Image Compression Project

GA-Based Image Compression using GEP

Approach – Use Matlab

• Input a grayscale image of 256x256 bytes
• Compress the image using the GEP approach
  – Compare compressed size to, in terms of bits/pixel
    • Original image (we know it is 8+ - Why 8+ and not 8?)
    • JPEG and JPEG 2000 compression

• Week 3-5
  – Decide on sample gray-scale images and get JPEG and JPEG2000 versions of images along with compression ratios
  – Develop Matlab-based GEP multi-genic evaluator
  – Determine structure and semantics of GA-based GEP algorithm for image compression
    • Given a linear string of integers F(X), evolve a multigenic GEP such that given X it outputs F(X)

Hypothesis to Test

• A GA-based GEP system is a potentially effective too for image compression

• Week 1-2
  – Introduction lectures
• Week 3-5 Overview
  The first three weeks will be spent getting familiar with GEP, finding some images to work with (a gray scale Lena is almost a must), developing a format for a Matlab compatible multi-genic genetic expression (GE), and developing a Matlab function(s) to evaluate a multi-genic GE

• Week 6-7 Overview
  We will not evolve a GE for an entire picture, but multiple GE’s for portions of pictures. The portions will be transformed, quantized, etc. into something that is “simple” enough to evolve the GE.

  One important issue will be to determine when we have a simple enough image segment.

  We will likely apply our algorithm to find or create the needed image segments and then work on the segments separately but using one single GEP algorithm.

  The output of this process will be a series of evolved GE’s representing the image.

  Hopefully the space required by the GE’s will be less than that required by the original image.
• Week 6-7
  – Develop a means to convert an image to an intermediate form in which the image is represented as a string or strings of integers.
    • DCT => quantize => linearize
    • DWT (levels?) => quantize => linearize
  – Develop means for testing complexity of an image segment in intermediate form
    • If too complex, segment further; otherwise can’t find a GEP to compress it.
  – Convert images to intermediate forms
    • Will also need ability to reverse this form back to the original image
  – Develop algorithm for applying GEP-based GA to the strings to get a compacted representation

• Week 7-10
  – Fine-tune GA/GEP algorithm
  – Start report (outline and data needed)
  – Generate results, i.e. perform some compressions, and see how they work
  – Finish data collection
  – Complete report and presentation