Measure of set: 16/81

Measure of projection: 4/9

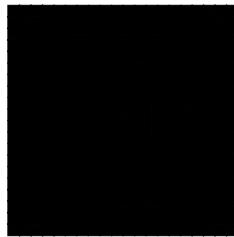
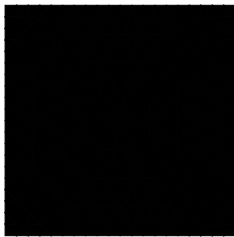
AN APPROXIMATION TO THE CANTOR SQUARE

GraphicsRow[Graphics /@ Disp /@ C2, Frame → All, ImageSize → Full]



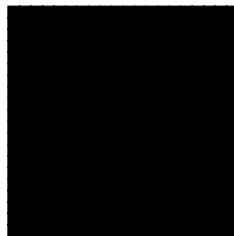
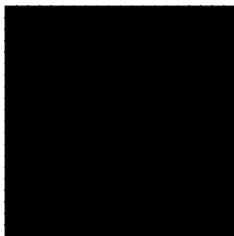
ROTATIONS

Shadow[1, θ] (* n, θ ... takes θ in Degrees *)



Measure of set: 4/9

Measure of projection: 2/3



Manipulate[Quiet[Shadow[n, θ]], {{n, 0}, 0, Length[C2] - 1, 1}, {{ θ , 0}, 0, 90}]

