Horizon Line Detection: Edge-less and Edge-based Methods

Touqeer Ahmad\(^1\), George Bebis\(^1\), Monica Nicolescu\(^1\), Emma Regentova\(^2\), Ara Nefian\(^3\) and Terry Fong\(^3\)

\(^1\)University of Nevada, Reno
\(^2\)University of Nevada, Las Vegas
\(^3\)NASA Ames Research Center
Outline

• Problem Definition
• Applications of a Detected Horizon Line
• Literature Survey
• Limitations of Earlier Methods
• Lie et al. : Prominent Edge based Method
• Proposed Approach 1 : Edge-Based HLD
• Proposed Approach 2 : Edge-Less HLD
• Experiments/Results
• Conclusion/Future Work
• Questions
Horizon Line Detection (HLD)

- Segmenting an Image into Sky and non-Sky (ground, water, mountains etc.) regions
- Challenging due to background clutter (e.g. clouds, fog etc.) and variation in non-sky regions
Applications

• Horizon line has previously been used for:
  – Ship detection [Felatyev et al., 2006]
  – UAV navigation [Ettinger et al., 2002] [Kim et al., 2011]
  – Visual geo-localization [Baatz et al., 2012]

Applications: Rover Localization

• Localization – estimating the 3D position and orientation with respect to some reference point

• Conventional localization techniques would not work in space because:
  – No GPS
  – Lack of good structural features/landmarks

• Horizon line can be used as a visual cue for rover localization.
Rover Localization (cont’d)

DEM

Where Am I?
Applications: Large Scale Visual Geo-Localization and Annotation

Image + 3D model = Registered image
Applications: Obstacle Avoidance & Smooth Navigation for UAVs/MAVs
Background

- Horizon line detection methods belong to two main categories:
  - **Edge-based methods**: group together horizon edge pixels (e.g., using Hough Transform [Zafarifar et al., 2008] Dynamic Programming [Lie et al., 2005])
  - **Region-based methods**: segment the image into sky and non-sky regions (e.g., using clustering [Boroujeni, 2012])

Background

- Felatyev et al.[1] – Machine Learning approach using texture features (intensity, entropy, smoothness, uniformity)
- McGee et al.[2] – SVMs to find linear boundaries
- Ettinger et al.[3] – Gaussian to model sky/non-sky and Bayesian estimation of optimum boundaries

Background (cont’d)

• Baatz et al.[1] – Visual geo-localization of images in mountainous terrain – matching extracted horizon from RGB to ones extracted from DEM of large scale

• Todorovic et al.[2] – Gaussian modeling of sky/non sky based on color (hue, intensity) and texture (complex wavelet transform) features

• Lie et al.[3] – dynamic programming approach using edges

Limitations

Based on Faulty Assumptions:

- Horizon Line is a linear boundary.
- Horizon lies in the upper half of the image.
- Sky and non-sky regions are equi-probable.
- There exists a specific light field above horizon.
- Non-sky regions have not much difference in texture/color (results in poor generalization)

Edge based method/s:

- Edge detection – inherently unstable
- Gaps due to edges being not strong or occluded e.g. by clouds
- Gap filling is not guaranteed and also linear
Lie et al.

- Prominent HLD approach based on Edges
- First to formulate HLD as a multi-stag graph problem and using DP
- Assumes
  - Horizon Line lies in the upper half of the image

Steps of Lie et al.

1. Apply edge detection on given query image

2. Formulate edge image as an MxN graph,
   \[ G(V, E, \Psi, \Phi) \] each pixel becomes a node,
   Initialize nodal costs
   \[
   \Psi(i, j) = \begin{cases} 
   l, & \text{if } I(x, y) = 1. \\ 
   \infty, & \text{if } I(x, y) = 0. 
   \end{cases} \quad (1)
   \]

3. A node in stage ‘j’ connected to few nodes in ‘j+1’ depending upon parameter delta,
   Initialize link costs
   \[
   \Phi(i, k, j) = \begin{cases} 
   |i - k|, & \text{if } I(i, j) = I(k, j + 1) = 1 \\
   \text{and } |i - k| \leq \delta \\
   \infty, & \text{otherwise.} 
   \end{cases} \quad (2)
   \]
Steps of Lie et al. (contd.)

4. Initialize node costs in stages 1 and N proportional to vertical positions

\[
\Psi(i, j) = \begin{cases} 
(i + 1)^2, & \text{if } j = 1 \text{ or } j = N \\
\Psi(i, j), & \text{otherwise.}
\end{cases}
\]  

5. Perform gap filling using high cost dummy nodes depending upon tolerance-of-gap parameter.

6. Add source (s) and destination (t) nodes, apply Dynamic Programming to find shortest path.
Lie et al. : Faulty Assumptions

parameters: $\delta$ and $\text{tog}$

$\delta$: – 1, $\text{tog}$: – 4
Proposed Approach 1: Edge-based HLD

• Extract most stable edges (i.e., Maximally Stable Extremal Edges or MSEEs).
• Filter out non-horizon pixels by applying a Support Vector Machine (SVM) which is trained using local features/combinations.
• Apply a Dynamic Programming (DP) shortest path algorithm to extract horizon line.
Ground Truth Labeling & Key Points Selection

• Manually labeled horizon locations for all images

• Key-points:
  – Positive: Uniformly chosen from horizon
  – Negative: Randomly chosen from non-horizon

![Image with labeled points]
Texture Features & Classifier

• Trained SVM classifiers
• Using local texture features and their combinations:
  – SIFT (128d)
  – LBP (58d)
  – HOG (31d)
  – SIFT+LBP (128+58)
  – SIFT+HOG (128+31)
  – LBP + HOG (58+31)
  – SIFT+LBP+HOG (128+58+51)
• 5-fold cross validation on Basalt Hills data set (45 images)
Edge-Based HLD : Training

- Training Images
  - Maximally Stable Extremal Edges Computation
    - MSEE
    - Negative Key Points: Randomly Sampled positions from MSEE
  - Positive Key Points: Sampled from Ground Truth
    - +ive Key Points
    - Generating Features around Key Points
      - Feature File
      - Appending +1/-1 Class Labels to Class Instances

- Input File
- Training the Classifier
- Trained Classifier

- Ground Truth Horizon Locations
Edge-Based HLD : Testing

1. Query Image
2. Maximally Stable Extremal Edges Computation
3. MSEE
4. Generating Features around Each Edge
5. Feature File
6. Discarding Negatively Classifier Edges
7. Class Labels
8. Classification Label for Each Instance
9. Trained Classifier
10. Dynamic Programming (Shortest Path)
11. Detected Horizon
Edge-based HLD
Proposed Approach 2: Edge-less HLD

1. Given query image, compute dense classifier (classification score for each pixel) score image (DCSI) using trained classifier (SVM/CNN)

\[ D(x, y) = \Gamma(V(x, y)) \]

2. Use DCSI to compute mDCSI

\[ mD(x, y) = \begin{cases} 
D(x, y), & \text{if } x \in L(m)_y \\
\ l, & \text{otherwise; } l < 0. 
\end{cases} \]

3. Formulate mDCSI as an MxN graph (No edge detection + parameters: no gap filling)

\[ \Psi(i, j) = mD(x, y) \]

\[ \Phi(i, k, j) = \begin{cases} 
0, & \text{if } |i - k| \leq \delta \\
\ \infty, & \text{otherwise.} 
\end{cases} \]

4. Apply DP to find the horizon line.
Edge-less HLD
Edge-less HLD
Experimental Details : Data Sets

Three data sets :
1. Basalt Hills – 45 images
2. Web – 80 images

Training :
• 9 images from Basalt Hills data set
• 343 +ive (uniformly from gt horizon) 343 – ive (randomly from non-horizon locations) key points

Resolution : 519x1388
Experimental Details: Features and Classifiers

Edge Based HLD:
Features:
• SIFT
• LBP
• HOG
• SIFT-LBP
• SIFT-HOG
• LBP-HOG
• SIFT-LBP-HOG
  – All in 16x16 image patches
Classifiers:
• SVM, with Linear Kernel

Test Sets:
• Basalt Hills, Web

Edge Less HLD:
Features:
• Normalized pixel intensities
  16x16 patches
Classifiers:
• SVM, with Polynomial Kernel
• CNN
  – 2 Covolutional(C)-Subsample(S) Layers
  – 1st CS Layer : 4 levels
    • C Mask 5x5, S Mask 2x2
  – 2nd CS Layer : 8 levels
    • C Mask 3x3, S Mask 2x2

Test Sets:
• Basalt Hills, Web
Experimental Details: Quantitative Analysis

Ground truth (GT) horizons are marked manually for all images.

HL detected by each formulation compared against GT: Average Absolute Error

\[ S = \frac{1}{N} \sum_{j=1}^{N} |P_{d(j)} - P_{g(j)}| \]
Results

- Comparing against Lie et al.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Basalt Hills</th>
<th>Web</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\mu$</td>
<td>$\sigma$</td>
</tr>
<tr>
<td>Lie et al. [4]</td>
<td>5.55</td>
<td>9.46</td>
</tr>
<tr>
<td>Gradient Info. [3]</td>
<td>3.99</td>
<td>6.35</td>
</tr>
<tr>
<td>SIFT+HOG Edges [3]</td>
<td>0.57</td>
<td>1.02</td>
</tr>
<tr>
<td>SIFT+HOG Scores [3]</td>
<td>0.41</td>
<td>0.81</td>
</tr>
<tr>
<td>SIFT+HOG Scores + Gradient [3]</td>
<td>0.43</td>
<td>0.81</td>
</tr>
<tr>
<td>Proposed : SVM-mDCSI</td>
<td>1.01</td>
<td>0.29</td>
</tr>
<tr>
<td>Proposed : CNN-mDCSI</td>
<td>0.75</td>
<td>0.23</td>
</tr>
<tr>
<td>SVM-mDCSI+Gradient</td>
<td>0.60</td>
<td>0.29</td>
</tr>
<tr>
<td>Fusion: SVM-DCSI+MSEE Edges</td>
<td>0.73</td>
<td>0.32</td>
</tr>
<tr>
<td>Fusion: SVM-DCSI+Canny Edges</td>
<td>0.77</td>
<td>0.35</td>
</tr>
</tbody>
</table>
Edge Based HLD: Some Visual Results
Edge Less HLD : Some Visual Results
HLD : Video
Lie et al. : Failure Examples

- Faulty Assumption : Horizon Lies in the upper half (introduced bias)
Lie et al. : Failure Examples

- Gaps due to edge detection
Proposed Approach 1: Failure Examples
Proposed Approach 2: Failure Examples
Proposed Approach: Reason of Failure

- Nodes in stage ‘j’ only allowed to be connected to nodes in ‘j+1’

- Very Small Train Set (9 images)
  - From a single data set, not many cloudy examples
  - Still very good generalization
  - Failed to find a good solution for only 9 out of 80 challenging images (web data set)
<Conclusion>

• Two machine learning based horizon line detection algorithms proposed
  – One relies on edges
  – Other solely on classification
  – A fusion strategy is proposed

• Both methods outperform the conventional edge based method

• Various texture features and nodal costs evaluated for Edge based method.

• SVM and CNN classifiers evaluated for Edge less method


<Future Work>

- Planetary Rover Localization
  - Estimating orientation/position of rover by matching rover image based horizon with synthetic horizon projected from DEM map
  - Using Particle Swarm Optimization and DCSIs to compute fitness for hypothesis synthetic horizons
  - Combining horizon based orientation estimates with SfM based estimates
<Acknowledgement>

This material is based upon work supported by NASA EPSCoR under Cooperative Agreement No. NNX11AM09A.
Questions/Suggestions

Thank you for Listening