GAMM: Genetic Algorithms with Meta-Models for Vision

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

ABSTRACT

Recent adaptive image interpretation systems can reach optimal performance for a given domain via machine learning, without human intervention. The policies are learned over an extensive generic image processing operator library. One of the principal weaknesses of the method lies with the large size of such libraries, which can make the machine learning process intractable. We demonstrate how evolutionary algorithms can be used to reduce the size of the operator library, thereby speeding up learning of the policy while still keeping human experts out of the development loop. Experiments in a challenging domain of forestry image interpretation exhibited a 95% reduction in the average time required to interpret an image, while maintaining the image interpretation accuracy of the full library.

Categories and Subject Descriptors

I.4.8 [**Computing Methodologies**]: Image Processing and Computer Vision—*Scene Analysis*

General Terms

Performance

Keywords

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

1. INTRODUCTION

Image interpretation and object recognition are important and highly challenging problems with numerous practical applications. Hand-crafted image interpretation systems suffer from an expensive design cycle, a high demand for subject matter and computer vision expertise, and difficulties with portability and maintenance. Consequently, in the last three decades, various *automated* ways of constructing image interpretation systems have been explored [3].

GECCO'05, June 25–29, 2005, Washington, DC, USA.

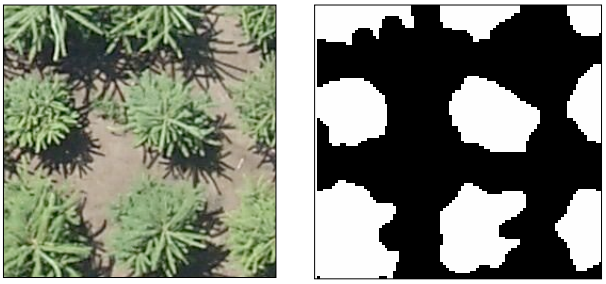

Figure 1: A fragment of an aerial image taken over a spruce plot is shown on the left. The right image is the desired interpretation with spruce canopies labeled in white. It is provided as a part of the training set.

A promising recent approach casts the image interpretation task as a sequential decision making process over a library of image processing operators [11]. Interpreting an image is then reduced to selecting a sequence of operators, and their parameters, to apply to it. In order to keep the required human expertise to a minimum, a control policy over an image processing operator library is *machine-learned* for a particular domain. The learning process involves trying all valid limited length sequences of operators on expert-labeled training data. A control policy that selects an optimal operator sequence for an input image can then be learned [2, 11]. Figure 1 shows a typical labeling of a forestry image. While our method is domain-independent, we demonstrate it in the domain of forest image interpretation. This is a challenging and practically much-needed task as detailed in the rest of this section.

Forest maps and inventories have become a critical tool for wood resource management (planting and cutting), ecosystem governance and wild-life research. Unfortunately, forest mapping at the level of individual trees is a continuous and costly undertaking. Canada alone has an estimated 344 million hectares of forests to inventory on a 10-20 year cycle [13]. At these scales, ground-based surveys are not feasible. Researchers have therefore turned to developing automated systems to produce forest maps from airborne images. The final goal is to measure the type (species), position, height, crown diameter, wood volume and age class for every tree in the survey area.

The task of large-scale forest mapping from aerial images presents formidable challenges, including: (i) massive amounts of high-resolution data, in order to recog-

nize and measure individual tree crowns, (ii) construction and maintenance of (and providing access to) very large databases; Canada alone has an estimated 10^{11} trees, (iii) geo-referencing of airborne images for validation purposes, and (iv) orthorectification of aerial images, particularly given that elevation maps are often unavailable at the accuracy required. Of a special interest are the challenges created by the image content, including variations in sun and camera angle and the resulting shadow and tree crown overlap. These challenges have had an adverse effect on special purpose algorithms for individual tree identification [1]. The task is substantially challenging even to expert human interpreters, resulting in up to 40% error in comparison to ground-based surveys [5].

In this paper we focus on the tree labeling problem as a first step towards an overall solution. For each image, the task is to identify all pixels belonging to canopies of trees of a certain kind. In the illustrations throughout the paper, we label spruce canopies. Within each image depicting the user provided ground-truth, the target canopy pixels are labeled white, while the rest of image is left black (Figure 1).

2. PERFORMANCE ELEMENT: ADAPTIVE IMAGE INTERPRETATION

In order to appreciate the complexity and challenges of automated image processing operator library (IPL) selection, we will first briefly review the *performance element*: adaptive image interpretation systems [11]. Research on such systems pursues the following objectives: (i) rapid system development for a wide class of image interpretation domains; (ii) low demands on subject matter and computer vision expertise; (iii) accelerated domain portability, system upgrades, and maintenance; (iv) adaptive image interpretation wherein the system adjusts its operation dynamically to a given image; (v) user-controlled trade-offs between recognition accuracy and resources utilized.

In order to make such systems adaptive and cross-domain portable, use of large readily available off-the-shelf IPLs is favored. However, the domain independence of such libraries requires an intelligent policy to control the application of library operators. Operation of such a control policy is a complex and adaptive process. It is *complex* insomuch as there is rarely a one-step mapping from image data to object label; instead, a series of operator applications are required to bridge the gap between raw pixels and semantically meaningful labels. Examples of the operators include region segmentation, texture filters, and the construction of 3D depth maps. In this paper, we consider operators (i.e., vision routines) with the same function but different instantiations of parameters as *different* operators. For instance, a binary threshold routine with the threshold of 128 (i.e., all pixels brighter than 128 are turned white, all others are turned black) is considered a different operator than a binary threshold routine with the threshold of 200. Figure 2 presents a fragment of the IPL operator dependency graph for the forest image interpretation domain. Image recognition is an *adaptive* process in the sense that there is no fixed sequence of actions that will work well for most images. For instance, the steps required to locate and identify isolated trees are different from the steps required to find connected stands of trees. Therefore, the success of image interpretation depends on the ability of the system to *learn* an *adaptive* control policy over an IPL.

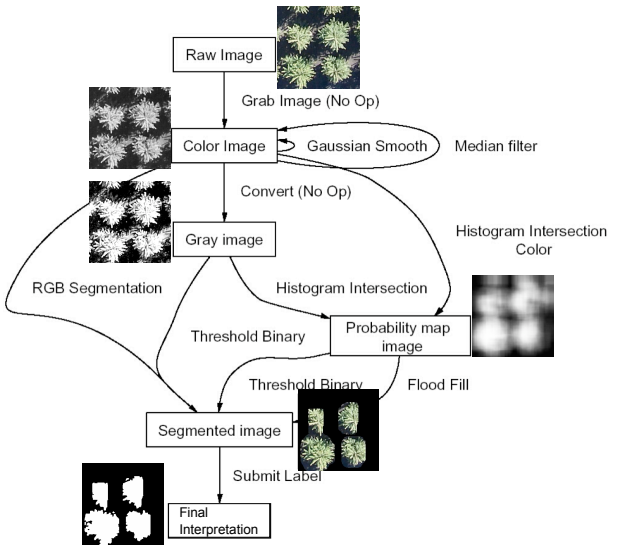

Figure 2: A fragment of the state-action graph used in our experiments. States are labeled with their vision data types and have forest samples shown next to them. Vision operators are shown as the arcs.

In the following we will consider a particular system called MR ADORE (Multi Resolution ADaptive Object REcognition) [11]. We begin with the Markov decision process (MDP) ([16]) as the basic mathematical model by casting the IPL operators as the MDP actions and the results of their applications as the MDP states (Figure 2). The system operates in two modes as follows.

During the *off-line training stage* (Figure 3), available subject matter expertise is encoded as a collection of training images with the corresponding desired interpretation (the so-called ground truth). Figure 1 demonstrates an example of such a pair (input image, ground truth label). Off-line training continues by invoking an off-policy reinforcement learning algorithm that uses deep backups without bootstrapping to acquire a value function [16]. Specifically, at first, all feasible limited-length sequences of IPL operators are applied to each training image. We refer to this process as a *full expansion*. The resulting interpretations are evaluated against the ground truth provided by the user. We use a pixel-level similarity scoring metric defined as the ratio of the number of pixels labeled as the target class (e.g., spruce) by both the system and the expert to the total number of pixels labeled as the target class by either one of them. According to such a metric, an interpretation identical to the user-supplied label scores 1 while a totally disjoint interpretation will get a score of 0.

The interpretation scores are then "backed up" along the IPL operator sequences using dynamic programming. As a result, the value function $Q : S \times A \rightarrow \mathbb{R}$ is computed for the expanded states $S' \subset S$ and applied actions $A' \subset A$. The value of $Q(s, a)$ corresponds to the best interpretation score the system can expect by applying operator a in state s and acting optimally thereafter [20]. In reinforcement learning terms, we are representing the task as a finite horizon non-

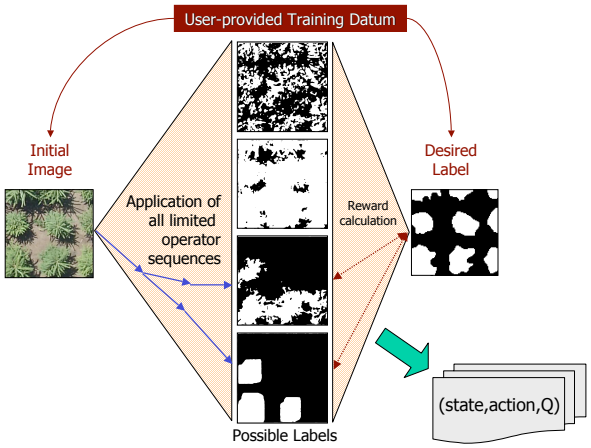

Figure 3: Off-line training stage: all limited-length operator sequences are applied to each training image. The resulting image interpretations are evaluated against the desired label. State-action rewards are then computed.

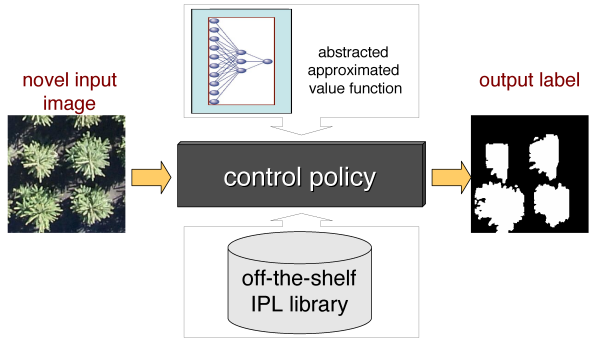

Figure 4: On-line operation: the control policy uses an approximate value function to select the best sequence of operators from the IPL library. As the result, an image interpretation label is produced.

discounted problem wherein all intermediate rewards are zero except these collected by outputting an image interpretation. The latter is a positive reward proportional to the quality of the interpretation.

The collected training set of Q-values $\{[s, a, Q(s, a)]\}$ samples a tiny fraction of the state action $S \times A$ space. Thus, we use function approximation methods to extrapolate the value function onto the entire space. To make approximation tractable, raw multimegabyte states are reduced to 192 real-valued features and are presented to Artificial Neural Networks (NNs), which in turn act as function approximators.

During the *on-line interpretation stage*, the system receives a novel image and interprets it, as shown in Figure 4. The previously-learned value function now guides the control policy to apply vision operators from the IPL library. Several policies have been tested [11]. In this paper, we employ the least-commitment policy that uses the value function greedily.

3. LEARNING TASK: AUTOMATED OPERATOR SELECTION

During the off-line phase, MR ADORE explores the state space by expanding the training data provided by the user. In doing so it applies all operator sequences up to a certain length. Longer sequences are preferred for better image interpretation, since more operators can be applied for more precise transformations of the input image into the desired label. Even the modest increase from four to six operators can be beneficial, as illustrated in Figure 5. On the other hand, the size of the state space explored ($|S' \times A'|$) increases exponentially with length and therefore the exploration process quickly becomes prohibitively expensive. With even a fairly compact operator set of 60 operators used in [11], the effective branching factor is approximately 26.5. The 292-operator library used in this paper is even more computationally demanding, with a branching factor of approximately 47.

Thus, there are three conflicting factors at work: (i) large off-the-shelf image processing operator libraries are required to make MR ADORE cross-domain portable, (ii) long operator sequences are needed to achieve high interpretation quality, and (iii) combinatorial explosion during the learning phase can impose prohibitive requirements on the storage and processing power. Fortunately, most domain-independent operator libraries almost invariably contain numerous redundant or ineffective operators when a specific domain is considered. This is especially true when we consider closely parameterized instantiations of each operator. For instance, threshold(200) and threshold(199) are likely to be too similar to justify inclusion of both in the image processing library. Thus, the feasibility of the policy learning phase as well as subsequent on-line performance critically depends on the selection of an efficient and compact parameterized operator subset for the domain of interest.

Previous systems such as in [2, 11] relied on *manual* selection of highly relevant non-redundant operators thereby keeping the resulting IPL small and the off-line state space exploration tractable. Unfortunately, such solutions defeat the main objective of MR ADORE-like systems: their automatic construction for a given domain. Consequently, an important objective for machine learning research is *automation* of operator library selection. In the following, we position this task in the context of related research, discuss applicability of existing methods, and evaluate a novel method for automated IPL selection.

4. EXISTING METHODS

Selecting an optimal operator library is similar to selecting an optimal set of features insomuch as the individual operators/features are interdependent, possibly redundant, and their performance can be fully evaluated only within the target system. Thus, we will first briefly review representative feature selection literature and then discuss the differences.

There are two important dimensions to consider: the type of search in the space of feature sets and the optimization criteria. The number of feature sets is exponential in the number of features and therefore incomplete heuristically guided search methods are typically preferred. For feature selection, greedy algorithms have been used [8, 14]. An improvement in performance over traditional greedy methods

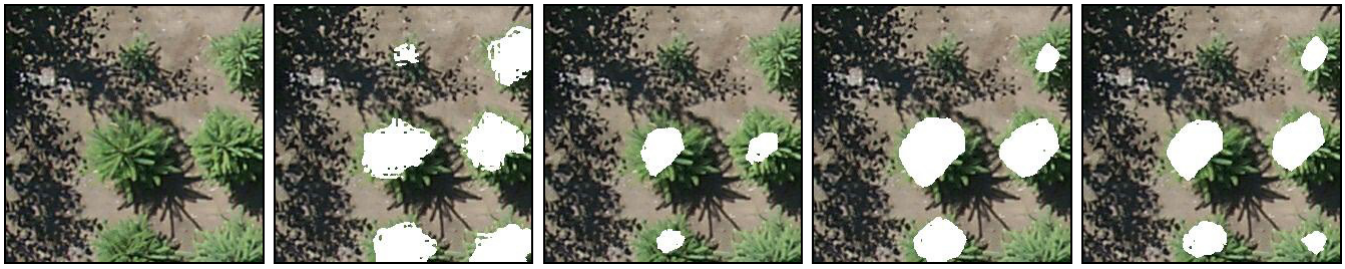

Figure 5: Longer operator sequences lead to a better label. From left to right: the original image, desired user-provided label, the best labels with an operator sequence of length 4, 5, and 6. All labels are superimposed on the source images for illustration purposes.

was later achieved with Genetic Algorithms (GAs) [18] . In addition to the weighted combination as the fitness function approach [15, 17], pareto-optimal GAs have been found successful for feature selection [4, 12].

Along the second dimension, two primary approaches have been studied. Wrapper methods [9, 19] measure the actual performance of the target system with a candidate feature set. While being accurate, such optimization criteria can be prohibitively expensive in itself. For instance, in the context of MR ADORE, measuring the performance of a typical operator set on a test suite of 35 images takes around 12 hours on a dual AMD Athlon MP+ 2400 Linux server [10].

Filter feature selection methods [8, 14] use system-independent criteria-based feature redundancy and relevance, and similar measures. While such criteria are frequently less expensive to compute, they are decoupled from the actual performance element and may not always account for the influence of domain specifics on the quality of a feature set. Additionally, once features are selected, all of them are applied to the data token at hand *simultaneously*. This is in contrast to image processing operators, which are applied to the initial image *sequentially* with one operator's output being the next operator's input. Furthermore, operator application is guided by a dynamic control policy (e.g., best-first NN-guided policy in [2]) and can involve loops and back-tracking. The fact that many operators also require parameters while features do not presents another complication. These factors impede the mapping from operator sets to the resulting target system performance thereby limiting the applicability of filter methods.

5. PROPOSED NOVEL APPROACH

Wrapper approaches use the correct optimization criteria but can be prohibitively expensive. Filter approaches are computationally feasible but do not necessarily deal well with complex interdependencies among operators since they have no access to performance of the actual system. In our experiments, we extend the work in [6] and combine the best of wrapper and filter approaches by using a wrapper-like search in the space of operator sets. Unlike traditional wrapper methods, we guide the search with a *fast* but at the same time *domain-specific* fitness function. In order to reduce the amount of human intervention, we first tabulate all operators parameterizations in a uniform fashion, (i.e., using no domain expertise), and then employ machine learning methods to acquire the fitness function *automatically*. These *meta-models* ([7]) are generalized over training data to provide our heuristic search with a quick and accurate ranking for any given action set. This setup results in a four-step process, as follows:

Step 1: we evaluate a small collection of selected operator sets via running each of them with the actual system (MR ADORE) on a set of training images as shown in Figure 6. For each operator set A, all limited-length sequences of operators from A are applied to each training image. Each sequence is assigned an image interpretation accuracy for the image label it produces. The maximum image interpretation accuracy for all sequences from operator set A averaged over all training images is stored as $r(A)$. Each operator set also incurs an *execution cost* $c(A)$, which is a measure of the total time taken for the full expansion with this set. The operator set's *fitness* $f(A)$ is then defined as $f(A) = \alpha r(A) - \beta c(A) + \sigma$ where α and β are scaling coefficient, and σ is an additive constant. These variables enable the user to control the trade-offs in a domain specific fashion.

Step 2: step one results in a collection of operator sets and their fitness values $\{A, f(A)\}$. In the second step, we generalize this collection using machine learning (ML) techniques (Figure 6). As a result, a generalized fitness function r_{ML} is acquired.

Step 3: once machine learning is over, we use the generalized fitness function r_{ML} as the optimization criteria in genetic algorithms[1], in the space of operator sets (Figure 7).

Step 4: the operator sets found by the search are then evaluated against a set of validation images. The best m operator sets are output to the user to be used in the system of interest.

We call this method *Genetic Algorithms with Meta Models* (GAMM) and compare it to other operator set selection methods in our experiments.

6. EMPIRICAL EVALUATION

Experimental results presented below serve to support two claims made at the onset of the paper. Specifically, section 6.1 supports the claim that it is possible to *automatically* select a highly compact subset of the initial IPL, while retaining the interpretation performance of the original library. Section 6.2 shows that such reduction is beneficial not only due to the great acceleration of the training phase, but also as it reduces the number of suboptimal labelings for the control policy to select from. In all experiments, α, β and σ are set to 0.5. In this way, image interpretation accuracy and execution cost are given equal bearing in determining an operator set's fitness.

[1]We have also explored other randomized heuristic search methods such as simulated annealing (SA).

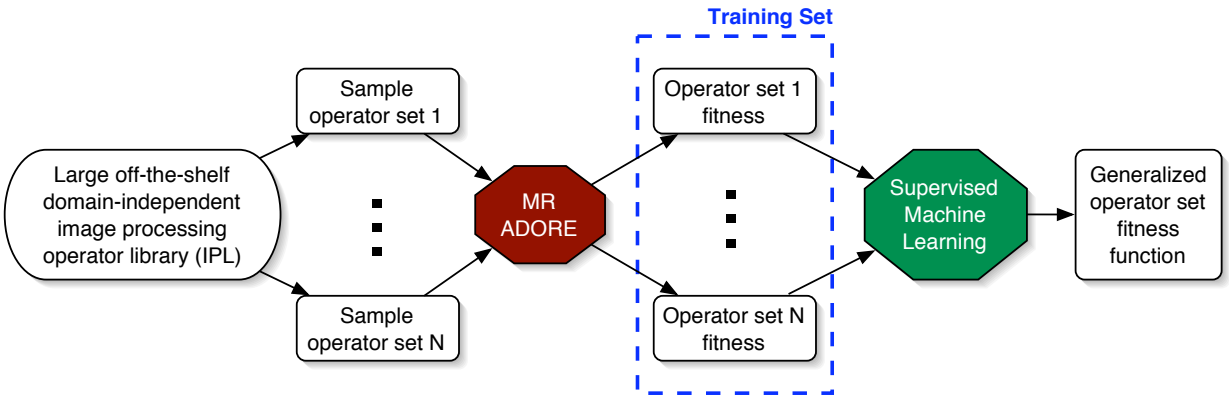

Figure 6: Supervised machine learning methods are used to generalize fitness of sampled operator sets into an approximation to the actual fitness criteria.

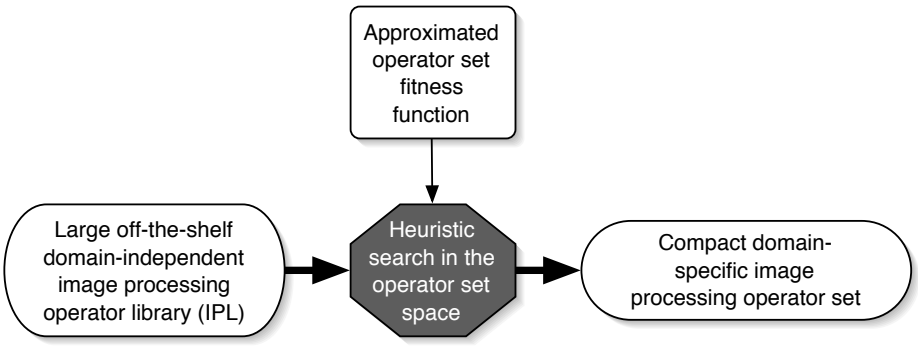

Figure 7: GAMM: genetic search is conducted in the space of operator sets. It is guided by a machine-learned approximation to the performance function of the actual system.

6.1 Accelerated Training

Several experiments were performed in order to determine the best machine learning algorithms to use as fitness functions within the evolutionary search, as well as parameters for both the fitness functions and the search methods themselves, and which search method to use.

For machine learning fitness functions, while we found that artificial neural networks (NNs) yielded the best results, we did experiment with the naïve Bayes (NB) classifier. The best population sizes for the GA were found to be {100,500}, the best number of iterations were found to be {1000,10000} and the best mutation rates were found to be {0.05,0.2}. GAs outperformed Simulated Annealing in all preliminary experiments [10]. We then ran our four step method with 72 images of young spruce plots maintained by the Alberta Research Council in Vegreville, AB. The images were captured in 24-bit colour at 256×256 pixels per image. The empirical setup is summarized in Table 1.

We used the Top filter method (Algorithm 1) of selecting operators to compare against GAMM. This method ranks operators based upon the average fitness of sets in which they are present. This is analogous to the +/- statistic in hockey, where a player on the ice is given a + if his or her team scores a goal, and is given a - when the opposition scores. We also compared against a set chosen by a domain-expert, a randomly selected set and the full set of operators.

Table 1: Cross-validation experiment methods and parameters.

Methods and Data Used	
Number of training data	1165
Number of training images	24
Number of validation images	24
Number of testing images	24
Selection methods used	GAs, Random Selection, Filter Method, Full Set, Domain Expert Selection
Fitness functions (meta-models) used (within GAs)	Artificial Neural Networks, Naïve Bayes
Search Method Parameters	
Populations	100, 500
Iterations	1000, 10000
Mutation (Flip) Rates	0.05, 0.2
Experimental time	90 days

Algorithm 1 A filter method, called "Top" for selecting operator sets. **Input:** Training data, desired number of operators d in set **Output:** Domain specific operator set(s)

```
for each operator A in the full set do
    for each training datum {attributes,fitness} do
        if A is present in the training datum then
            Add this fitness to A's total
            Increment count
    Calculate A's fitness by dividing total by count
Sort actions by their fitness
Output top d operators
```

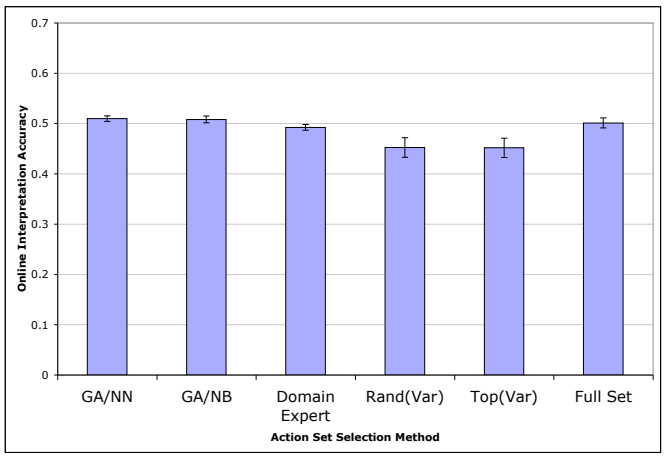

Figure 8: MR ADORE's on-line interpretation accuracy with sets chosen by GA/NN, GA/NB, domain expert, Top(0-292) of operators, random(0-292) operators, and the full set of 292 operators. Here "(0-292)" indicates a random number of operators chosen at each fold, ranging from 0 to 292.

Both the filter method and the random method of selecting operators chose variable sized operator sets (changing with each fold), between 0 and 292 operators (in addition to three mandatory operators present in each set). We ran 81 cross-validation folds with the {GAs} ×{NB,NN} set of combinations within the GAMM method, as well as the comparison methods.

In the image interpretation domain, we are most interested in the performance of a system on *novel* images, where the labeling is not yet known. In MR ADORE, the *on-line module* uses a machine-learned control policy to select a labeling for novel images. In Figure 8 we show the on-line image interpretation accuracy of the operator sets chosen by the six different methods of operator set selection. Figure 9 demonstrates that the GAMM produces the operator sets with the best on-line fitness, which combines both interpretation accuracy and execution cost. These figures show that GAMM is capable of effectively trimming an operator set to minimize the execution cost while retaining the interpretation accuracy of the full image processing library.

6.2 Improving On-line Performance

The second set of experiments was conducted to explore the second hypothesis of the paper. Namely, can GAMM not only speed up learning, but also increase the image interpretation accuracy by eliminating operator sequences?

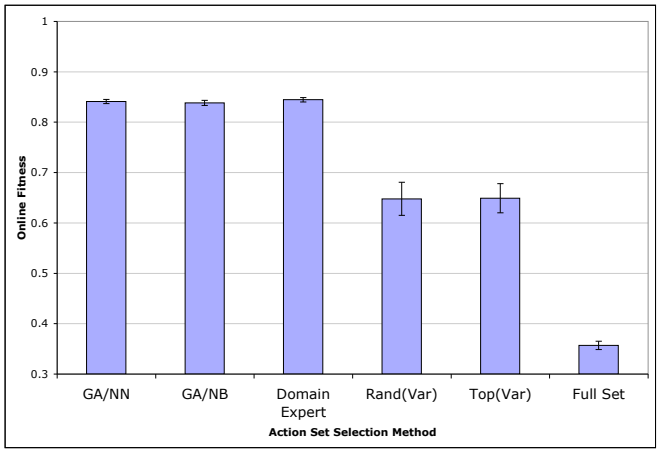

Figure 9: MR ADORE's on-line fitness with sets chosen by GA/NN, GA/NB (both GAMM), domain expert, Top(0-292) of operators, random(0-292) operators, and the full set of 292 operators. GAMM matches the fitness of the domain expert set, and greatly outperforms the Top and Random methods, as well as the full operator set.

Any policy π other than the optimal policy π^* will commit errors in its selection of actions in an MDP. We devised a set of experiments to simulate the degree of error incurred by the on-line policy. We did this in an attempt to understand why, how, and when the on-line module of MR ADORE chooses *sub-optimal* final labelings, given an operator set. We invoked the off-line (oracle) module to test the operator sets chosen by each method, but introduced random choices of image interpretations to mimic the mistakes made by the on-line policy.

First, we ran the off-line module (in the same way as before) with an ε chance of randomly choosing an image interpretation from the choices generated by an operator set, and thus a $1-\varepsilon$ chance of choosing the optimal image interpretation (since this is known off-line). Henceforth we will refer to this model as the *ε-perfect* control policy.

We evaluated seven methods of choosing operator sets with MR ADORE running with the optimal off-line policy, on-line policy and *ε-perfect* policy with $\varepsilon \in \{0.1, 0.3, 0.5, 0.7, 0.9 \text{ and } 1\}$. Results are shown in Figure 10. Note that with the perfect off-line policy, the GA/NN and GA/NB sets are the best for image interpretation accuracy, but that these methods experience a greater accuracy drop than the Top(50) method once randomness is introduced. We fixed operator set sizes for the Top method at 50, in order to see if larger operator sets suffered more loss when the chance for a random choice was increased, and because GAMM typically chose operator sets of approximately 50 operators. As can be seen on the graph, larger operator sets are more susceptible to interpretation accuracy loss when random choice is added than are smaller operator sets. This finding supports the thesis that more choice can be detrimental with a machine-learned control policy.

6.3 Extensions: Seeding and Elitism

The Top and GAMM methods need not be used disjointly. We can use the Top method to *seed* the GAs with sets in

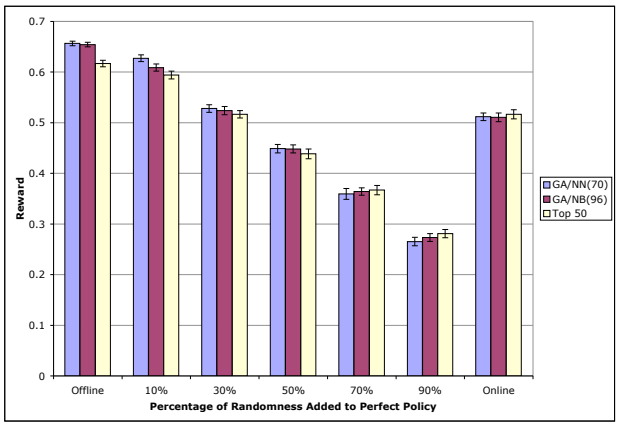

Figure 10: Three different methods of operator set selection evaluated on different control policies. The numbers after the methods are the average (in parentheses) or fixed (not in parentheses) number of operators in the selected sets.

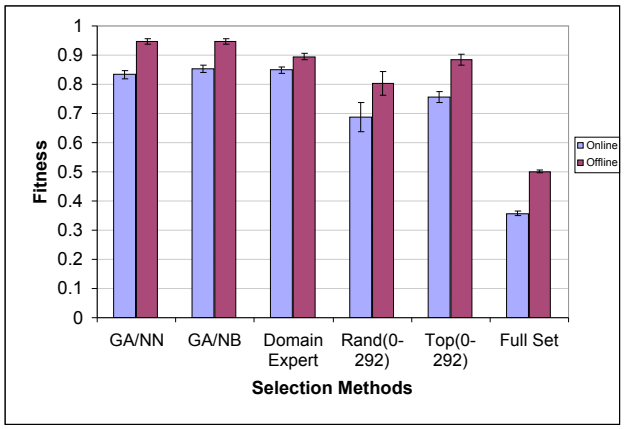

Figure 11: Average on-line and off-line fitness of sets chosen by two GAMM methods, a domain expert, a random method, the Top method, choosing all operators (the full set). (12 cross-validation folds)

their initial population that it deems to be the best for the given task. The meta-models must also be trained using these seed operator sets, in order to ensure these potential solutions are assigned the correct fitness when they are ranked within the GAs. Thus, the seed operator sets are added to the training data used in step 2 (after first obtaining their fitnesses using step 1). In our experiments, we use 10 seed operator sets, to see if the GAs can make use of the information gained through the Top method.

Our implementation of genetic algorithms chooses two chromosomes to replace with two new chromosomes at each generation. A weighted random method is used, so that *usually* lower ranked chromosomes are chosen for replacement. This leaves open the possible disadvantage of breeding out the best solutions found to date, with the advantage of avoiding becoming stuck in a local optimum. Still, it seems counter-intuitive to throw away the best solution(s) found in the search. Elitism can be used to preserve the top n action sets, where $0 < n < P$ (where P is the population size). We experimented with *elitism* of one and three (i.e., the top three solutions cannot be bred out of the population). We chose not to implement seeding in the absence of any elitism, since the seed sets could potentially be bred out quickly. Figure 11 shows the comparison of GAMM to four other operator set selection methods. As representatives of the GA/NN and GA/NB class, we use the average performance of the sets chosen by the best elitism/seeding combination. In both cases, seeding the GAs with 10 operator sets chosen by the Top method and using elitism of one generated the best results. The GAMM-chosen sets have performance on par with the domain expert chosen set, and outperform the other methods.

Table 2 summarizes the experimental results. The three columns correspond to averages over 34 cross validation folds (with seeding and elitism used in GAMM) of image interpretation accuracy, execution cost and fitness on novel images. Image interpretation accuracy is given according to the pixel-level similarity scoring metric defined in section 2. Execution cost is given as the percentage of the execution cost of the full set. Fitness is given as defined at the beginning of this section. Operator sets chosen by GAMM are competitive with that chosen by a domain expert with three years of experience with the system, and greatly outperform the full operator set in terms of both image interpretation accuracy *and* execution cost on novel images.

Table 2: Comparison of the performance of operator sets chosen by GAMM and a domain expert, and the full operator set.

	Accuracy	**Cost**	**Fitness**
GA/NN	0.505	4.7%	0.837
GA/NB	0.513	6.2%	0.836
Domain Expert	0.498	1.5%	0.849
Full Set	0.501	100%	0.357

7. CONCLUSIONS AND FUTURE WORK

In this paper we proposed and evaluated a novel domain-independent method for selecting sets of actions in a Markov decision process. The method, called GAMM, employs machine learning to acquire a meta-model for a specific domain without human intervention. It then conducts an evolutionary search in the space of action sets. Aided by a seeding method called Top and elitism within the GAs, GAMM is able to select highly compact action subsets without virtually any performance loss.

When applied in the realm of adaptive image interpretation systems, GAMM reduced the off-line training time by 95% while retaining the image interpretation quality of the initial massive image processing library. This is a welcome step towards the total automation in the design of such systems as it previously required a domain expert to pick vision operators, and their parameters, by hand.

Given the promising performance of GAMM in the image interpretation domain as well as a navigation task [10], it will be of interest to explore the limits of GAMM's applicability. We are presently developing several formal measures of redundancy in action sets and will attempt to correlate these measures to GAMM's performance.

8. ACKNOWLEDGEMENTS

This paper is an extension of [10]. We appreciate contributions by 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).

9. REFERENCES

[1] D. Culvenor. Tida: an alg. for the delineation of tree crowns in high spatial resolution remotely sensed imagery. *Computers & Geosciences*, 28(1):33–44, 2002.

[2] B. Draper, J. Bins, and K. Baek. ADORE: adaptive object recognition. *Videre*, (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 IJCNN*, Washington, D.C., 1999.

[5] F. Gougeon and D. Leckie. Forest information extraction from high spatial resolution images using an individual tree crown approach. Technical report, Pacific Forestry Centre, 2003.

[6] J. Jarmulak and S. Craw. Genetic algorithms for feature selection and weighting. In *In Proceedings of the IJCAI'99 workshop on Automating the Construction of Case Based Reasoners*, pages 28–33, 1999.

[7] 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.

[8] 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.

[9] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

[10] G. Lee. Automated action set selection in Markov decision processes. Master's thesis, Department of Computing Science, University of Alberta, 2004.

[11] I. Levner and V. Bulitko. Machine learning for adaptive image interpretation. In *Proceedings of the 16th Innovative Applications of Artificial Intelligence '04 conference*, 2004.

[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^{th} 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. *PRL*, 15(11):1119–1125, November 1994.

[15] 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*, Honoloulu, Hawaii, 2002.

[16] R. Sutton and A. Barto. *Reinforcement Learning: An Intro*. MIT Press, 1998.

[17] 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.

[18] 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.

[19] P. Viola and M. Jones. Fast and robust classification using asymmetric adaboost and a detector cascade. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, MA, 2002. MIT Press.

[20] C. Watkins. *Learning from Delayed Rewards*. PhD thesis, King's College, University of Cambridge, UK, 1989.