

# Genetic Algorithms for Action Set Selection Across Domains: A Demonstration

Greg Lee and Vadim Bulitko  
Department of Computing Science  
University of Alberta  
Edmonton, Alberta, Canada, T6G 2E8  
{greglee | bulitko}@cs.ualberta.ca

## ABSTRACT

Action set selection in Markov Decision Processes (MDPs) is an area of research that has received little attention. On the other hand, the set of actions available to an MDP agent can have a significant impact on the ability of the agent to gain optimal rewards. Last year at GECCO'05, the first automated action set selection tool powered by genetic algorithms was presented. The demonstration of its capabilities, though intriguing, was limited to a single domain. In this paper, we apply the tool to a more challenging problem of oil sand image interpretation. In the new experiments, genetic algorithms evolved a compact high-performance set of image processing operators, decreasing interpretation time by 98% while improving image interpretation accuracy by 55%. These results exceed the original performance and suggest certain cross-domain portability of the approach.

## Categories and Subject Descriptors

I.4.8 [Computing Methodologies]: Artificial Intelligence—*Learning*

## General Terms

Performance

## Keywords

genetic algorithms, machine learning, heuristic search, Markov decision process, adaptive image interpretation

## 1. INTRODUCTION

Many real-world problems involve the need for image data analysis, with massive amounts of data to be processed in a limited amount of time. Information extraction from visual data is a challenging and time-consuming task. One example lies with oil sand mining operation which involves crushing collected ore lumps as a pre-processing step to oil

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO'06, July 8–12, 2006, Seattle, Washington, USA.  
Copyright 2006 ACM 1-59593-186-4/06/0007 ...\$5.00.

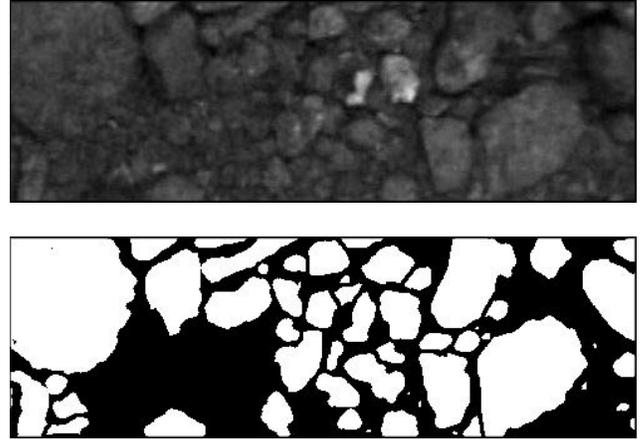


Figure 1: The oil sand ore segmentation task. Ore lumps are shown on a conveyor belt (top), manually segmented (bottom).

extraction. Size distribution of crushed ore fragments is a critical indicator of pre-processing quality and a key to oil extraction efficiency.

Size analysis techniques do exist (centrifugation, mechanical sieving, sedimentation), but these methods typically involve physical manipulation of the actual materials being measured, whereas computer vision techniques operate with real-time visual feed from a video camera inspecting ore coming out of a crusher. The usual procedure is to segment images of ore fragments on the conveyor belt (e.g., Figure 1), and then measure sizes of the segmented areas.

In order to segment the oil sand ore *automatically*, an Ore Size Analyst (OSA) system was created [20]. This system used a fixed hand-selected set of vision operators, each with a manually tuned fixed set of parameters. Not only was the trial-and-error tuning process laborious and expensive, but also the resulting performance did not generalize well onto a broad spectrum of field conditions.

A new generation of image interpretation systems utilizes automated machine learning techniques to attain a robust level of performance in the field [3]. They learn adaptive control policies to handle large variations in the input data. The policies are reinforcement-learned [17] over an extensive generic set of image processing operators since image interpretation is treated as a Markov Decision Process (MDP) [1]. Reinforcement Learning (RL) is commonly used within MDP settings wherein an agent is interacting with

its environment, developing a world model and receiving rewards based on the results of its actions. In these settings, the agent learns a control policy that tells it how to act in the environment in an attempt to maximize its cumulative reward. In the vision domain, actions are vision operators that manipulate the image in the process of producing an interpretation. The rewards are defined as accuracies of the image interpretations produced.

An objective of Reinforcement Learning is fully autonomous learning by the agent. One of the last vestiges of human intervention is action set selection. Action set selection is important because a large set of actions can confuse the agent and slow down its learning. On the other hand, a small set of actions can provide the agent with too few options to act optimally in the environment.

The number of possible action sets usually grows exponentially with the number of distinct actions. In the past, researchers were limited to selecting action sets manually through a laborious process of trial and error. Last year at GECCO'05 the first Genetic Algorithms powered tool, called GAMM, for *automated* selection of action sets for a reinforcement learning agent was presented [9]. It was applied to an existing reinforcement learning vision system called MLCV (Machine Learning for Computer Vision) [10] in the domain of automated forestry image interpretation. GAMM's application to automated action set selection resulted in 95% reduction of the system's running time while preserving the interpretation accuracy of the original action set. This demonstration, though impressive, was limited to only a single domain and left GAMM's scalability and cross-domain portability as open questions.

The primary contribution of this paper is a demonstration of cross-domain portability of GAMM. We took the tool directly from the forestry domain [9] and applied it to a substantially different and arguably more challenging problem of image interpretation of oil sand ore fragments [20]. In the new domain GAMM exceeded its performance in the original application and reduced the running time of the machine learning computer vision system by 98% while improving its image interpretation accuracy by 55%.

We attempted to make the paper as self-contained as possible and organized it as follows. We first describe the oil sand ore image domain in section 2. Next, in section 3 we outline previous research in image interpretation, before describing the MLCV system used to interpret oil sand ore images in section 4. This is followed by a description of the GAMM tool in section 5. After overviewing related research in feature selection in section 6, we present empirical results in section 7. Concluding remarks are provided in section 8.

## 2. OIL SAND ORE IMAGE INTERPRETATION

Proper measurement of crushed ore fragments leads to improved performance of the subsequent oil extraction process, as adjustments can be made to machinery based upon the size distribution of the ore. An example of a segmented oil sand ore image is shown in Figure 1.

Ore segmentation is an outdoor year-around 24-hour operation. This adds numerous complications to the image segmentation process, since the ore may be covered in snow, wet from rain, and lit by natural or artificial light. In addition to these obstacles, the ore itself may be of different

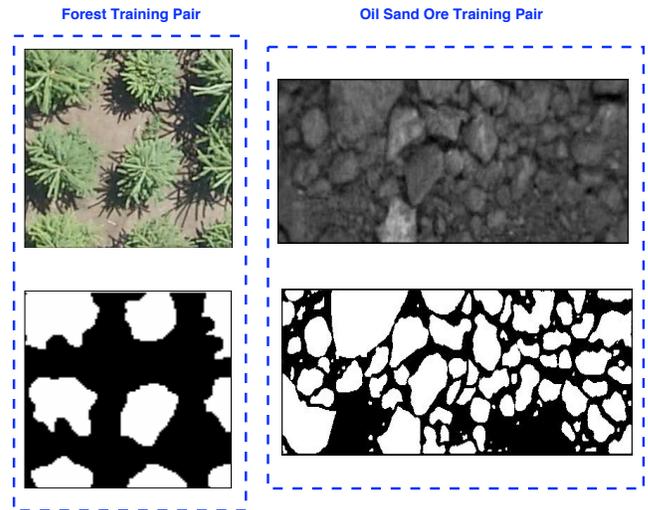


Figure 2: Training data for the forestry domain (left) and for the oil sand ore domain (right).

types, depending upon the depth at which it is mined. Thus, in order to be robust, a computer vision system segmenting ore images ought to be adaptive to wide variations in the input images.

Previously, the MLCV system was successfully applied to the forestry domain [9]. The oil sand ore domain is more challenging for several reasons. First, the images lack color information, rendering color operators used in the forestry domain, such as color histogram intersection, color correlation and red-green-blue (RGB) segmentation inapplicable. Consequently, the authors of [20] used the following image processing operators in the oil sand ore domain: bilateral noise removal, contrast enhancement, threshold, and morphological segmentation. In this study we use the following image interpretation operators: Gaussian filter, threshold and morphological segmentation. Figure 2 shows typical input images and their segmentations for both domains.

A second difference between the forestry and oil sand ore domains is that the pixel-level scoring function used in the former domain [9, 10] does not provide a meaningful measure of interpretation performance on oil sand ore image. It is more important to obtain an accurate size distribution of the ore fragments than it is to properly label each object pixel. Thus, we adopt a *sum of individual intersection over union* metric of [20], where interpretation accuracy within each fragment is measured. Figure 3 illustrates the difference. At the top, A and B show the original image, and the ground-truth (i.e., desired) interpretation respectively. The interpretations on the bottom (C and D) correctly label about the same number of pixels. Thus, the pixel-level intersection-over-union score of [10] would be approximately the same for both interpretations. In terms of the fragment size distribution, interpretation C is noticeably more accurate than D as the latter mistakenly conglomerates several ore pieces. This is accounted for by the per-fragment intersection-over-union metric of [20] which scores C at 0.43 and D at 0.17 (scores range between 0 and 1). Figure 4 shows distributions of fragment sizes for the ground truth (interpretation B), interpretation C, and interpretation D from Figure 3. The distribution is over the *cumulative per-*

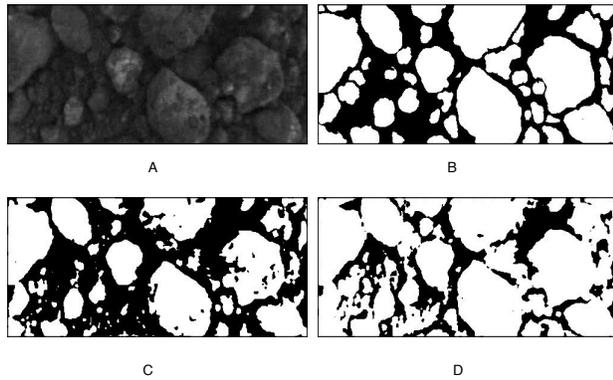


Figure 3: MLCV in the oil sand ore domain uses a scoring metric that penalizes interpretations for combining individual fragments of ore. Top: A and B are the original image and the ground-truth interpretation, respectively. Bottom: C and D are two distinct interpretations of A. Interpretation C receives a higher score than interpretation D because the distribution of ore fragment sizes measured from C is closer to the ground-truth distribution than the distribution measured from D.

*centage passing*, which represents the percentage of identified ore fragments smaller than a given diameter. This is a commonly used plot in the mining industry to test interpretation quality. Interpretation C’s segmentation matches that of the ground truth much more closely than that of interpretation D.

### 3. RELATED RESEARCH IN AUTOMATED IMAGE INTERPRETATION

Genetic programming has been applied directly to the image interpretations process [5]. Howard and Roberts use a two-step genetic program to recognize boats in Synthetic Aperture Radar (SAR) images, as well as ground vehicles in Infrared Linescan Imagery (IRLS). The first step lies with evolving a program that attempts to discriminate object pixels from a small collection of non-object pixels. The fittest detector evolved is then applied to the whole image, resulting in many false positives. In the second stage, another round of genetic programming is applied, and pixels are labeled as object pixels only if both the first and second stage detectors label them as the object. The evolved programs are based on pixel and local neighborhood statistics.

The detector evolved by the genetic programming approach is effective in recognizing both the boats and ground vehicles in their experimental settings. This is an example of *direct application* of genetic programming to image interpretation. The GAMM tool is not designed to interpret images, but instead to choose action sets for MDP agents. Its application in this paper is to choosing sets of vision operators for a vision system, but it can be applied to non-vision MDP systems such as terrain navigation.

There have been a variety of methods applied to forestry image interpretation. Example-based image models involve matching templates of trees to areas of an image [11]. This

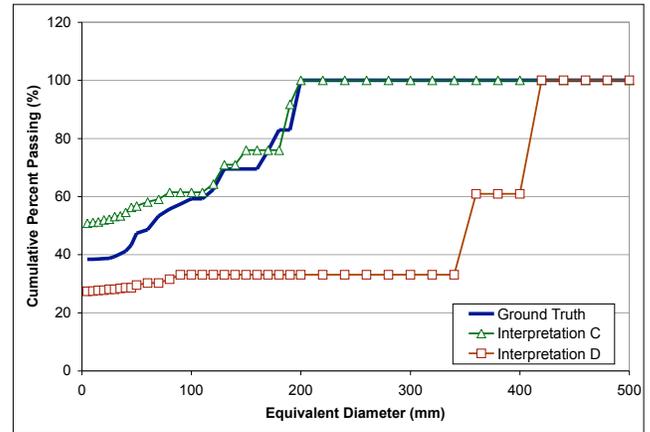


Figure 4: Cumulative percent passing plots of three interpretations. Interpretation C is closer to the ground truth than interpretation D.

requires a large database of different tree crowns, since aerial images will have various tree species and sizes, as well as different illumination and terrain slant.

Model-based methods involve less image feature processing than other methods and use elementary image features to hypothesize large numbers of regions to match with 3D CAD tree models. The STCI system [13] uses a tree crown matching technique much like the example-based method, but derives its template mathematically, modeling the summit of the tree crown as a generalized ellipsoid of revolution. Ray-tracing techniques are used to complete the tree crown template [8]. Much like the example-based methods, model-based methods require many different templates. Both methods also have difficulty dealing with dense overlapping foliage, changes in terrain and irregularities in tree crown shape. In addition to this, both involve much development time and take advantage of domain specifics.

MLCV has been shown to be robust to noise in the training data, illumination, and camera angle variations [10]. It requires little human time to port from one vision domain to another [20], and does not require domain expertise. We describe MLCV in detail in the next section.

### 4. ADAPTIVE IMAGE INTERPRETATION

In order to evaluate action set selection, we experimented with the Machine Learning for Vision (MLCV) system [20], in an attempt to optimize the set of vision operators (viewed as MDP actions in MLCV) with respect to balancing image interpretation accuracy and interpretation time. In order to make the paper self-contained, we describe the MLCV system below.

MLCV is the evolution of computer vision systems called Multi-Resolution Adaptive Object Recognition (MR ADORE) [10] and Adaptive Object Recognition (ADORE) [2]. MLCV models image interpretation as a Markov Decision Process. Namely, vision operators are the MDP actions, and images (or parts of images) are the MDP states. Operators are applied to images to produce further images, until an image interpretation is produced.

MLCV operates in two stages. In the *training stage* all valid limited-length operator sequences are expanded, which

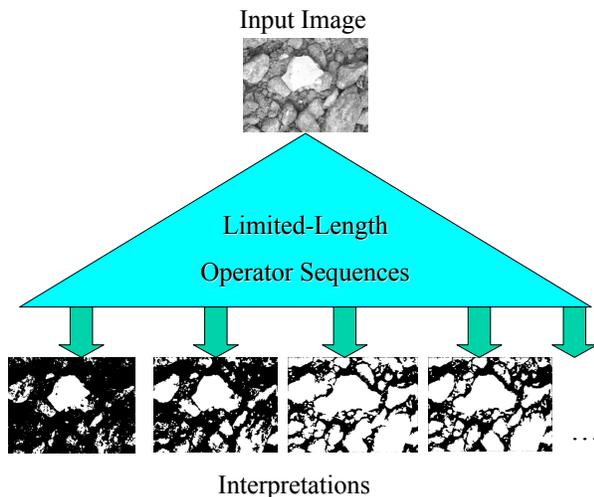


Figure 5: During MLCV’s training stage, all valid fixed-length sequences of image processing operators are applied to a training image. The resulting interpretations are then scored with respect to a user-supplied ground truth. The scores are used to learn a state-action value function.

is called a *full expansion*. Figure 5 illustrates the process. Rewards are then computed by comparing the resulting interpretations to the user-supplied ground-truth. Off-policy reinforcement learning with roll-outs is used to learn a value function to be used for MLCV operation of novel input images. Specifically, rewards are *backed-up* along the operator sequences using dynamic programming, producing a state-action value function  $Q : S \times A \rightarrow \mathbb{R}$ , computed for the actual expanded states  $S' \subset S$  and actions  $A' \subset A$ . The value of  $Q(s, a)$  is the expected cumulative reward if action  $a$  is taken in state  $s$ , and optimal choices are made thereafter until an interpretation is produced. The Q-notation comes from Watkins’ Q-learning [21].

During the *online stage*, MLCV greedily uses its machine-learned state-action value function ( $Q$ ) to select vision operators while interpreting a novel image. It then outputs to the user what it predicts to be the best interpretation (i.e., the one with the highest expected reward). Figure 6 illustrates the process.

## 5. AUTOMATED ACTION SET SELECTION

Performance of an MDP-based reinforcement learning agent depends on its choice of actions. Too many actions can confuse the agent, presenting it with too many (inferior) choices during learning and acting. Too few actions can hinder the agent by limiting its options, thus not allowing it to perform optimally in its environment. In order to automate the action set selection process in MDPs, a tool called Genetic Algorithms with Meta-Models (GAMM) was developed [9]. For reader’s convenience, we summarize the key elements of GAMM below.

GAMM begins by providing an MDP agent with random subsets of the full action set. The agent then acts and learns in the environment with each set. This generates training data of the form  $\{a, f(a)\}$  with  $f(a) = \alpha r(a) + \beta c(a) + \sigma$

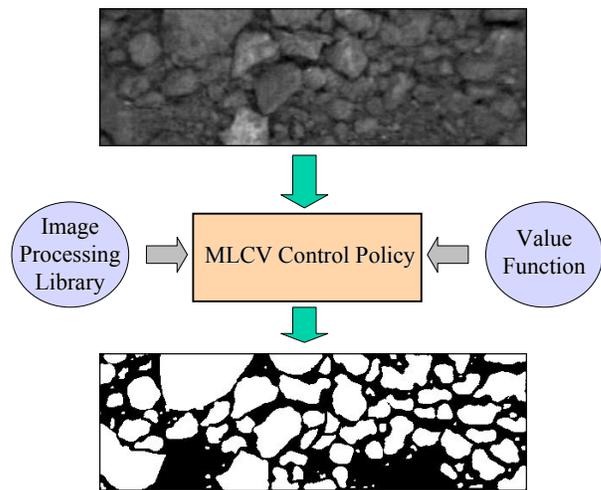


Figure 6: Online operation of MLCV. The control policy selects operators greedily with respect to its learnt value function. This is done in an attempt to produce an image interpretation with the highest *expected accuracy*.

being the *fitness of action set*  $a$ , calculated as a combination of the reward collected  $r(a)$  and the running time  $c(a)$ . Here  $\alpha$  and  $\beta$  are scaling coefficients, while  $\sigma$  is an additive constant. Each initial random action subset contributes one data point to the training set. As a tractable number of such subsets is bound to be small in any realistically scaled domain, GAMM extrapolates the data onto the collection of all possible subsets of the original action set. This is done using a generalization method such as machine learning with artificial neural networks. The resulting generalized fitness function is called *meta-model* following [6]. Figure 7 illustrates the process.

Next, a genetic algorithm performs a search in the space of action sets, using the machine-learned meta-model as a fitness function. The set  $a$  with the maximum  $f(a)$  is the output of GAMM, Figure 8.

## 6. RELATED RESEARCH IN FEATURE SELECTION

GAMM was a pioneering tool for automated selection of action sets for agents in Markov Decision Processes. A related field of research is automatic feature selection. Features are similar to MDP actions in that they can be interdependent, redundant and that their performance can only be fully evaluated by running the actual system. Below, we revisit the discussion of these methods found in [9].

Traditionally, greedy methods were used for feature selection [7, 14], but an improvement was found by implementing a GA for the task [19]. Within these GAs used for feature selection, both weighted combinations of criteria [16, 18] and pareto-optimal optimization [4, 12] proved successful.

While feature selection bears some similarity to action set selection, there are important differences. All features are applied to the data simultaneously, while actions are applied sequentially, with the output of one action being the input of the following action. Also, the application of ac-

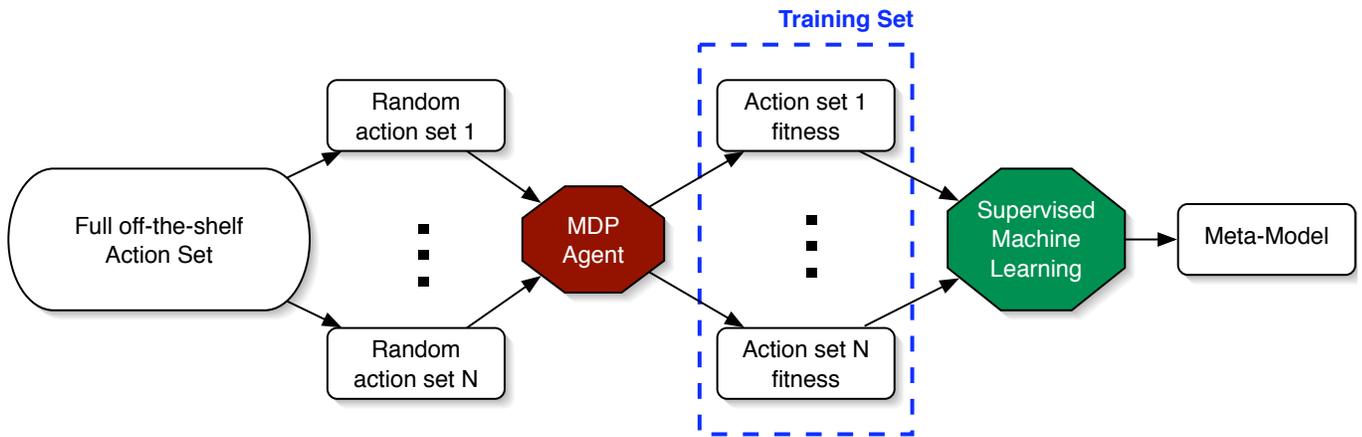


Figure 7: GAMM: Part 1: Meta-models are learned from evaluating  $N$  random subsets of the full action set, and combining the rewards collected and running time into a fitness value.

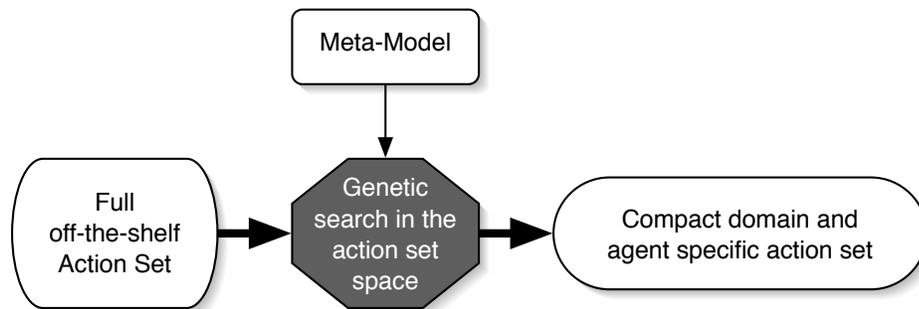


Figure 8: GAMM: Part 2: Discovery of an action set for a MDP agent.

tions is guided by a dynamic control policy that can involve loops, back-tracking and early termination. Parameterization is typical with actions, a requirement seen much less when working with features. This limits the applicability of feature selection methods to action set selection.

## 7. EMPIRICAL EVALUATION

In order to demonstrate portability of the GAMM tool, we applied it to the MLCV system for oil sands ore image interpretation. This is the first application of GAMM outside of the original publication. The tool itself was made available to us by the authors of [9] and was used “as is” in the new domain. A total of 150 labeled ore images were made available to us. These images were captured with a camera suspended above a conveyor belt carrying ore from a mine in Fort McMurray, Alberta. In each fold, a selection of 100 images were used for MLCV training and the other 50 images were used for testing. This process was repeated over 100 cross-validation folds each partitioning the 150 images randomly. The control parameters are found in Table 1.

During each generation of the genetic algorithms within the GAMM tool, two individuals were chosen for mating from the evolving population of action sets using rank selection (i.e., the probability of a chromosome being chosen is

proportional to its rank within the population). Using uniform crossover, two offspring were produced, and replaced two members of the population, again using rank selection. The children were then mutated according to the mutation rate. The initial population was created using a weighted random scheme.

Two different meta-models to approximate the actual fitness of action sets were used, namely, artificial neural networks (GA/NN) and naïve Bayes (GA/NB).

As in the original application of GAMM [9], the competition consisted of a method called “Top” (Algorithm 1), that ranks an operator  $a$  based on the average fitness of action sets containing  $a$ . We also compared against the full action set, a hand-selected action set, and a randomly selected action set. The full action set is shown in Table 2. Both the Top method and the randomly selected action sets are composed of a random number of operators at each cross-validation fold. This is because the Top method only ranks operators, it does not indicate what number of them to include in a set.

Section 7.1 shows the performance of MLCV with a perfect policy. That is, instead of using the regular control policy that MLCV learns during its training phase, the best produced interpretation was output to the user for each novel image. This can be viewed as the *potential* for each

**Table 1: Cross-validation experiment methods and parameters.**

Methods and Data Used	
Number of training operator sets:	1000
Number of training images:	50
Number of validation images:	50
Number of testing images:	50
Selection methods used:	Genetic Algorithms, Random selection, Top, Full set, Manual selection
Fitness functions (meta-models) used (within GAs):	Artificial Neural Networks, Naïve Bayes
GAMM Parameters	
Populations:	100, 500
Generations:	1000, 10000
Mutation Rates:	0.05, 0.2
Crossover:	Uniform
Fitness Equation Coefficients	
$\alpha$ :	$1/(2 \cdot MaxAccuracy)$
$\beta$ :	$1/(2 \cdot MaxCost)$
$\gamma$ :	$1/2$

**Algorithm 1** A filter method, called “Top” for selecting action sets.

**Input:** Training data, desired number of operators  $d$  in set

**Output:** Domain specific action set

- 1: for each operator  $A$  in the full set do
- 2:   for each training datum {attributes,fitness} do
- 3:     if  $A$  is present in the training datum then
- 4:       Add this fitness to  $A$ 's total
- 5:       Increment count
- 6:     Calculate  $A$ 's fitness by dividing total by count
- 7:     Sort actions by their fitness
- 8:     Output top  $d$  operators

action set, since this is the best fitness MLCV can possibly achieve with each set.

Section 7.2 shows results with the MLCV machine-learned policy guiding the decision concerning which interpretation to output. In all sections, the fitness of an action set  $a$  was calculated as:

$$f(a) = 0.5 \frac{accuracy}{MaxAccuracy} - 0.5 \frac{cost}{MaxCost} + 0.5$$

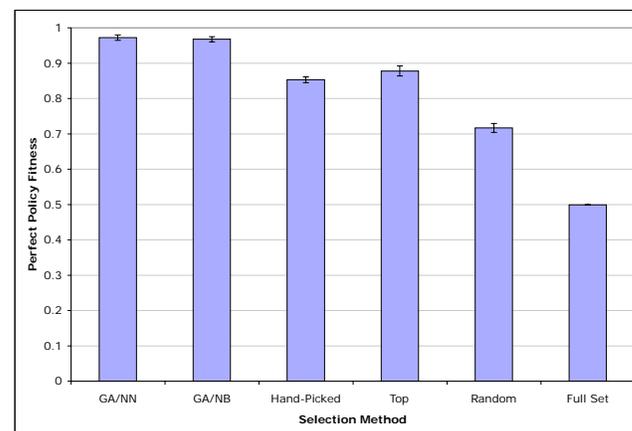
and falls in the range  $[0, 1]$ . The terms  $MaxAccuracy$  and  $MaxCost$  are the interpretation accuracy and the execution cost of the full action set, using the perfect policy.

## 7.1 Perfect Policy

In these experiments, MLCV was run with action sets chosen by each of the described selection methods. The performance was measured on 50 test (i.e., never seen before) oil sand ore images. For each image, MLCV with a given action set produces at least one interpretation of the image. Using the full action set, for instance, 3066 interpretations of each image were produced. With the perfect policy, MLCV output the best interpretation with respect to the scoring

**Table 2: The full action set used in oil sand ore experiments, consisting of 33 parameterized vision operators. The morphological segmentation operator does not take any parameters.**

Operator/Parameter	Parameter values
Gaussian Filter/ <i>Gaussian Matrix Dimension</i>	1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21
Threshold/ <i>Threshold Value</i>	25, 35, 45, 55, 65, 75, 85, 95, 105, 115, 125, 135, 145, 155, 165, 175, 185, 195, 205, 215, 225
Morphological Segmentation	no parameters



**Figure 9: Fitness of action sets found by six different selection methods, using the perfect policy.**

metric. Figure 9 shows the fitness of the action sets chosen by six selection methods with the perfect policy.

The full set achieves a fitness of 0.5 in every fold, since it achieves both  $MaxAccuracy$  and  $MaxCost$ . The sets found by the GAMM with a neural network for a meta-model achieve 97% of the maximum interpretation accuracy possible, while taking only 2% the running time of the full set. The fittest action set found by GAMM, with respect to the perfect policy, is shown in Table 3.

**Table 3: The fittest action set found by GAMM, with respect to the perfect policy.**

Operator/Parameter	Parameter values
Gaussian Filter/ <i>Gaussian Matrix Dimension</i>	15
Threshold/ <i>Threshold Value</i>	45, 55, 105, 115, 125, 145
Morphological Segmentation	no parameters

## 7.2 Machine-learned Policy

In these experiments, MLCV first learned a state-action value function ( $Q$ ) and then used it greedily to choose an

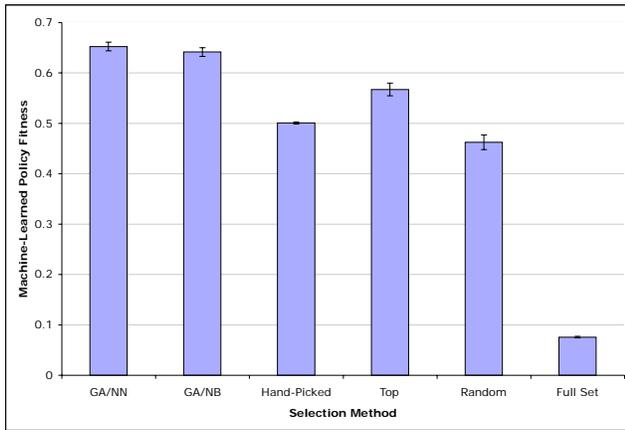


Figure 10: Fitness of action sets found by six different selection methods, using machine learned policy.

Table 4: The fittest action set found by GAMM, with respect to MLCV’s machine-learned policy.

Operator/Parameter	Parameter values
Gaussian Filter/ Gaussian Matrix Dimension	5
Threshold/ Threshold Value	35, 45, 55, 115, 125, 135, 145
Morphological Segmentation	no parameters

interpretation of a novel image (cf., Section 5). Figure 10 shows the fitness of action sets chosen by six selection methods with the machine-learned policy, on 50 test images. Here, the fitness of an action set is determined by the following process. First, the selection method produces an action set. Then, a policy is learned with the produced action set. This policy is then evaluated on test images (with the produced action set). The result is what is presented as the accuracy and cost of the selection method.

The running time of MLCV using sets found by GAMM is 2% that of running MLCV with the full set, as the fittest action set found by GAMM contained only 9 operators, compared to 33 in the full set. This demonstrates GAMM’s ability to remove redundant and ineffective operators given domain specifics. Secondly, the sets found by GAMM with the neural network meta-model actually *outperform* the full set in terms of interpretation accuracy, with the full set obtaining only 45% the interpretation accuracy of the GA/NN set. This demonstrates GAMM’s ability to tailor its operator set to MLCV’s machine-learned control policy. Namely, a bulk of low-quality interpretations are never generated with the compact GAMM-selected operator set. This leaves MLCV with fewer opportunities to err. The improvement in interpretation performance despite having fewer choices parallels human behavior where providing people with fewer choices in life causes them to spend less time learning about useless choices [15].

## 8. DISCUSSION AND FUTURE WORK

A recent generation of computer vision systems take advantage of reinforcement learning techniques to adapt their operation to a broad spectrum of varying input conditions. This accelerates the development cycle and makes these systems robust enough to deploy in the field. One of the last vestiges of human intervention with such systems is the need to select a compact high-performance set of actions. Until last year, this task had remained a laborious and error-prone manual process based on trial and error. At GECCO’05 genetic algorithms were shown successful in automating this task. The impressive demonstration was based on a single point and raised the questions to portability and scalability of the approach.

This paper presented the first demonstration of cross-domain portability of the approach. We applied the tool revealed last year to a different and more challenging domain. In it, genetics algorithms were able to increase the interpretation accuracy by 55% while reducing the running time of the system by 98%. This exceeds the performance gains published in the original paper last year.

The importance of these results is three-fold. First, the empirical evidence supports the claim that adaptive image interpretation systems that use machine learning to acquire a control policy over image processing operators, benefit from having a *smaller* number of choices. Indeed, the presence of many suboptimal action choices, though harmless with a perfect control policy, confuses machine learned control policies. Second, the new results demonstrate that the tool proposed year (GAMM) can be applied to a different domain without tweaks or modifications. Third, the results in this paper provide preliminary evidence that GAMM can scale up to more difficult domains.

Future work includes further experimenting with parameters of the genetic algorithms, applications to yet more challenging domains, and comparing against other methods of action set selection. Also, we will investigate a *redundancy measure* in an attempt to assess potential gains of applying GAMM to a given action set.

## 9. ACKNOWLEDGEMENTS

We appreciate contributions by Xiaoli Wang, Mark Polak, Hong Zhang, Ron Kube, Syncrude Research Ltd., Ilya Levner, Bruce Draper, Guanwen Zhang, Dorothy Lau, Terry Caelli, Li Cheng, Joan Fang, Michael Chung, Wesley Mackay, René Malenfant, Jonathan Newton, and the Alberta Research Council team. We are also grateful for the support from the University of Alberta, the Natural Sciences and Engineering Research Council (NSERC), the Informatics Circle Of Research Excellence (iCORE), and the Alberta Ingenuity Centre for Machine Learning (AICML).

## 10. REFERENCES

- [1] B. Draper. Object recognition as a Markov decision process. In *Proceedings of the International Conference on Pattern Recognition*, volume IV, pages 95–99, Vienna, Austria, 1996.
- [2] B. Draper, J. Bins, and K. Baek. ADORE: adaptive object recognition. *Videre: A Journal of Computer Vision Research*, (4):86–99, 2000.
- [3] B. A. Draper. From knowledge bases to Markov models to PCA. In *Proceedings of Workshop on*

- Computer Vision System Control Architectures*, Austria, 2003.
- [4] C. Emmanouilidis, A. Hunter, J. MacIntyre, and C. Cox. Multiple Criteria Genetic Algorithms for Feature Selection in Neurofuzzy Modeling. In *In Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, Washington, D.C., 1999.
- [5] D. Howard and S. C. Roberts. A staged genetic programming strategy for image analysis. In W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, and R. E. Smith, editors, *Proceedings of the Genetic and Evolutionary Computation Conference*, volume 2, pages 1047–1052, Orlando, Florida, USA, 13-17 July 1999. Morgan Kaufmann.
- [6] Y. Jin, M. Olhofer, and B. Sendhoff. Managing approximate models in evolutionary aerodynamic design optimization. In *Proceedings of the 2001 Congress on Evolutionary Computation CEC2001*, pages 592–599, COEX, World Trade Center, 159 Samseong-dong, Gangnam-gu, Seoul, Korea, 27-30 May 2001. IEEE Press.
- [7] K. Kira and L. Rendell. The feature selection problem: Traditional methods and a new algorithm. In *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI-92)*, pages 129–134, 1992.
- [8] M. Larsen and M. Rudemo. Using ray-traced templates to find individual trees in aerial photos. In *Proceedings of the 10th Scandinavian Conference on Image Analysis*, volume 2, pages 1007–1014, Lappeenranta, Finland, 1997.
- [9] G. Lee and V. Bulitko. GAMM: genetic algorithms with meta-models for vision. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, pages 2029–2036, Washington, DC, 2005.
- [10] I. Levner and V. Bulitko. Machine learning for adaptive image interpretation. In *Proceedings of the National Conference on Artificial Intelligence (AAAI) and Innovative Applications of Artificial Intelligence Conference (IAAI)*, pages 870 – 876, San Jose, California, 2004.
- [11] D. Murgu. Individual tree detection and localization in aerial imagery. Master’s thesis, Department of Computer Science, University of British Columbia, 1996.
- [12] L. S. Oliveira, N. Benahmed, R. Sabourin, F. Bortolozzi, and C. Y. Suen. Feature subset selection using gas for handwritten digit recognition. In *In Proceedings of the 14<sup>th</sup> Brazilian Symposium on Computer Graphics and Image Processing*, pages 362–369, Florianópolis-Brazil, 2001. IEEE Computer Society.
- [13] R. Pollock. A model-based approach to automatically locating tree crowns in high spatial resolution images. In J. Desachy, editor, *Image and Signal Processing for Remote Sensing*, 1994.
- [14] P. Pudil, J. Novovicova, and J. Kittler. Floating search methods in feature-selection. *Pattern Recognition Letters (PRL)*, 15(11):1119–1125, November 1994.
- [15] B. Schwartz. The tyranny of choice. *Scientific American*, pages 70–75, April 2004.
- [16] Z. Sun, X. Yuan, G. Bebis, and S. Louis. Neural-network-based gender classification using genetic eigen-feature extraction. In *In Proceedings of IEEE International Joint Conference on Neural Networks*, Honolulu, Hawaii, 2002.
- [17] R. Sutton and A. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.
- [18] H. Vafaie and K. D. Jong. Genetic algorithms as a tool for feature selection in machine learning. In *In Proceeding of the 4th International Conference on Tools with Artificial Intelligence*, pages 200–204, Arlington, VA, 1992.
- [19] H. Vafaie and K. D. Jong. Robust feature selection algorithms. In *In Proceedings of the Fifth Conference on Tools for Artificial Intelligence*, pages 356–363, Boston, MA, 1993. IEEE Computer Society Press.
- [20] X. Wang, M. Polak, V. Bulitko, and H. Zhang. Machine learning for adaptive parameter selection in ore image segmentation. In *Proceedings of the National Conference on Artificial Intelligence (AAAI), Workshop on Learning in Computer Vision*, Pittsburgh, Pennsylvania, 2005.
- [21] C. Watkins. *Learning from Delayed Rewards*. PhD thesis, King’s College, University of Cambridge, UK, 1989.