Performance of Natural Still Image Processing

Akis Gkaitatzis

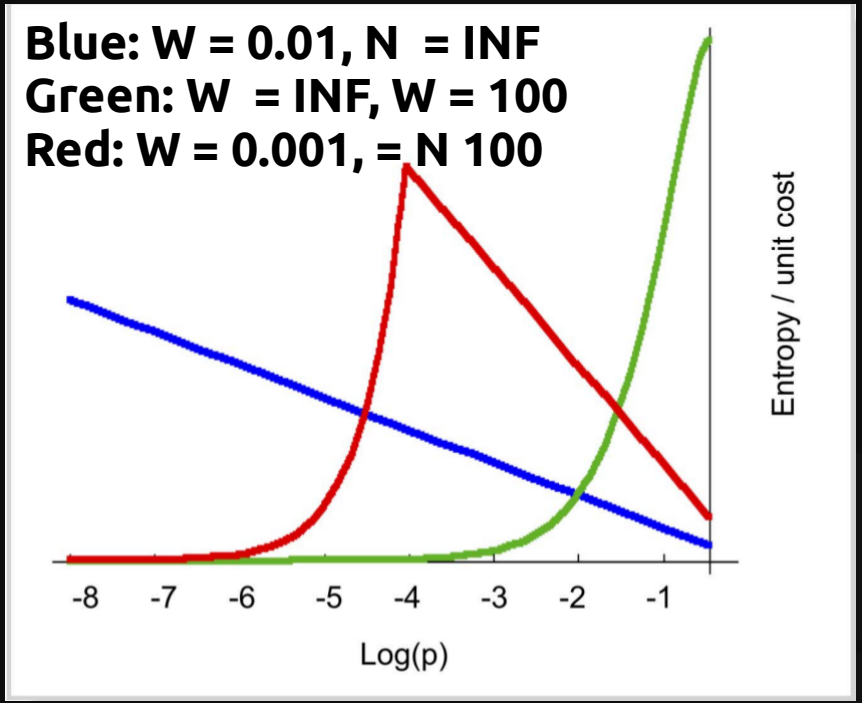
General Assembly & Executive Board Meeting
2015.11.12

Theory

- Need to perform data reduction on an image while maintaining the necessary information for visual identification of the object
- We want to keep tha patterns which maximize the value of

$$f(p) = \frac{-p\log(p)}{max(1/N, p/W)}$$

where p if the pattern frequency, N the number of patterns we want to keep and W the bandwidth cost



shrimpSim

- Simulation is necessary
  - LUT generation for the hardware
  - Verification of hardware's operation
  - Proof of concept
- shrimpSim is developed with C++11 and OpenCV
  - Currently works for patterns with 2 levels of grey (black and white)

Simulation of Image Processing I

- Transform coloured image to grey-scale (256 levels)

Simulation of Image Processing II

- Transform greyscale to b&w with a threshold (default 150)

120 (Cleaner image)

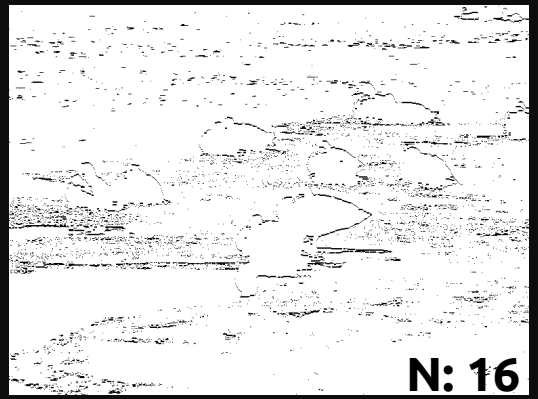
150

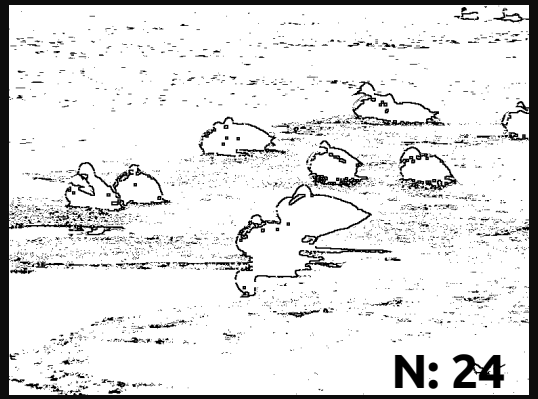
180

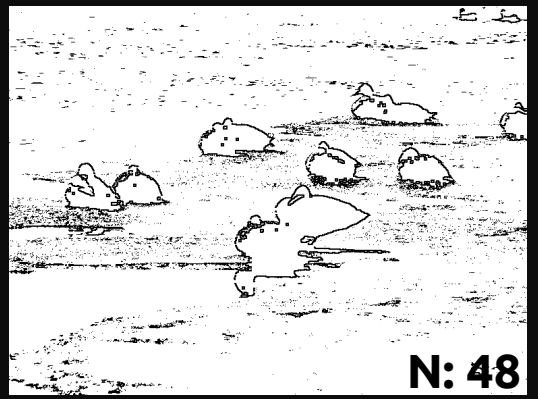
Simulation of Image Processing III

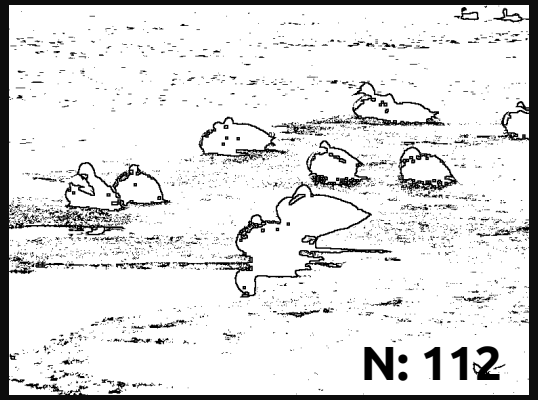
- Loop over the image, find all 3x3 patterns
- Calculate the frequency and entropy of each pattern
- Select N patterns with the maximum entropy
- Loop over the image, keep in place only selected patterns

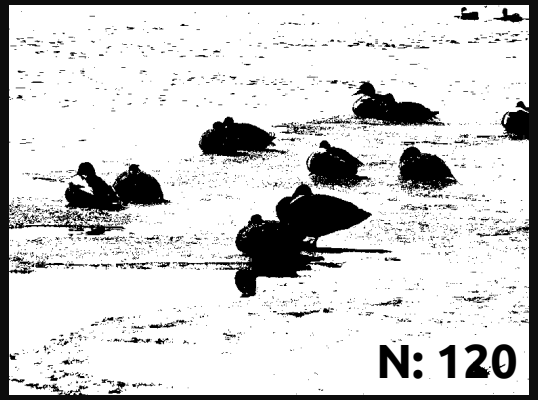
Simulation of Image Processing IV







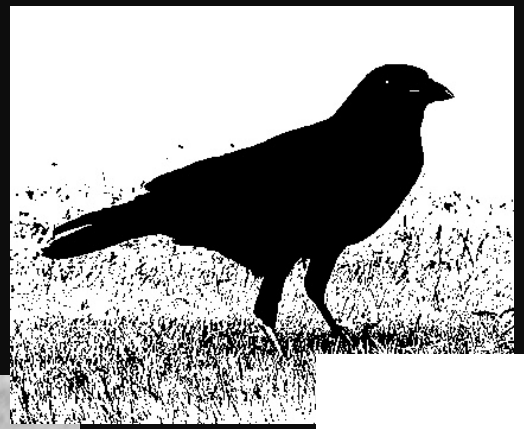




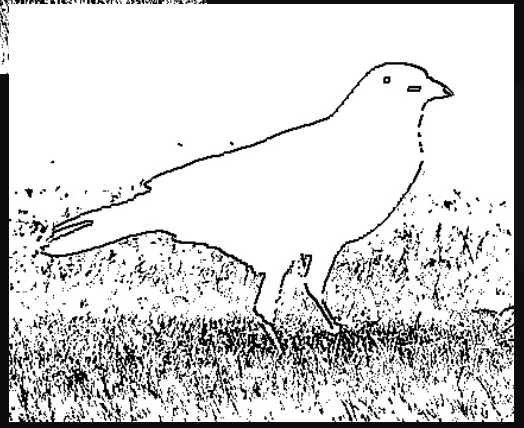
Crow Image Results

Coloured

Grey

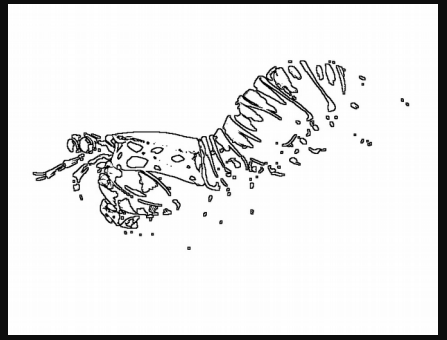


Black & White
Threshold: 150



Reconstructed
W: 0.05
N: 50

Shrimp Image Results



Reconstructed
W = 0.05, N = 50

Future

- The first version proved that the concept is concrete but
  - It was working only for black and white images
  - Did not support Pattern Banks
- Second version is currently under development
  - Developed with C++0x, uses Boost and OpenCV
  - Able to work on N levels of grey
  - Supports Pattern Banks
  - Support for 3rd dimention

Grazie!