Graph Based Image Segmentation and Layer Tracking in GPR Data

2016 09 07

Pete Dobbins
Joe Wilson
Outline

• Introduction
• Background
• Graph-Based (GB) Segmentation
• Ongoing Efforts
• Future Work
Introduction

• This work creates a scene model from ground penetrating radar (GPR) data.
• Data is collected by hand-held and vehicular-mounted detection systems.
• The goal is to identify within the model these key elements:
  – Ground layer
  – Sub-surface layers
  – Explosive (landmine) objects
  – Non-explosive (clutter) objects
Scene Frame

Introduction
A-scan (single channel voltage)

Introduction
B-scan (frame of channels)
C-scan (sequence of frames)
Data Representation

• *B-scans ( frames )* are made up of a series of *A-scans*.

• A sequence of *B-scans* is referred to as a *C-scan* and the complete scene we examine.

• Each *A-scan* is a reading collected by a GPR device. Relevant properties of the collected data are: *channel, depth, frame, and voltage*. 
Data Representation: C-scan View
Data Representation: C-scan View
Data Representation

**Channel:** the receiver channel on the collection device.

**Depth:** the time series resolved depth at which the voltage is found.

**Frame:** the frame in which the voltage is found.

**Voltage:** the energy signature collected by the receiver on the GPR device.
Data Representation

In our representation,

• the *depth* is the first dimension,
• the *channel* is the second dimension, and
• the *frame* is the third dimension.
RPC Layers

Previous work\textsuperscript{[1]} found one-dimensional layers in a two dimensional frame using Reciprocal Pointer Chains (RPC).
Tracking RPC Layers

The initial version used the Viterbi method to track these one-dimensional RPC layers across frames.
Tracking RPC Layers

This created a two-dimensional perspective of layers in the three-dimensional scene.
Tracking RPC Layers

Layers appeared as a partially connected wire-frames within the scene.
Limitations to this solution are:

- **Performance** – execution delays
- **Results** – lack of boundary analysis between layers limit results
- **Structure**
  - requires trellis (or grid)
  - trellis natural to vehicular data
  - hand-held sweeping will often be more dynamic, disjoint
Observations

• Frames of voltage returns in the GPR data can be viewed as an *image*.
• Many pixels of approximately the same value are seen, fitting into *equivalence classes*.
• *Background* and the *ground* layer constitute most of the image, when removed will help isolate objects of interest for examination.
• Image *segmentation* will label similar image elements.
Initial Segmentation

• Color-based segmentation using k-means clustering\textsuperscript{[2]}
• Cartoon model\textsuperscript{[3]}
Color-based segmentation using k-means clustering\cite{2}

- Luminosity, $a$, $b$ ($L^*a^*b^*$) color values
- K-Means clustering to label
- Complications
  - Execution time
  - Isolating elements
  - Tracking across frames
Cartoon Model$^3$

Gestalt$^{4,5}$ feature analysis for human visual perception

• Pixel intensity
  – voltage

• Pixel contour
  – Gabor Filter with 8 orientations

Execute MRF over features
Initial Cartoon Model Frame Sample
Initial MRF

The pattern of depth by channel by frame naturally lends itself to a grid and can easily be structured as a Markov Random Field (MRF). However, standard scene sizes will be on the order of 415 depths by 24 channels by 61 frames or 607,560 individual nodes in the graph. A network this size is computationally infeasible. Therefore, consider how to reduce the representation size of the MRF.
Super-Pixels and Super-Voxels

• Image *pixels* are the individual graphical elements in a two-dimensional image and in this context have value and contour properties.
• A *super-pixel* is a region of pixels that are closely related by some similarity measure.
• A *voxel* is a pixel in three-dimensional space.
• A *super-voxel* is a region of voxels that are closely related by a some similarity measure. [6,7,8,9]
MRF Graph Reduction Process

1. Identify regions
   - Each super-pixel still provides features: voltage, depth, channel, and frame
   - Regions provide statistics and global view
   - Efficient Graph-Based (GB) Image Segmentation[^8] identifies super-pixels
   - Efficient Hierarchical Graph Based Video Segmentation[^9] identifies super-voxels

2. Cluster related regions
   - Competitive Agglomeration[^10]

3. Execute MRF
   - Cartoon model[^3]
Efficient Graph-Based (GB) Image Segmentation\textsuperscript{[8]} Overview

Key points are to:

• form a graph of image,
• define a predicate, and
• use predicate to segment with global properties.
GB Overview

“...preserve detail in low-variability image regions while ignoring detail in high-variability regions.”[8]
Method Considerations

Capture perceptual groupings or regions
  – Consider global (non-local) characteristics
  – Specify technique and representation
  – Facilitate comparison to other techniques

Algorithm Efficiency
  – Approach linear time in number of pixels
  – Application to video processing
Method Details

- Pixels are nodes in graph
- Certain neighbors connected, based upon selected set of features
- Edges are weights / dissimilarity between pixels
Background Thoughts

Locality

– Clustering
– Normalized Cuts\textsuperscript{[11]}

Variation

– Gradient
– Intensity
– Uniformity
Locality: Normalized Cuts\textsuperscript{[11]}

- Cut connection to dissimilar regions
- Accounts for region similarity
- Captures non-local properties
- NP-hard problem
- Provides characterization, not final segmentation
Variation

*Constant, slow, or wide variation does not define region or separation.*
Variation

Cannot break graph edges only based upon variability

- Would merge to large blocks (too broad)
  - or -
- Would create lots of small blocks in high variation region (too specific)

Uniformity criteria subset does not work either

- Intensity
- Gradient
Intensity

Intensity variation alone does not determine region separation. This algorithm considers

• Element difference across region boundary
  – relative to –

• Element difference within the regions

A key observation is when cross-boundary intensity is relatively different than the within-region intensity of at least one of the two neighboring regions.
GB Setup

• Graph $G = (V, E)$
• Vertices
  – nodes
  – elements to segment
  – $v_i \in V$
GB Setup

• Edges
  – dissimilarity measure
  – between neighboring elements
  – \((v_i, v_j) \in E\)

• Dissimilarity weight
  – Intensity, color, motion, location, or other local attribute
  – \(w(v_i, v_j) > 0\)
GB Setup

• Segmentation $S$
  – partition of $V$
  – Into component (region) $C \in S$
  – corresponds to $G' = (V, E')$
  – where $E' \subseteq E$

• Edges in same component have low weight
• Edges in different components have high weight
Pairwise Region Comparison

If there is a boundary between two components (regions), define predicate D, and then

• Consider element dissimilarity along the boundary between the two components (cross-region differences)
  – relative to –

• Dissimilarity among neighboring elements within each of the two components (within-region differences)

Adaptive to characteristics of the data
Difference Between (Cross-Region)

Given $C_1, C_2 \subseteq V$ the minimum weight edge connecting $C_1$ and $C_2$ is:

$$
Dif (C_1, C_2) = \min_{v_1 \in C_1, v_2 \in C_2, (v_1, v_2) \in E} w((v_1, v_2))
$$

When $C_1$ and $C_2$ are not neighbors:

$$
Dif (C_1, C_2) = \infty
$$
Difference Between (Cross-Region)

Note, this only considers the smallest edge between the two regions. In practice authors found this was satisfactory.

Using the median weight makes finding a good segmentation NP-hard (discussed in Appendix).
Internal Difference (Within-Region)

Given $C \subseteq V$ and $MST(C, E)$ is the minimum spanning tree of the component,

$$\text{Int}(C) = \max_{e \in MST(C, E)} w(e)$$
Region Comparison Predicate

Where the minimum internal difference, $MInt$,

$$MInt(C_1, C_2) = \min(\text{Int}(C_1) + \tau(C_1), \text{Int}(C_2) + \tau(C_2))$$

When $|C| = 1$, $\text{Int}(C) = 0$, $\tau$ provides a threshold:

$$\tau(C) = k / |C|$$

The constant, $k$, provides scale and the larger the $k$ prefers larger components. Note, as $|C|$ grows, $k$ becomes less influential.
Therefore, for our predicate $D$,

$$D(C_1, C_2) = \begin{cases} 
\text{true} & \text{if } \text{Dif}(C_1, C_2) > \text{MInt}(C_1, C_2) \\
\text{false} & \text{otherwise}
\end{cases}$$
Segmentation Definitions

A segmentation $T$ is a *refinement* of a segmentation $S$ when

$$\forall i \ C_i \in T \text{ and } \exists j \ C_j \in S \ \exists \ C_i \subseteq C_j.$$ 

$T$ is a *proper* refinement of $S$ when $T \neq S$. 

Segmentation Components

A segmentation $S$ is *too fine* when there are too many components. $C_1, C_2 \in S$ and there is not boundary between $C_1$ and $C_2$.

A segmentation $S$ is too *coarse* when there are too few components. This is observed when segmentation $T$ can be found such that

- $T$ is a proper refinement of $S$
- $T$ is not too fine
Segmentation Algorithm

Input: \( G = (V, E); \; |V| = n; \; |E| = m. \)

Output: Segmentation of \( V \) into components \( S = (C_1, \ldots, C_r) \).
Segmentation Algorithm

0. Sort $E$ into $\pi = (o_1, \ldots, o_m)$ by non-decreasing weight where $o_q = (v_i, v_j)$.

1. Initial segmentation $S^0$ where $C_i = v_i$.

2. Repeat step 3 for $q = 1, \ldots, m$.

3. Given segmentation $S^{q-1}$,
   if $C_i \neq C_j$ and $w((v_i, v_j)) \leq \text{MInt}(C_i, C_j)$,
   obtain $S^q$ by merging $C_i$ and $C_j$
   otherwise $S^q = S^{q-1}$.
   When performing $C_k = C_i \cup C_j$, update the MST of $C_k$.

4. Return $S = S^m$.
Example Node Field
There is low dissimilarity (high similarity) between (1) the same color row and column and (2) the neighboring columns of the same color row.

\[ Row_2 Col_B \sim Row_2 Col_C \]

There is high dissimilarity between (3) columns on the same color row separated by another column.

\[ Row_2 Col_A \nmid Row_2 Col_C \]

There is high dissimilarity between (4) different color rows.

\[ Row_1 Col_A \nmid Row_2 Col_A \]
Legend Example / Rules

A solid line of a given color, blue / green / red, denotes within-region node connections of that color where the lowest dissimilarity is known.

A dashed line of a given color denotes ambiguity regarding which connection has the lowest dissimilarity, however the nodes are connected in the MST by some path.

A dotted black line denotes a cross region node connection.
Step 1: Segmentation $S^0$
Example Scenario

Consider four edges:  \( o_1, \ldots o_a, \ldots o_b, \ldots o_c, \ldots \)

These provide an example of the four options that can be observed as described on slide #49, briefly:

1. Same color
2. Same color neighbors
3. Same color non-neighbors
4. Different color
$S^0$ and Edge Scenarios

\[ C_1 \rightarrow v_1 \rightarrow o_1 \rightarrow v_2 \rightarrow o_1 \rightarrow C_2 \]

\[ C_2 \rightarrow o_1 \rightarrow v_2 \rightarrow o_1 \rightarrow C_3 \]

\[ C_3 \rightarrow o_c \rightarrow v_3 \rightarrow C_3 \]

\[ C_9 \rightarrow v_9 \rightarrow o_a \rightarrow v_{10} \rightarrow o_a \rightarrow C_{10} \]

\[ C_{10} \rightarrow o_b \rightarrow v_{10} \rightarrow o_b \rightarrow v_{11} \rightarrow o_c \rightarrow C_{11} \]
$S^1$ – Step 3: Rule #1
$S^1$ – Step 3: Merging

$C_1$

$C_3$

$C_9$

$C_{10}$

$C_{11}$
$S^a$ – Step 3: Rule #2

GB Segmentation: Algorithm
$S^a$ – Step 3: Merging

$\nu_1$ \hspace{1cm} $C_1$ \hspace{1cm} $\nu_3$

$\nu_9$ \hspace{1cm} $\nu_{10}$ \hspace{1cm} $C_9$

$\nu_{11}$ \hspace{1cm} $C_{11}$
$S^b$ – Step 3: Rule #3

![Graph Diagram]
$S^b$ – Step 3: No Merge

$C_1$

$v_1 - v_2 - v_3$

$C_9$

$v_9 - v_{10}$

$C_{11}$

$v_{11}$
$S^c$ – Step 3: Rule #4
$S^c$ – Step 3: No Merge

\begin{itemize}
  \item \textit{C}_1
  \item \textit{C}_9
  \item \textit{C}_{11}
\end{itemize}
Final Node Connections
Final Segmentation $S^m$

$C_1$

$C_9$

$C_{25}$

GB Segmentation: Algorithm
Each pixel, $p_i$, is a vertex or node, $v_i$. In two-dimensions, the 8-neighborhood of nodes is the set of pixels surrounding $p_i$. 

$\nu_{10}$ neighborhood
Grid Graph

Select an edge weight function.

- One example is intensity, where $I(p_i)$ is the intensity of pixel $p_i$.
- Then, the weight function, $w$, is:

$$w ((v_i, v_j)) = |I(p_i) - I(p_j)|$$
Grid Graph

• Initial structure for our testing
• Naturally fits vehicular-mounted data
• With adjustments, works with hand-held collected by robot
• Human operator data will be more dynamic and very difficult to fit into a grid graph
Human User Hand-Held Distinctions

The general procedure implemented is:

– operate device in sweep mode over large area,
– until a region of interest (ROI) is found,
– then perform investigation mode over ROI, and
– perform region processing (RP) over the more extensively investigated region.\textsuperscript{[12]}

The ROI will be centered on a specific object of interest, observed by the device operator when sweeping the area.
Sweep Mode

During *sweep* mode, movement of the device follows the pattern of:

- Sweeping from *left*-to-*right*
- Advancing the device forward
- Sweeping from *right*-to-*left*
Investigation Mode

During *investigation* mode, the device is swept more than twenty times over the ROI, including combinations of:

- Directly over the object of interest
- Nearby and approaching the object of interest
- Along the cross-track orientation
- Along the down-track orientation
Implications

Investigation mode is what I will spend my time examining. However, building a graph of points will be difficult

– Sweeps crossing over themselves
– Reading intervals will not form a consistent pattern
– Very unstructured graph

Solution: use Nearest Neighbor approach.
Nearest Neighbor

Select feature set and

• define spatial bound $b$: all points within $b$ are neighbors.
  – Or –

• define neighbor size $z$: closest $z$ points to point $p_i$ are neighbors of $p_i$.

Then, define a weight function for comparing regions, as in Grid Graph.
New Graph

• Regions are node set of super-voxels
• Edges connect neighboring regions
• Fits into the Nearest Neighbor GB view
• Need to incorporate the principles of GBH/video GB
Returning to the MRF

The MRF has been executed using GB super-pixel and EHGB super-voxel regions. These region features are provided to the MRF:

- Mean voltage
- Mean Gabor Filter
  - Over 6 / 16 orientations
  - From different perspectives
    - Down track (3 / 8 orientations)
    - Cross track (3 / 8 orientations)
    - Plan view (from above / below) – removed
Results: Region Display

- Super-Voxels
- CA Clusters
- MRF Regions
- Unconnected MRF Regions
- Frame Navigation
- Voltage
- Statistics
- 3D Element View

Ongoing Efforts
Results: Region Display

• Frame navigation GUI
• Label displays
  – Voltage B-scan
  – Super-Voxels
  – Competitive Agglomeration Clusters
  – MRF Regions
  – Unconnected MRF Regions
• Three-dimensional region viewing
• Process statistics
Frame #25 ( w/ Object )

Event #2
----------
BandWidth 1.50
Gamma 1
Lambda 8
Offset 2.25000e+01
Orientations 8
Psi 0  1.5708
Real & Imag

Ongoing Efforts
Frame #26 ( w/ Object )

Ongoing Efforts
Frame #27 (w/ Object)

Ongoing Efforts
Frame #28 ( w/ Object )
Frame #29 ( w/ Object )

---

Event #2
----------
BandWidth 1.50
Gamma 1
Lambda 8
Offset 2.25000e+01
Orientations 8
Psi 0 1.5708
Real & Imag

---

GBH -- 155
CA -- 5
MRF -- 3
Unconnected MRF -- 5

---

Ongoing Efforts
Frame #30 ( w/ Object )

Event #2
--------------
BandWidth 1.50
Gamma 1
Lambda 8
Offset 2.25000e+01
Orientations 8
Psi 0 1.5708
Real & Imag
Frame #31 ( w/ Object )

Ongoing Efforts
Results: 3D Element View
Results: 3D Element View
Results: 3D Element View
Results: 3D Element View (default)
Results: 3D Element View (zoom)
Results: 3D Element View
Results: Truth Tool

Voltage

Different Label Sets

Unconnected MRF Regions

Click within Unconnected Regions to Create or re-assign labels

Frame Navigation
Results: Truth Tool

Current Unconnected MRF Label

Enter an existing Unconnected MRF region label – or – a new region label number

Super-Voxel Region Label

Select Label

current label: 30
current region: 293963

OK Cancel
Results: Truth Tool

• Frame navigation GUI
• Displays
  – Voltage B-scan
  – Unconnected MRF regions
  – Labels used for each model
• Interactive region re-labeling
Future Work

• Implement new graph structure with GB Nearest Neighbor methodology
• Create a Dirichlet distribution\cite{13} over region features
• Perform Semi-Supervised Learning\cite{14} on classification
  – Using truth tool to label
Conclusion\textsuperscript{[15]}

• I have done some really good and important work.
• There is more to be done.
• I know how to do it.
References


