Arnol' d Cat Transform Generator

By Rongzhong Li

Definition: http://en.wikipedia.org/wiki/Arnold's_cat _map

Change the filename here. Then you can simply go to Evaluation and evaluate the whole notebook.

In[1]:=
```
filename =
  "D:\\userdata\\desktop\\Arnol'd cat\\cat.jpg";
```

In[2]:=
```
M = {{2, 1}, {1, 1}}; (*transform matrix*)
Import[filename]
{w, h} = Import[filename, "ImageSize"];
Print[Style[
  StringJoin["Size: ", ToString[w], "×", ToString[h], " pixels"], FontFamily -> "Arial",

  20]]
```

Out[3]=

Size: 252×189 pixels

You may need to expect how complex the transform will be.
So don' t abuse your machine with too large size.

In[6]:=
```
size = Max[w, h];
Arnold[N_] := {n = N;
  m1 = map[0] = Table[{i, j}, {i, n}, {j, n}];
  m2 = Table[{0, 0}, {i, n}, {j, n}];
  g[x_, y_] := Mod[M. (x, y), n, 1];
  For[t = 1, m2 ≠ map[0], t++, {For[i = 1, i < n + 1, i++,
     For[j = 1, j < n + 1, j++, m2[[g[i, j][[1, 1]], g[i, j][[2, 1]]]] = m1[[i, j]]]],
   map[t] = m1 = m2}]}
Arnold[size];
Print[Style[StringJoin["It will take ", ToString[t - 1],
   " steps to return its original image. The GIF file would be ",
   ToString[Round[size^2 * t/10^6, 0.01]], " × compress_rate MB."], FontFamily → "Arial", 20]]
```

It will take 24 steps to return its original image.

The GIF file would be 1.59 × compress_rate MB.

Execute the following codes to show detailed transform by steps

In[10]:=
```
color = Import[filename, "RGBColorArray"];
Sqcolor = SparseArray[color, size, RGBColor[1, 1, 1]];
pic[t_] :=
  ArrayPlot[Table[{Sqcolor[[map[t][[i, j]][[1]], map[t][[i, j]][[2]]]]}, {i, h}, {j, w}],
   PixelConstrained → True];
Sqpic[t_] := ArrayPlot[Table[{Sqcolor[[map[t][[i, j]][[1]], map[t][[i, j]][[2]]]]},
    {i, size}, {j, size}], PixelConstrained → 1];
Manipulate[pic[n], {n, 0, t - 1, 1}]
```

Out[14]=

Export the GIF animation to the same directory

In[15]:=
```
blank = ArrayPlot[Table[RGBColor[1, 1, 1], {i, h}, {j, w}]];
p0 = pic[0];
gif = StringReplace[filename, "." ~~ ___ → "_ani.gif"];
Export[gif, Join[Table[p0, {i, 10}],
   Table[pic[i], {i, 0, t - 1}], Table[p0, {i, 10}], Table[blank, {i, 20}]], "gif"];
Print[Style[StringJoin["GIF file:\n\t", StringReplace[filename, "." ~~ ___ → "_ani.gif"]],
  FontFamily → "Arial", 20]]
```

GIF file:

D:\userdata\desktop\Arnol'd cat\cat_ani.gif

What' s cool is that you can encode and decode your image using this cycle

- **First generate the transform series and save a coded image at certain steps.**

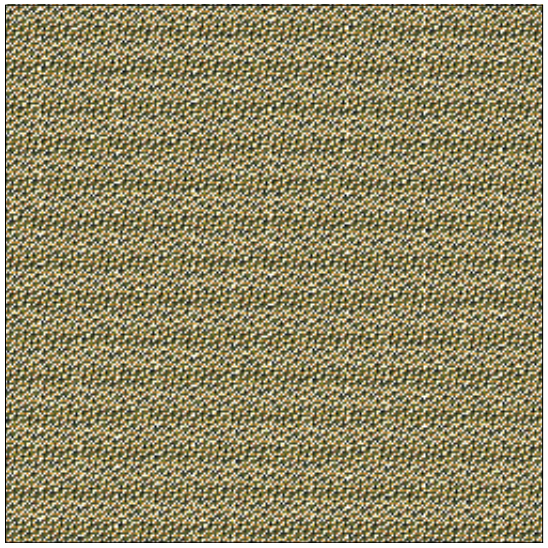
In[20]:=
```
step = 6;
```

In[21]:=
```
codeformat = "bmp";
name = StringSplit[filename, "."][[1]];
format = StringSplit[filename, "."][[2]];
coded = StringJoin[name, "_coded@step", ToString[step], ".", codeformat];
decoded = StringJoin[name, "_decoded", ".", format];
Export[coded, Sqpic[step], codeformat];
Print[Style[StringJoin["Coded file:\n\t", name,
   "_coded@step", ToString[step], ".", codeformat], FontFamily → "Arial", 20]]
Sqpic[
 step]
```

Coded file:

D:\userdata\desktop\Arnol'd cat\cat_coded@step6.bmp

Out[28]=



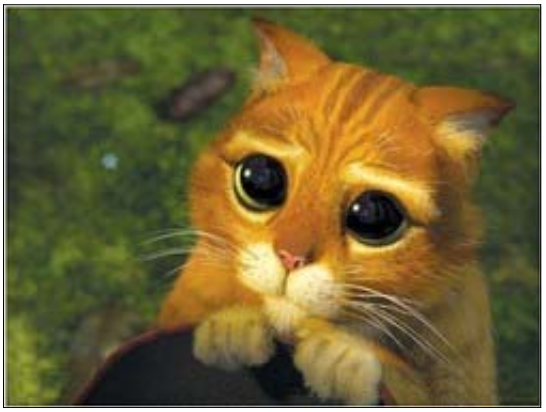
- **Then use the saved image as a new input and transform again.**

In[29]:=
```
Recolor = Import[coded, "RGBColorArray"];
ReSqcolor = SparseArray[Recolor, size, RGBColor[1, 1, 1]];
Repic[t_] :=
  ArrayPlot[Table[{ReSqcolor[[map[t][[i, j]][[1]], map[t][[i, j]][[2]]]]}, {i, h}, {j, w}],
   PixelConstrained → 1];
ReSqpic[t_] := ArrayPlot[Table[{ReSqcolor[[map[t][[i, j]][[1]], map[t][[i, j]][[2]]]]},
    {i, size}, {j, size}], PixelConstrained → 1];
return = t - 1 - step;
Print[Style[StringJoin["Decoded file:\n\t", decoded], FontFamily → "Arial", 20]]
Export[decoded, Repic[return], format];
Repic[return]
```

Decoded file:

D:\userdata\desktop\Arnol'd cat\cat_decoded.jpg

Out[36]=



- **The decoded image is slightly different from the original one due to image compression. You can get a better effect if you start from early steps, or use high quality pic format.**

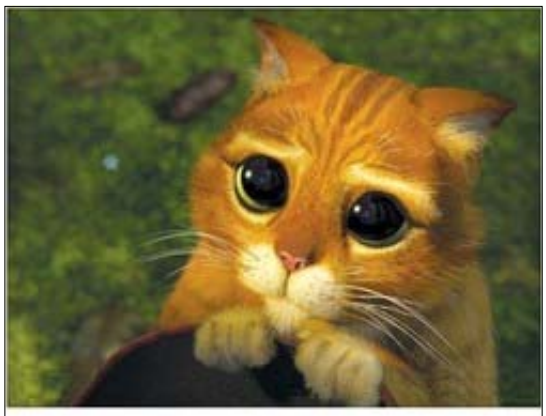
Overall Procedure

In[37]:=
```
Import[filename]
DoubleLongRightArrow[MatrixForm[{"Squared", Sqpic[0]}], MatrixForm[{"Coded", Sqpic[step]}],
 MatrixForm[{"Reformed", ReSqpic[return]}], MatrixForm[{"Decoded", Repic[return]}]]
```

Out[37]=

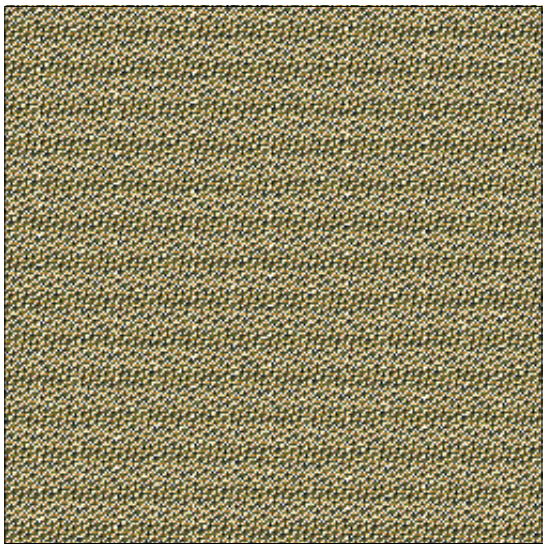
Out[38]=

Squared



⟹

Coded

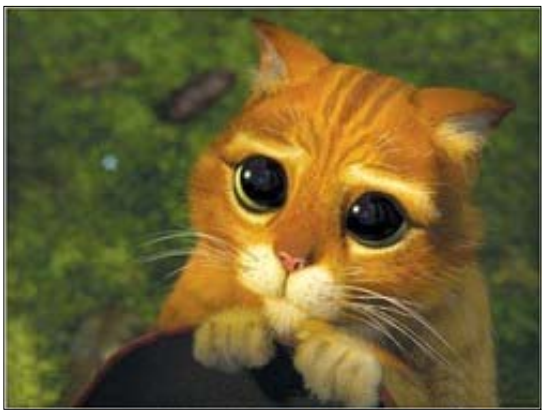


⟹

Reformed

⟹

Decoded



And you can even change the transform matrix M to get a new coding key.