Improved Pruning and Parallel Techniques in Multi-scale Object Detection

Munther Abualkibash, Ahmed El-Sayed, Ausif Mahmood
Department of Computer Science and Engineering
University of Bridgeport, Bridgeport, CT
{mabualki, aelsayed, mahmood}@bridgeport.edu

Abstract

This poster presents a parallel implementation of an object detection algorithm, as well as an improved pruning technique, which is an important part in an object detection implementation. We focus on face detection, even though the techniques developed are applicable to object detection in general. We implement the well known face detection algorithm of Viola-Jones and parallelize the important steps, and then come up with a better pruning algorithm when many nearby windows indicate a detection. We also consider the effect of multiple scales and present a pruning algorithm that minimizes the detected windows such that a single window indicating the presence of a detected face can be concluded. Our pruning algorithm maximizes the face detection such that a few false positives may be detected, but all faces present are correctly identified.

For each scaled image, faces will be detected based on the best 200 features selected by AdaBoost, followed by our pruning technique.

Improved pruning algorithm

Detected face presented as many rectangles around the face. Our pruning algorithm eliminates duplicated rectangles around the face and reduce false positive results.

Figure 2. Original Image

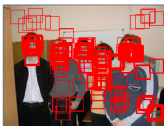


Figure 3. Detected faces for one scale

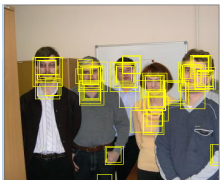
Pruning Algorithm

•Create a matrix that has same width and height as the image.
•When face is detected, store 1 else 0
• If one , and total count of one's in 8-neighbor > 4
•Keep it one, else change it to zero.

Multi-scale

- Each image scaled down between 5 and 10 times by10% each time.

- Combining the result:
  In each phase of scaling, some faces will be detected. After scaling the image down many times, all the result will be combined together.



ERROR: stackunderflow
OFFENDING COMMAND: begin

STACK: