Fractal Image Compression

INF5080

Arild Berggen

Presentation @ 11 May 2005

Fractal Image Compression

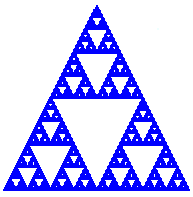
- Is a lossy compression method developed in the late 80's
- Is patented
- Compression ratios from 1:4 to 1:100
- Compressing is slow.
- Decoding is fast.
- Decoding is resolution independent (Resolution enhancements → Zoom)
- Best suited for images of natural scenes, such as mountains, trees, clouds etc.
- Compresses colour images better than grey scale images.

Fractals?

- Fractal compression has nothing to do with fractals such as the infamous Mandelbrot and Julia sets.
- But it uses the same principles of iteration, and self similarity.
- The fractal in fractal compression is a (Partitioned) Iterated Function System

Iterated Function System

- A set of linear functions (transformations) when applied to a starting point will generate a new point.
- x1 = ax + by + e
  y1 = cx + dy + f
- When the transformations are repeated on the new points iteratively an image is generated.
- Can produce complex images.
- Is it possible to do the reverse? To generate functions from an image?
- Problem not solved today.
- Instead of finding transformations that describe the whole picture, find transformations that only apply to portions of a picture
  - Partitioned Iterated Function Systems.



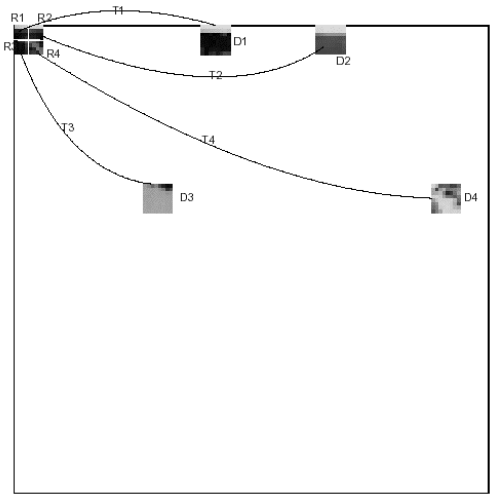
Fractals in the real-world

- If you look at the texture of the floor of the room you are currently in, you will notice that there are many repeating patterns.
- If you take a copy of a small part of the floor, you will find that there are many areas that are nearly identical to that copy.
- And if you rotate it, scale it, mirror it, you will find that you can get even more areas to match.
- If you now store the positions of the copy and the matching area of the floor, including the rotations etc you just made, you can create a mathematical representation of that part of the floor.
- You can do this until you have described the entire floor.
- This is what fractal compression does.

Fractal Compression

- The idea is to look for parts in a picture that resembles other parts in the same picture.
- The picture is divided into range blocks (small) and domain blocks (big).
- Uses a domain block to describe a range block.
- For each range block, search for the domain block that most closely resembles the range block.
- Transformations such as scaling, translation, rotating, sharing, scaling etc and adjustment of brightness/contrast are used on the domain block in order to get the best match.

Compression



- Mapping of range blocks (R) and domain blocks (D) and their transformations (T)

Compression

- The searching that is done when trying to find the most suited domain block for each range block takes a lot of time.
  - Have to check many transformations per domain block.
- Hard to automate properly, and one gets higher compression ratios with human intervention.
  - A human finds the best domain block.
- Can take from 5 seconds up to 5 hours (or more) to compress an image.
- What is stored in the compressed file:
  - The translation done on the domain blocks to match the position of their associated range blocks
  - The transformations done on the domain blocks.
  - The colour, brightness, contrast adjustments.
- This information is called fractal codes.
- No pixels are stored at all, only mathematical functions.

Range blocks and domain blocks

- Typical pixel sizes of the range and domain blocks:

Range	4x4	8x8
Domain	8x8	16x16

- The domain blocks are double the size of the range blocks.
- Smaller size of the blocks means a larger compressed file, because of more fractal codes.

The compressed file

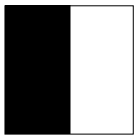
- Compression of a **320x200 24 bit true colour image.**

Compression	Size (KB)	Rate
none	192.0	1:1
low	15.4	1:12
medium	10.6	1:19
high	4.1	1:48

Decompression

- Decompression is done by applying the transformations and translations described in the fractal codes on an arbitrary image (usually just a grey background) iteratively until an image is produced that looks approximately like the original.
- Easy to automate.
- Is fast.
- Number of iterations varies from 4 to 12+
- Decoding is resolution independent:
  - If a 64x64 image was compressed, it can be decompressed to any size (e.g. 128x128) without as much loss in quality as for a normal zoom.

Decompression



Initial image

The transformations and mappings are applied on the initial image iteratively until the image is restored.

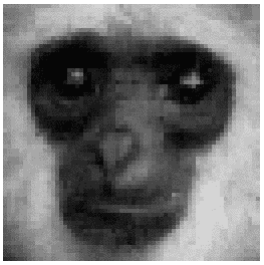
1st iteration 2nd 3rd 4th 5th 6th

Resolution independent decompression (Zoom)

- Note that the zoomed fractal image seem to contain more detail than the zoomed JPEG.
- But the detail has been generated, so zooming in on a person's face will not show pores.





Full sized original | Fractal image | Zoomed JPEG

Zoom x8

JPEG vs. Fractal Compression

- JPEG is better at low compression ratios, and Fractal Compression is better at high.
- Crossover point at 40:1
- The fractal compressed images have a much more natural looking noise than JPEG.
- Same decompression time as JPEG, sometimes faster.
- Fractal compression much slower compression time than JPEG.

- Can zoom on the fractal image and the image will still have a natural look → higher effective compression ratios(?)

Applet demo

- An interactive java applet demo that shows both compression and decompression.

- http://www.eurecom.fr/~image/DEMOS/FRACTAL/english/

Notes

- A promising technology in the late 80's but did not become widely used:
  - It was patented.
  - Slow compression.
  - Only better than JPEG at low quality levels.
  - JPEG was used instead.
- Not standardized, but patent holders Iterated Systems has created a file format called FIF (Fractal Image Format)
  - Plug-in in Netscape, PhotoShop etc.
  - Licensed for use in Microsoft Encarta Encyclopedia
  - Can buy a licence → Get a Windows dll containing functions to compress, decompress images.
- Is currently no public domain documentation available.
  - There are hope that the decompression part will be released.
- Is now outperformed by wavelet compression.
- Work is going on to incorporate DCT and wavelet into fractal compression.

References:

- http://www.cs.northwestern.edu/~agupta/_projects/image_processing/web/FractalImageCompression/
- http://www.faqs.org/faqs/compression-faq/part2/section-8.html
- http://www.cs.queensu.ca/home/xiao/doc/coding.pdf
- http://etd02.lnx390.lsu.edu/docs/available/etd-0124103-142051/unrestricted/02Chapter3.pdf