Accelerating Satellite Image Based Large-Scale Settlement Detection with GPU

Dilip R. Patlolla
Anil M. Cheriyadat
Eddie A. Bright
Jeanette E. Weaver

Oak Ridge National Laboratory
Oak Ridge, TN

GPU Technology Conference 2013
San Jose, CA
• Advanced geospatial models built on settlement characteristics such as size, shape, location, growth can reveal social stability and political durability.

• Automated settlement detection process involves extracting, representing, and identifying patterns characterizing settlements in image data.

• Strong demand to understand the settlement characteristics from aerial imagery.
Problem

- Land surface ~150 Million km² translates into roughly 600 Trillion pixels to be processed with 0.5 m imagery.
- Pixel to feature descriptor mapping is an expensive, but key step in automated scene analysis.
- Application of settlement detection algorithms on large image databases can easily saturate the processing capabilities of conventional CPUs.
- Parallel feature extraction using Single Instruction Multiple Threads (SIMT) architecture of GPU.
- Efficient GPU implementation of multiscale features.
- **Exploitation of multi-GPU architecture** for large-scale image analysis.
- Reduce redundant data transfers and computations.
Features are computed for each scale:
\[ d^s = \{ h_{i,j}^s, c_{i,j}^s, g_{i,j}^s ; s = 0,1, \ldots N \} \]

\( N \) is the total number of scales.
\( i,j \) represent the index of the pixel blocks.

For each pixel-block at every scale, we reduce the high-dimensional feature vectors to a scalar value by projecting the feature vector onto the SVM decision plane,
\[ z^s = (W^T)^s d^s \]

SVM decision function is given by
\[ f(z) = \sum_{s=0}^{N} w^s + b \]

At each pixel-block \((i,j)\) the SVM decision function simply becomes \( f(z_{i,j}) > 0 \) for settlement.
\( W \) and \( b \) are SVM model parameters.
GPU Based Vision Systems

- Cornelius et al. - Scale invariant feature descriptor generation and matching on a GPU
- Bernabe et al. - Satellite image classification on GPU, but limited to pre-processing filtering, post-processing
- GPU based HOG for Pedestrian Detection
- Parallel GLCM by shahbahrami et al. but targeted to a Cell processor

Significant speedups have been achieved in all cases
Computer Vision Challenges

- With high-resolution imagery, **pixel features** are not discriminatory.
- Multi-scale features provide accurate settlement characterization.
- We aggregate three low level features from 5 spatial scales.
  - Histogram of Gradient Orientations (HOG): $h$
  - Local pixel intensity statistics (Spectral): $c$
  - Gray-Level Co-Occurrence Matrix (GLCM): $g$
- A linear support vector machine (SVM) classifier is used to map the feature vector ($z$) to one of the binary classes (settlement and non-settlement).
Settlement Mapping Process

Features are computed for each scale: 
\[ d^s = \{ h_{i,j}^s, c_{i,j}^s, g_{i,j}^s ; s = 0,1,\ldots,N \} \]

\( N \) is the total number of scales. 
\( i,j \) represent the index of the pixel blocks

For each pixel-block at every scale, we reduce the high-dimensional feature vectors to a scalar value by projecting the feature vector onto the SVM decision plane,

\[
z^s = (W^T)^s d^s
\]

SVM decision function is given by 

\[
f(x) = \sum_{s=0}^{N} w^s + b
\]

At each pixel-block \((i,j)\) the SVM decision function simply becomes \( f(z_{i,j}) > 0 \) for settlement. 
\( W \) and \( b \) are SVM model parameters
M is the pixel Block size.

\[
\text{for } j = 1 \text{ to } M \text{ do} \\
\quad \text{index} = (j, \text{baseidx}, M) \\
\quad \text{OA}_{\text{bin}} = \text{OA}[\text{index}] \\
\quad \text{HIST}_{\text{shared}}[\text{OA}_{\text{bin}}] += \text{GR}[\text{index}] \\
\text{end for} \\
\text{Sync} \\
\text{for } k = 1 \text{ to } \text{Size(HIST)}/M \text{ do} \\
\quad \text{stride} = M \\
\quad \text{Update HIST}_{\text{global}}[\text{baseidx}; k; \text{stride}] \\
\text{end for} \\
\text{Sync}
\]
MULTISCALE HOG

- 64 threads compute the mean of the base scale histograms
- No Memory Conflicts
- Reduced redundancy
GLCM

N threads working
NxN quantized pixel block
In Shared Memory

\[ con(c) = \sum_{m=1}^{Q} \sum_{n=1}^{Q} (m - n)^2 P(c)_{m,n} \]

\[ g_{i,j}^{0} = con(k)(c(1)_{i,j}^{0})^2 \]

P is the normalized co-occurrence matrix.
Q represents the number of quantized gray levels.
c(1)_{i,j}^{0} represents the mean pixel intensity for the pixel-block at (i; j)
and k = \text{arg min}_k con(c).
Hardware

Xeon + Tesla 4U Server

• CPU
  - 2 Intel Xeon E5620 Westmere
  - 2.4 GHz Quad Core
  - 48 GB DDR3 Memory

• GPU
  - 4 Nvidia Tesla C2075
  - 448 CUDA cores
  - 6 GB GDDR5 Memory
Validation

• Settlement Detection Model learned from a dataset consisting of 60 scenes
• Each Scene
  - represents settlement structures from different parts of the globe
  - Size: 2048 x 2048 at 0.6 m spatial resolution
• Tested on a dataset consisting of 55 scenes with 86% accuracy
• Results validated with CPU implementation.

Few scenes
ACCURACY

Overall detection accuracy under different pixel-block size and scale settings.

Contribution of features and different scale settings.
PROCESSING TIME

FEATURE RUNTIME

Runtime occupation percentages for various operations during the settlement extraction processing time for various features at varying block sizes.

- GLCM
- HoG & Spectral

Tiles Size:
- 8x8
- 16x16
- 24x24
- 32x32

Processing Time (ms)
SPEEDUP

- Image Normalization: 45x
- GLCM: 15x
- HoG and Spectral: 200x

High no. of computations per transferred data
Memory Conflicts
Less Computation per data transferred
LAHORE SETTLEMENT EXTRACTION

Data
- 0.6 m spatial resolution
- RGB bands processed

Processed 1200 sq. km in 56 sec

Wall to Wall processing

Non-settlement accuracy (class 0) is 96.7281%
Settlement accuracy (class 1) is 82.3651%
Overall Accuracy = 0.93

Predicted

<table>
<thead>
<tr>
<th>Actual</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>71869</td>
<td>2431</td>
</tr>
<tr>
<td>1</td>
<td>4535</td>
<td>21181</td>
</tr>
</tbody>
</table>

Class          Recall  Precision  Harmonic Mean
0              0.97      0.94      0.95
1              0.82      0.90      0.86
• Increase the accuracy.
  Speedup of separable filter response GPU implementation over non separable filter response \(~40\times\)

• So what about non separable filters????
  Solution : rotate the image at the cost of extra processing time but still get a speedup of \(~30\times\)
  over non separable

MPI+GPU implementation to tackle country scale imagery on a cluster - already in place. 😊
THANK YOU