Real Time Salient Object Detection

Team Details:

Nikhil Venkat(181IT224)
Naman Vijay(181IT129)
Ayush Rahangdale(181IT109)
Ankit Gupta(181IT107)

Introduction

The goal of salient object detection is to identify the most important objects in a scene. Recently, it has attracted much attention for its wide range of applications such as image retargeting, object recognition and image segmentation , to name a few. As a preprocessing step, a desirable saliency detection algorithm should be computational efficient for practical usage.

Concept

- A real-time salient object detection system based on the minimum spanning tree. Due to the fact that background regions are typically connected to the image boundaries, salient objects can be extracted by computing the distances to the boundaries.
- The minimum spanning tree representation of an image inherently reveals the object geometry information in a scene.
- This method largely reduces the search space of shortest paths, resulting an efficient and high quality distance transform algorithm.
- This is achieved by scanning the tree using bottom up and followed by top down approach.

Concept (contd.)

- Image I is treated as a standard 4-connected , undirected graph.
- Image pixels are treated as nodes of graph while edges between adjacent pixels is weighted by color/intensity differences.
- Minimum Spanning Tree is constructed by sequentially removing edges with large weights (Prims Algorithm).
- Numpy is used for Image normalization.
- Image width is a row in numpy matrix and image height is column.

Distance Transform with MST

‘d’ the barrier distance (BD) are used for salient object detection

d = $\max_{i=0}^{k} I(\pi(i)) - \min_{i=0}^{k} I(\pi(i))$

Let π = {π(0), ..., π(k)} denote a path on the graph. π(i) and π(i + 1) are adjacent pixels on image I. Given a distance metric f(π), the distance transform of a node v is

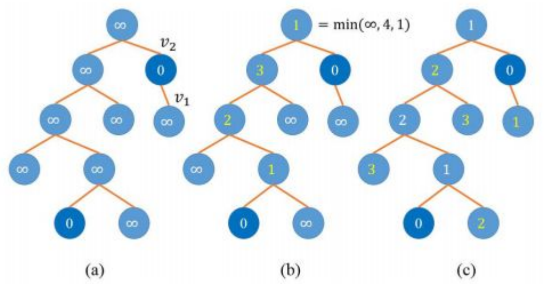
$$D(v) = \min_{\pi \in \Pi_{S,v}} f(\pi),$$

where $\Pi_{S,v}$ denotes the set of all paths connecting v and a seed in S.

Optimization

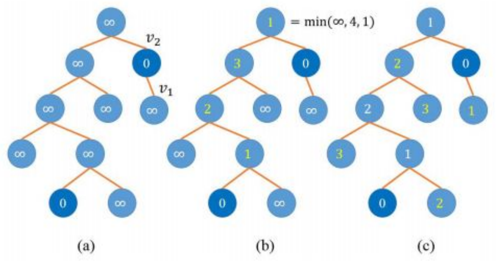
- Time complexity of finding distance metric for all node i.e storing max,min value with naive is $O(mn \log n)$ where n is the number of pixels in the image and m is the number of distinct values the image contains.
- Instead of searching the shortest distance on the image, we propose to find the shortest path on a minimum spanning tree. The proposed MST-based distance transform is an exact solution for distance defined on the tree.
- It consists of two passes: **bottom-up traversal** and **top-down traversal**. Given a set of seed nodes S, we initialize the distance values of seed nodes to 0 and all other nodes to ∞ . Top down traversal is followed by bottom-up traversal.
- This can be achieved in linear time by doing bfs over tree.

Bottom-up traversal



- Figure shows initial graph will all nodes marked as infinity.Do bottom up traversal start form leaves towards root.Mark distance level as infinity for others and 0 for all seeds nodes in image
- Starting from bottom to top mark distance label of parents as dist.child+1.This is done for all child and parent.dis=min of(child.dis+1) is taken. No need to change the label of seed node in image.
- Fig-b is illustrating bottom-up traversal for mst tree.
- Thus each node will contain 3 attributes i.e max distance along path,min distance and difference of both.(This is done with reference to seed nodes).

Top-down traversal



- After bottom-up traversal some nodes are still marked as infinity.
- After the bottom-up pass, a splitting node is expected to record the optimal distance value from one of its branches.
- In top-down pass, the optimal solution will be propagated down to other branches.

Results and Observations

Limitation

- The project is based on the concept that camera has to locate important objects of the image which are complete.
- If the salient objects touch the image boundary in the process causing image to be incomplete, the proposed method may produce bad results due to failure of algorithm near the boundary.

Salient object touching the boundary of the image

Correct input(Salient object is completely in the image)

Correct output

Summarizing steps:

1. Normalise the image as int(32) and store it in numpy. OpenCV library is used.
2. Run prim's algorithm on input image for discarding high intensity difference pixels.
3. After building a tree we find shortest and longest distance for a pixel relative to seed node.This is optimised by traversals for finding barrier distance.
4. Performing the bottom-up traversal and then the top down traversal results in the optimal distance transform.
5. If a seed node is the only seed node and is located at the root node of a sub-tree, then the distance transform of the nodes on the sub-tree is uniquely determined by this seed node. The corresponding distance transform will be obtained in the top-down pass.
6. No need to update seed node as everything is done relative to it.
7. Finally image numpy contains value according to seed threshold values which help in object detection.