Saliency Based Image Compression

Salem Cherenet

Department of Mechanical Engineering, College of Engineering, Carnegie Mellon University

Advised by David Wettergreen, Tai Sing Lee

Introduction

In areas that are unsafe for humans to directly study, robotic exploration is an excellent method for collecting and interpreting data. Robots can be used for exploration; but, they require operator input about what tasks to accomplish. In order for operators to plan out optimal tasking, information about the state of the rover is required. Panoramic imaging of the rover's last position gives insights into what tasks to accomplish next; however, each panorama is made up of over a hundred high resolution images; this makes sharing and interpreting information hard due to bandwidth limit and information overload. This paper introduces a saliency based image compression method where regions of panoramas containing informative information were given high resolution and the rest of the regions were sampled down using deep auto-encoder and TextureCam classifier.

Keywords and Definitions

Saliency Map: is a topographically arranged map that represents visual saliency of a corresponding visual scene. [6]

Information Overload: difficulty to make decision due to too much information [7].

Patch: a single image that makes up a panorama

Gaussian Pyramid: is a method of reducing image size by half iteratively by applying a Gaussian filter in each step. In a Gaussian pyramid, the base level is the original image

Laplacian Image: is the difference between the preceding level image and the current level image in Gaussian pyramid. (see Figure 1)

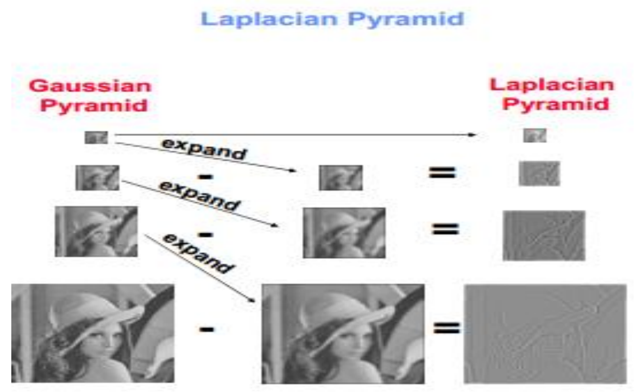


Figure 1: Laplacian and Gaussian pyramid
The bottom layer is the original image. The next layer image is found by convolving the original image with a Gaussian and resizing it to half it's original size. The difference between the original image and the convolved image is the Laplacian (the information lost due to down sampling). Therefore to get the original image the corresponding Laplacian image need to be added at each layer of the pyramid [5].

Method

Step 1:

Global Classification and Global Saliency Map

We defined global classification as classification over the whole panorama. This was done using TextureCam [2].

TABLE 1: List of Target classes and their approximated values based on observations during the LITA filed tests [1].

Target Class (A_k) Value		Relative Importance	Global Classification	Local Classification
0	Sun	1	9 – Spring Green	1
1	Clouds	0.8	7 - Brown	1
2	mountain hills	0.9	8 - Purple	4
3	Slopes/Drop-offs	0.5	5 - Yellow	3
4	Drainages/Channels	0.75	6 – Orange	3
5	Rocks and Sediment	0.3	4 - Blue	
6	Rover Parts	NA	2 - Red	2
7	Multiple Classes	NA	1 – White	2
8	Solid Color	NA	0 -Black	2
9	Sky	NA	3 – Green	1

Figure 2: Typical input panoramic image

Figure 3: Label Panorama for Figure 1

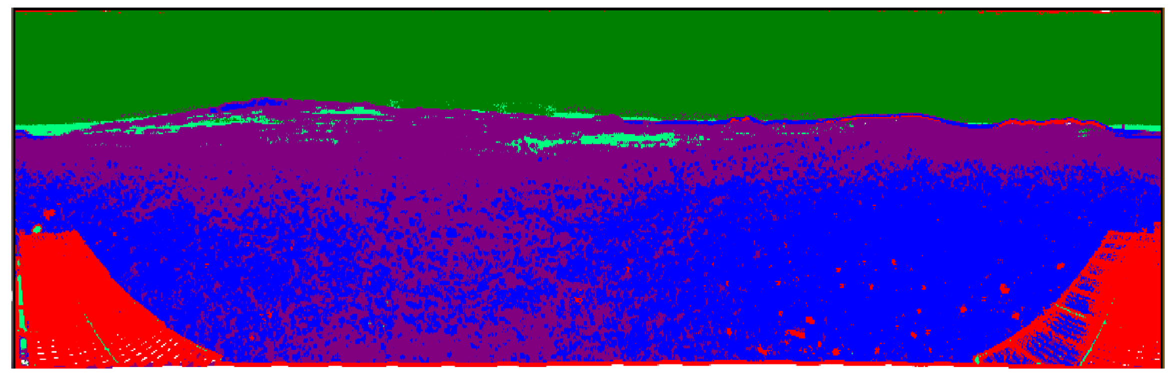
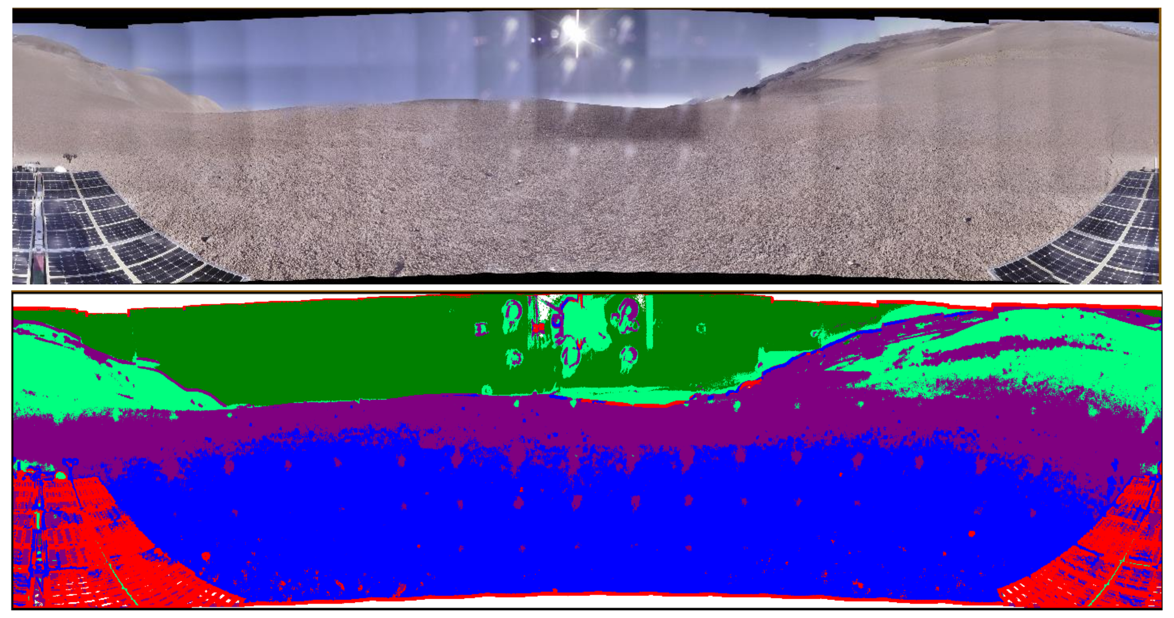


Figure 4: TextureCam take in a panoramic image (Figure 2) and a label image (Figure 3) for training and generates a classification. This image shows a typical classification output.



Methods Continued

Step 2:

Local Classification and Local Saliency Map

We defined local classification as classification over a single patch. This was done in four steps.

- **Image database creation**

Each patch was rotated (-90 to 90 degrees) and scaled (1.4 to 2 times) to generate scale and orientation invariant data.

- **Deep Auto-Encoder Feature Extraction**

~54000 images were used for training and testing

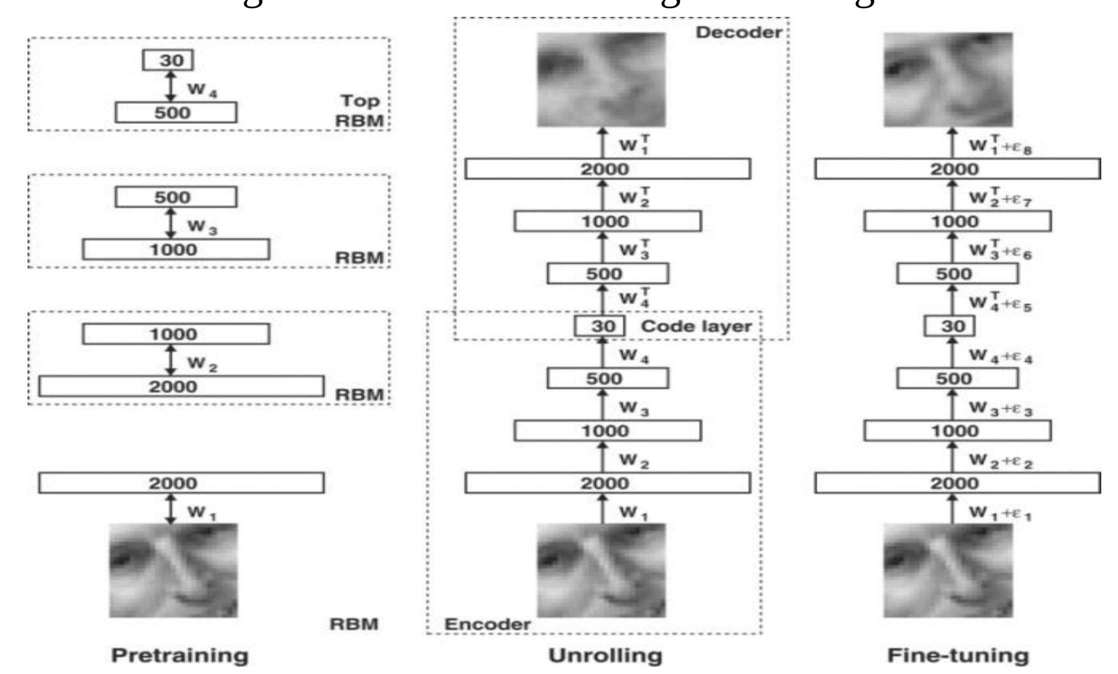


Figure 6: Pre-training consists of learning a stack of restricted Boltzmann machines (RBMs), each having only one layer of feature detectors. The learned feature activations of one RBM are used as the "data" for training the next RBM in the stack. After the pre-training, the RBMs are "unrolled" to create a deep auto-encoder, which is then fine-tuned using backpropagation of error derivatives [3]

The deep auto-encoder outputs 9, 32 by 32 feature images at the code layer for each class.

- **Support Vector Machine (SVM) Classification**

The code layer features from the deep auto-encoder were used to classify the feature images of unseen input data into the four different classes (Table 1 column5).

- **Resolution Adjustment**

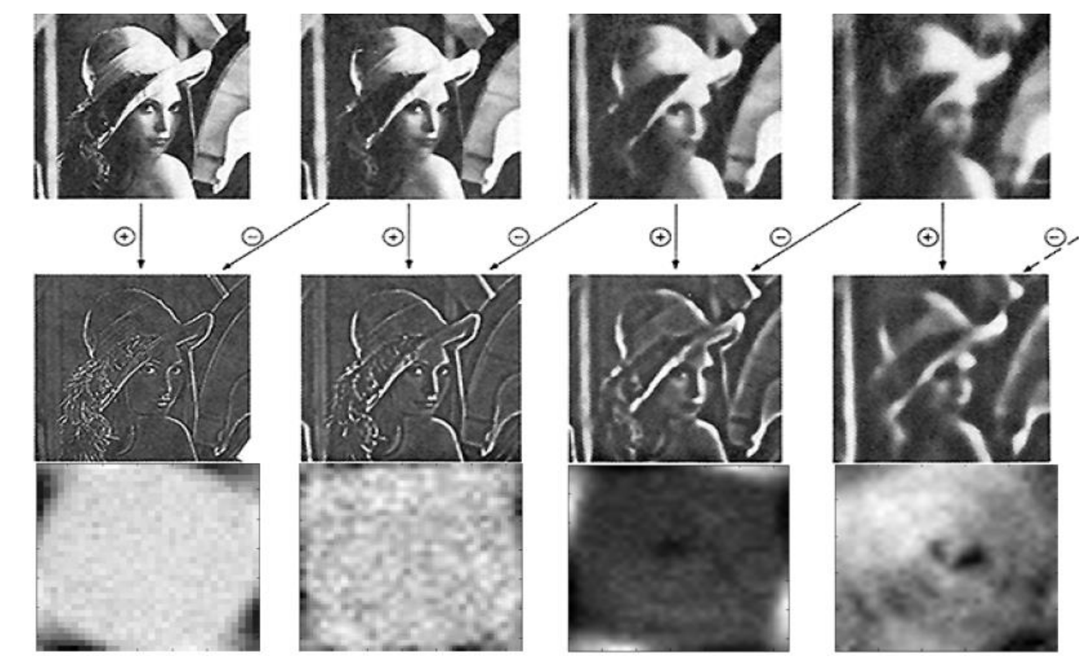


Figure 7: Resolution Adjustment process

Results

TextureCam Accuracy

TextureCam had a classification accuracy of 82% on unseen panorama.

Deep Auto-Encoder Reconstruction Accuracy

The robustness of the deep auto-encoder is measured by its reconstruction accuracy. The highest reconstruction squared error was 11.7 and the lowest was 1.552.

SVM Performance

Performing less than 50% during testing.

Image File Size Reduction

The panorama size was decreased by more than 90%. Work in progress also shows further reduction can be achieved by applying local saliency map.

Conclusion and Future Work

The main goal of this project was to reduce panoramic image's file size, and create local and global saliency maps. Our method enabled us to reduce panoramic image's file size by more than 90%, and apply saliency maps to serve as visual cues.

Global Saliency map accuracy could be increased by increasing learning window size, having better labeling technique, and changing image orientations during training.

Local Saliency map is still work in progress and further investigation needs to be done.

Acknowledgments

I would like to give special thanks to David Wettergreen, Tai Sing Lee, David Thompson , and Michael Furlong for being instrumental in giving me direction throughout this project.

I would also like to thank the GEM Fellowship program, SLAC National Accelerator Lab, and Carnegie Mellon Mechanical Engineering department for sponsoring my graduate study.

Reference:

[1] Glasgow, J., G. Thomas, E. Pudenz, N. Cabrol, D. Wettergreen, P. Coppin (2008), "Optimizing Information Value: Improving Rover Sensor Data Collection, IEEE Trans. Systems, Man and Cybernetics, Part A: Systems and Humans, 38(3).
[2] K. L. Wagstaff, D. R. Thompson, et al., "Smart Cameras for Remote Science Survey". Unpublished manuscript.
[3] Hinton, G. E., Osindero, S. and Teh, Y. (2006)
A fast learning algorithm for deep belief nets.
Neural Computation, 18, pp 1527-1554.
[4] Bertin, Emanuel. *Sextractor 1.0a User's Guide*. Paris: Institut D'Astrophysique De Paris. Pdf.
[5] Hel-Or, Hagit. "Image Pyramid." N.p., 21 Dec. 2010. Web. 10 Apr. 2013.
[6] Niebur, Ernst. "Saliency Map." *Saliency Map- Scholarpedia*. Scholarpedia, 2007. Web. 1 Apr. 2013.
[7] "Information Overload." *Wikipedia*. Wikimedia Foundation, 31 Mar. 2013. Web. 17 Apr. 2013.

THE NATIONAL GEM CONSORTIUM

NATIONAL ACCELERATOR LABORATORY

Mechanical Engineering
Carnegie Mellon University