Scalable Image Graph Matching and Analysis: Constructing a 3-D Model from 2-D EO-Imagery Collections

Karl Ni, Zachary Sun, and Nadya Bliss
karl.ni@ll.mit.edu, zsun86@ll.mit.edu, nt@ll.mit.edu
MIT Lincoln Laboratory, 244 Wood St., Lexington, MA 02420-9108

Introduction

Very often, high-valued individuals or assets are taken in digital photos, either intentionally or unintentionally. In such situations, meta-data (including GPS coordinates, annotated landmarks, potential threats, etc.) is often unavailable or inaccurate. Because any individual image may contain considerable amounts of information, the ability to understand and extract 3-dimensional scene information would be advantageous. We present a system and its related methodologies to promote situational awareness given single EO-image, without meta-data or any other media and modalities.

The proposed approach relies on several advances in computer vision that have been made over the past ten years. Specifically, object detection techniques and image registration techniques form the foundation of our system architecture. We modify and improve upon conventional approaches in both problem spaces, while fitting them to a proposed framework. This framework is presented with results that show the capability to detect targets in an image and geo-register them to an accuracy within a few meters.

The framework involves complicated training and setup procedures in order to promote real-time exploitation. Obviously, for generalization purposes, the required data set is an extensive collection of tens of thousands of images, all at high-resolution, and enough resources must be available to support such a training set.

While object detection training procedures can be completed within a few weeks serially, image registration procedures (whose ultimate goal is to create a 3-D point cloud) require several modularized tasks that could individually consume unwieldy runtimes. Without optimization, the potential run-time without 2713 images could take up to 715 hours…a month's worth of processing time, not to mention memory and storage constraints.

Fortunately, much of the training and setup can be parallelized, and the resultant distributed processing time can be reduced to 8 or 9 hours. The remainder of this abstract describes the computational methods in achieving a setup that is efficient, accurate, and quick. We will also discuss the exploitation of the generated product and its performance.

This work is sponsored by the Department of the Air Force under Air Force contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government

Processing Overview

The 3-D model consists of features taken from a 2-D set of images that overlap spatially in content. The final features are arranged in such a way that the 3-D point coordinates describe the originating structure of the content in the 2-D images. It is a collective data filtering process that simultaneously solves for scene structure while determining camera parameters for each image. The implementation is primarily taken from Noah Snavely's Photo Tourism [1].

Figure 1: Conceptual diagram of system.

There are three major tasks to building the 3-D model: extracting features from images, finding correspondences across images, and then finding the structure from motion based on the matching information. Specifically, we label these modules:

1. SIFT Feature Extraction [2]
2. Approximate Nearest Neighbor Matching [3]
3. Structure from Motion with Bundle Adjustment

The feature extraction labeled as step 1 is a keypoint detector that is shift and rotation invariant feature, which is somewhat robust to changes in luminance (image brightness). Specifically, it is called the Shift Invariant Feature Transform (SIFT) [2]. Not only does SIFT return a list of keypoint locations within an image, but it also gives a *unique* n-dimensional descriptor vector that can be used for matching. Conventional demonstrations have concluded that by using 128-dimensional vectors, the best balance between speed and performance can be achieved while reliably matching images.

Once features have been extracted, correspondences need to be established in step 2. For each pair of images, keypoints are matched by finding the nearest neighbor vector in the corresponding image, which is traditionally defined in Euclidean L-2 space. To speed up matching, Arya and Mount's approximate nearest neighbor package (ANN) [3] can be exploited. For image I and J, ANN builds a *kd*-tree of features in image J and then queries the tree for the best match of each feature in image I. Instead of defining a valid match by thresholding the distance, valid matches are determined using a ratio test. This test is defined by finding the best two nearest neighbor in image I with distances d_1 and d_2 where $d_1 < d_2$. Accept this features as a match if $d_1/d_2 < th$.

Finally, structure from motion (SfM) contains intricate optimization that makes it difficult to extricate parallelization techniques (though there is considerable potential). Part of the structure from motion process requires a process called bundle adjustment using a sparse bundle adjustment (SBA) package based on solving a numerical minimization problem using the Levenberg-Marquadt algorithm.

Optimization Overview

For typical 8MP images, SIFT returns, on average nine thousand features. Extracting features on average, 8MP images takes, on average, 40 seconds on a Pentium 4 processor. Typical data sets include thousands of images. This translates to roughly 11 hours of computation time.

Data Type	Number of Points	% of Original
Raw Pixels	9,563,111,424 pixels	100%
SIFT	12,976,125 features	0.1357%
3-D SIFT	83,796 features	0.0000876%

Table 1: LL-Grid Resources

Among 128 computing nodes, extracting SIFT features in parallel (as images in this step can be independently processed), the computation time for 1012 images runs in under five minutes. Of course, this scales linearly with the number of images to be added.

While it may seem unwieldy to match SIFT features, operating on entire images is nearly impossible, as 10 Gigapixels equates to roughly 77GB of uncompressed data. Extracting SIFT features over a single image pares the information to less than a percentage of *relevant* and *salient* data points. Moreover, because images share features among each other, there is no need to keep redundant features corresponding to the same 3-D geo-coordinates. That is, a representative 3-D SIFT feature can describe a unique geo-position shared across several 2-D images. Table 1 describes the data compression savings.

Finding these matched features among 2-D images is another task that stands to gain from parallelization. To match every image to every other image requires $n(n+1)/2$ image matches, where n is the number of images in the training set. The total matching time for a single processor on 1012 images would take roughly 256 hours. Among 128 nodes, this, of course, would be reduced to two hours.

LL-Grid Processing

The overall performance results are given in Figure 2. We can also build an understanding of the proportion of time used for each of the three tasks. The non-parallelized portion of the code cannot be reduced in time, which is why, for the most part, they consume the largest proportion. Of course, the matching and reconstruction times are highly dependent on the image content.

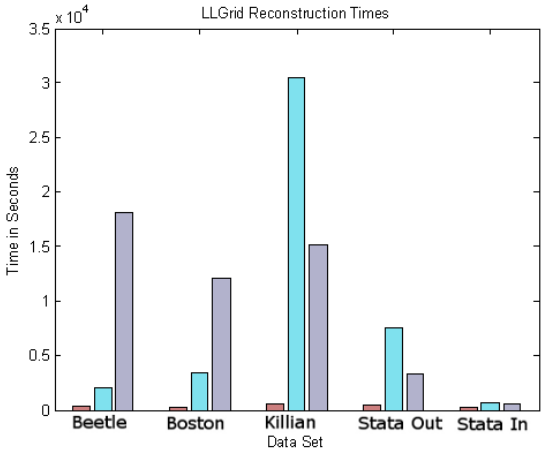

Figure 2: Performance results.

Summary and Future Work

We have described the implementation issues in constructing a 3-D static model using SIFT features. Without optimization and parallelization, SIFT extraction, matching, and bundle adjustment will render the model creating intractable. By running scripts to compute in parallel, the overall runtime for our set of images is acceptable.

Because it solves a single problem involving all data simultaneously, SfM remains the computational bottleneck of the system architecture.

References and Acknowledgements

We would like to thank Noah Snavely from Cornell University for all his guidance and aid in the construction of the 3-dimensional scene.

[1] N. Snavely, S. M. Seitz, and R. Szeliski, “Photo Tourism: Exploring photo collections in 3-D,” In the Proceedings of *SIGGRAPH*, pp. 835-846, New York, NY, USA, 2006. ACM Press

[2] D. G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” *International Journal of Computer Vision*, Vol. 60: pp 91-110, 2004.

[3] S. Arya, D. Mount, N. Netanyahu, R. Silverman, and A. Wu, “An Optimal Algorithm for Approximate Nearest Neighbor Searching in Fixed Dimensions,” In *ACM-SIAM Symposium on Discrete Algorithms*, pp. 573-582, 1984

[4] M. Lourakis and Argyros, “SBA: A Software Package for Generic Sparse Bundle Adjustment,” ACM Transactions on Mathematics Software

[5] K. Ni, Z. Sun, N. Bliss, and N. Snavely, “Construction and Exploitation of a 3-D Model from 2-D Image Features,” In the Proceedings of the *SPIE International Conference on Imaging*, Inverse Problems Session, SPIE-2010, Vol. 7533, San Jose, CA, U.S.A., January 2010

[6] Z. Sun, K. Ni, and N. Bliss, “A 3-D Feature Model for Image Matching”, In the Proceedings of the *IEEE International Conference on Acoustics, Speech, and Signal Processing*, ICASSP-2010, Dallas, TX, U.S.A., March 2010