An Attempt at Face Detection on SRC-6

David Meixner, Volodymyr Kindratenko
National Center for Supercomputing Applications
University of Illinois at Urbana-Champaign

Introduction

Face detection finds uses in image retrieval, surveillance, and many other applications. Intel's OpenCV Library has one of the most comprehensive algorithms implemented, however, because of its complexity, the algorithm does not execute in real-time. Thus, we investigated the use of SRC-6 for accelerating the execution of the OpenCV face detection algorithm. While this was not the case, the work led to a better understanding of the types of algorithms that perform well on the SRC-6.

The Algorithm

- Compare Haar-like features to classifiers
- Classifiers are cascaded
- If classifiers are statistically close to the search window, proceed to next classifier, otherwise this window is not a face
- The statistical analysis is a simple pixel sum over a rectangular region (use a summed area table)
- Begin with a search window of pixel size 20×20 and then repeat the process with increasing window sizes

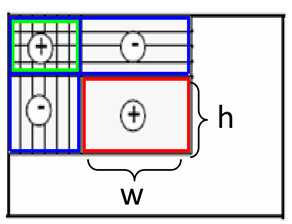
Implementation

- 4 Table lookups for each rectangle
- 2 - 3 rectangles for each feature
- Up to 12 table lookups for each feature
- Run multiple search windows in parallel
- 7 separate search windows fit on the two SRC-6 FPGAs (Virtex-II Pro xc2vp100)

Summed Area Table

A fast way to find the sum of pixels in any rectangular region

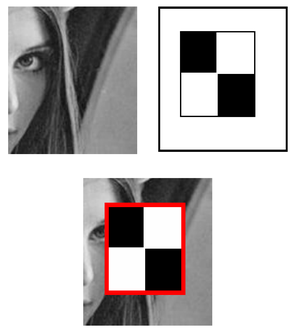
$$SAT(x,y)=\sum_{x'\le x,y'\le y} I(x',y')$$



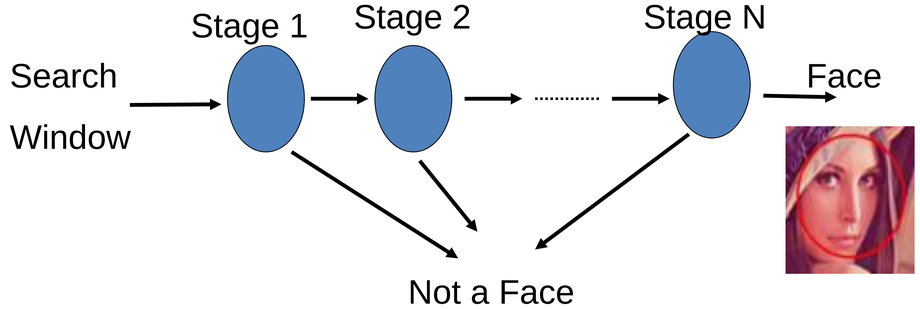
$$RecSum(r)=SAT(x,y)+SAT(x+w,y+h)-SAT(x,y+h)-SAT(x+w,y)$$

Examples of Haar-Like Features

Feature Comparison



Cascaded Classifier



- Each stage has an increasing number of features
- In this case, the first stage has 3, the last has 213
- ~70-80% of the candidates are rejected within the first two stages

Modifications

- Instead of increasing search window size, shrink picture
  - Search window remains small (20×20 pixels)
- Search window contents can be stored in BRAM instead of onboard memory
- The 12 lookups can now be done in one clock
- Multiple search windows can be run simultaneously

Iteration 1 Iteration 2

Pseudo Code

```
for (i=0; i<stages; i++) {
   stage_sum=0;
   for (j=0; j<features_in_stage[i]; j++) {
      feature_sum = whole_pixel_sum*weight1;
      feature_sum += black_pixel_sum1*wieght2;
      feature_sum += black_pixel_sum2*weight3;
      stage_sum += feature_sum;
   }
   if (stage_sum < stage_threshold[i]) {
      result = -1;
      EXIT;
   }
}
result = 1;
```

Lookup values for each sum = 12 total lookups

Conclusions

- Tested with image size of 640x480
- Summed area table is calculated in real-time
- Unfortunately, did not achieve faster results (0.5 fps)
- Code could not be fully pipelined (five nested loops)
- Limited FPGA resources
- Only 7 simultaneous search windows