Automated Feature Extraction for Object Recognition

Ilya Levner, Vadim Bulitko, Lihong Li, Greg Lee, Russell Greiner
University of Alberta
Department of Computing Science
Edmonton, Alberta, T6G 2E8, CANADA
ilya|bulitko|lihong|greglee|greiner@cs.ualberta.ca

September 23, 2003

Abstract

Automated image interpretation is an important task in numerous applications ranging from security systems to natural resource inventorization based on remote-sensing. Recently, a second generation of adaptive machine-learned image interpretation systems have shown expert-level performance in several challenging domains. While demonstrating an unprecedented improvement over hand-engineered and first generation machine-learned systems in terms of cross-domain portability, design-cycle time, and robustness, such systems are still severely limited. This paper reviews the anatomy of the state-of-the-art Multi resolution Adaptive Object Recognition framework (MR ADORE) and presents extensions that aim at removing the last vestiges of human intervention still present in the original design of ADORE. More specifically, feature selection is still a task performed by human domain experts thereby prohibiting automatic creation of image interpretation systems. This paper focuses on autonomous feature extraction methods aimed at removing the need for human expertise in the feature selection process.

Keywords: AI approaches to computer vision, Feature detection and feature extraction, Object recognition.

1 Introduction & Related Research

Image interpretation is an important and highly challenging problem with numerous practical applications. Unfortunately, manually engineering an image interpretation system entails a long and expensive design cycle as well as subject matter and computer vision expertise. Furthermore, hand-engineered systems are difficult to maintain, port to other domains, and tend to perform adequately only within a narrow range of operating conditions atypical of real world scenarios. In response to the aforementioned problems, various *automated* ways of constructing image interpretation systems have been explored in the last three decades [1].

Based on the notion of "goal-directed vision" [2], a promising approach for autonomous system creation lies with treating computer vision as a control problem over a space of image processing operators. Initial systems, such as the Schema System[2], had control policies consisting of *ad-hoc*, hand-engineered rules. While presenting a systemic way of designing image interpretation systems, the approach still required a large degree of human intervention. In the 1990's the second generation of control policy-based image interpretation systems came into existence. More than a systematic design methodology, such systems used theoretically well-founded machine learning frameworks for automatic acquisition of control strategies over a space of image processing operators. The two well-known pioneering examples are a Bayes net system [3] and a Markov decision process (MDP) based system [4].

Our research efforts have focused on extending the latter system, called ADaptive Object REcognition system (ADORE), which learned dynamic image interpretation strategies for finding buildings in aerial images [4]. As with many vision systems, it identified objects (in this case buildings) in a multi-step process. Raw images were the initial input data, while image regions containing identified buildings constituted the final output data; in between the data could be represented as intensity images, probability images, edges, lines, or curves. ADORE modelled image interpretation as a Markov decision process, where the intermediate representations were continuous state spaces, and the vision procedures were actions. The goal was to learn a dynamic control policy that selects the next action (i.e., image processing operator) at each step so as to maximize the quality of the final image interpretation.

As a pioneering system, ADORE proved that a machine learned control policy could be much more adaptive than its hand-engineered counterparts by outperforming any hand-crafted sequence of operators within its library. In addition, the system was quickly ported to recognize stationary (staplers, white-out, etc.) in office scenes and again was shown to outperform

Figure 1: Artificial tree plantations result in simple forest images. Shown on the left is an original photograph. The right image is the desired labeling provided by an expert as part of the training set.

operator sequences designed by human domain experts [5]. This paper discusses the need for hand-crafted features, which prevents the realization of fully autonomous image interpretation systems. The project that investigates approaches to fully autonomous object recognition systems is named MR ADORE for Multi-Resolution ADaptive Object REcognition.

The rest of the paper is organized as follows. First, we review the requirements and design of MR ADORE, in order to demonstrate the critical assumptions made and the resulting difficulties. We then present framework extensions that aim to completely replace domain experts with automated feature selection methods and conclude with a discussion of experimental results and future research directions.

2 MR ADORE: A Brief Overview

In order to increase system portability, the framework of MR ADORE was designed to use readily available off-the-shelf image processing operator libraries (IPLs). 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** in that there is rarely a one-step mapping from input images to final interpretation; instead, a series of operator applications are required to bridge the gap between raw pixels and semantic objects. Examples of the operators include region segmentation, texture filters, and the construction of 3D depth maps. Figure 2 presents graph depicting a partial IPL operator library where nodes represent data types and edges correspond to vision routines transforming input data tokens into output data tokens.

Image interpretation 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. The success of adaptive image interpretation systems therefore depends on the solution to the control problem: for

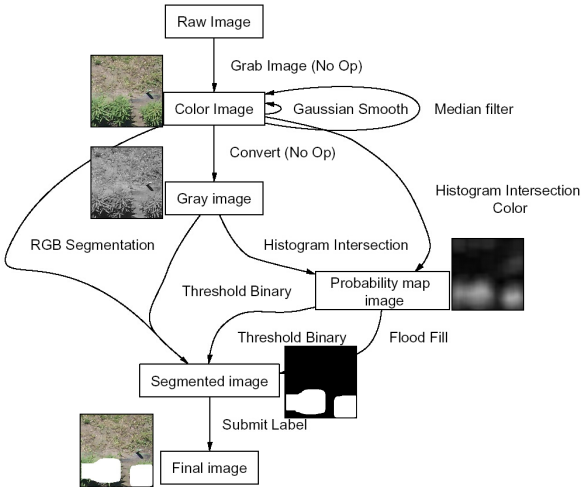


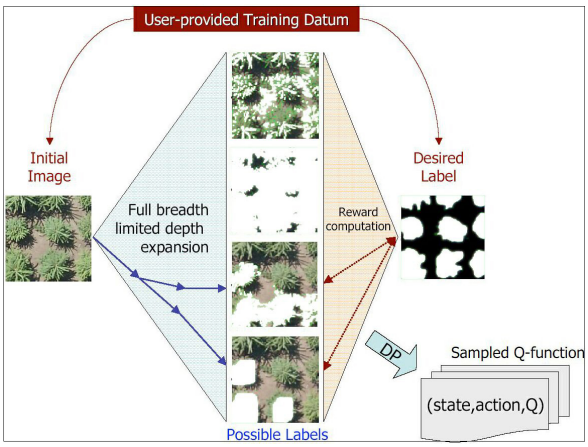
Figure 2: Partial operator graph for the domain of forest image interpretation. The nodes and the corresponding example images depict data processing layers, which in turn describe the *type* of MDP states present with MR ADORE. The edges represent vision routines (actions) that use input data tokens (states) to produce output data tokens.

a given image, what sequence of operator applications will most effectively and reliably interpret the image?

2.1 MR ADORE Operation

MR ADORE starts with a Markov decision process (MDP) [6] as the basic mathematical model by casting the IPL operators as MDP **actions** and the results of their applications (i.e., data tokens) as MDP **states**. In order to recognize a target object, MR ADORE employs the following off-line and on-line machine learning techniques. First, the domain expertise is encoded in the form of training data. Each training datum consists of two images, the input image, and its user-annotated counterpart allowing the output of the system to be compared to the desired image labeling (typically called ground-truth). Figure 1 demonstrates a training pair for the forestry image interpretation domain. Second, during the off-line stage the state space is explored via limited depth expansions of all training image pairs. Within a single expansion all sequences of IPL operators up to a certain user-controlled length are applied to a training image. Since training images are user-annotated with the desired output, terminal rewards can be computed based on the difference between the produced labeling and the desired labeling. System **rewards** are thus defined by creating a scoring metric that evaluates the quality of the final image interpretation with respect to the desired (used-provided) interpretation*. Then, dynamic programming methods [7] are used to

*For the experiments presented, the intersection over union scoring metric, $\frac{A \cap B}{A \cup B}$ is used. This pixel-based scoring metric computes the overlap between the set of hypothesis pixels produced by the



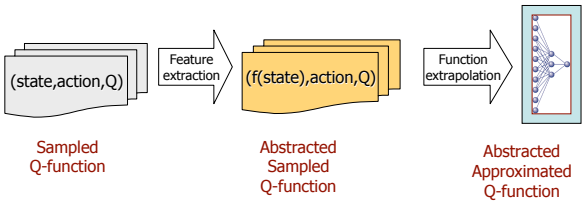


Figure 3: Off-line training phase. **Top:** Exploration of the state space is done by applying all possible operator sequences to a set of training images for which ground truth is provided. By comparing the interpretation resulting from an application of a sequence of operators to the ground truth, each hypothesis is assigned a quality measure (i.e., reward). The rewards are then propagated up the expansion tree in order to calculate Q-values to the intermediate data tokens. **Bottom:** Function approximators are trained on the features extracted from the data tokens produced during the exploration phase.

compute the value function for the explored parts of the state space. We represent the value function as $Q : S \times A \rightarrow R$ where S is the set of states and A is the set of actions (operators). The true $Q(s,a)$ computes the maximum cumulative reward the policy can expect to collect by taking action a in state s and acting optimally thereafter. (Figure 3)

Features (f), used as **observations** by the on-line system component, represent relevant attributes extracted from the unmanageably large states (i.e., data tokens). Features make supervised machine learning methods practically feasible, which in-turn are needed to extrapolate the sampled Q-values (computed by dynamic programming on the explored fraction of the state space) onto the entire space.

Finally, when presented with a novel input image, MR ADORE exploits the machine-learned heuristic value

system A and the set of pixels within the ground-truth image B. If set A and B are identical then their intersection is equal to their union and the score/reward is 1. As the two sets become more and more disjoint the reward decreases, indicating that the produced hypothesis corresponds poorly to the ground-truth.

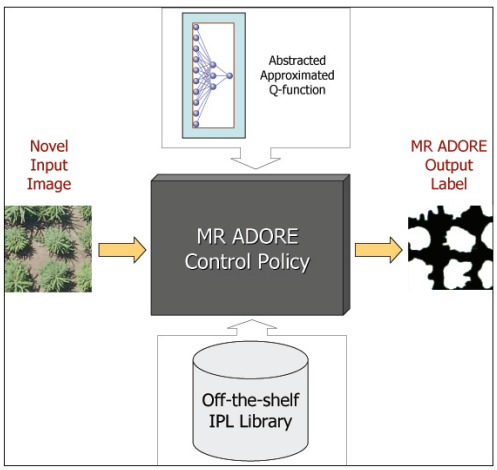


Figure 4: On-line operation. Using the machine leaned Q-function approximators the on-line policy greedily selects the state-action pair expected to yield maximum reward. the process terminates then a interpretation hypothesis is submitted to the user.

function $Q(f(s),a)$ over the abstracted state space, $f(S)$, in order to intelligently select operators from the IPL. The process terminates when the policy executes the action `Submit(`$\langle$*labeling*$\rangle$`)`, which becomes the final output of the system. (Figure 4).

2.2 Adaptive Control Policies

The purpose of the off-line learning phase within MR ADORE is to construct an on-line control policy. While best-first policies are theoretically capable of much more flexibility than static policies, they depend crucially on (i) data token features for *all* levels and (ii) adequate amounts of training data to train the Q-functions for *all* levels. Feature selection/creation can be substantially harder for earlier data processing levels, where the data tokens exhibit less structure [1, 8]. Compounding the problem, a single user-labeled training image delivers exponentially larger numbers of training tuples, $\langle$ state, action, reward $\rangle$, at later processing levels. However, the first processing level gets the mere $|A_1|$ tuples per training image since there is only one data token (the input image itself) and $|A_1|$ actions. As a net result, best-first control policies have been shown to backtrack frequently [4] as well as produce highly suboptimal interpretations [9], due to poor decision making at the top processing layers.

Rather than making control decisions at every level based on the frequently incomplete information provided by imperfect features, the **least-commitment policies** postpone their decisions until more structured and refined data tokens are derived. That is, all operator sequences up to a predefined depth are applied and only then the machine-learned control policy is engaged to select the appropriate action. Doing so allows the control system to make decisions

based on high-quality informative features, resulting in an overall increase in interpretation quality. As a side benefit, the machine learning process is greatly simplified since feature selection and value function approximation are performed for considerably fewer processing levels while benefiting from the largest amount of training data. In [10] such a policy was also shown to outperform the *best* static policy.

3 Automated Feature Extraction

Traditionally, machine learning algorithms require informative features as input in order to encode a control policy. However, manual feature extraction is a tedious process requiring extensive domain and vision expertise. Both, best-first and least-commitment policies, presented in previous sections, need high-quality features at all processing levels. While the least-commitment policy was designed to minimize human intervention, by requiring only a single set of features, the need for domain expertise was only reduced but *not* eliminated. As a result, automatic feature extraction procedures are highly desirable. Unlike previous experiments on policy creation, which used artificial neural networks (ANN) [11] and sparse networks of winnows (SNoW) [12] to induce a control policy, we attempt to use the k-nearest-neighbors (KNN) [13]. In contrast to the previous machine-learned approaches used, the KNN algorithm is a case-based algorithm that approximates the Q-value of an input token based on distance(s) to the nearest training example(s). By defining a distance metric based on pixel differences between input data and training examples, feature extraction can be avoided all together. On the other hand, the distance calculation can be greatly simplified if images can be compressed. By using the principle component analysis (PCA) [14] algorithm to reduce the dimensionality of the input data, the nearest-neighbor calculation is greatly simplified as the projection coefficients act as features describing each training example. Therefore, by treating raw pixels or PCA projection coefficients as features and employing the KNN algorithm it was expected that policy creation could be fully automated.

3.1 Empirical Evaluation

In order to test the aforementioned approaches, the following five policies were implemented:

Static policy Does not use any features. The static policy simply uses the best sequence of operators found at training time.

1-NN + raw pixels One-nearest-neighbor algorithm with a pixel based distance metric.

1-NN + PCA coefficients One-nearest-neighbor algorithm were the distance metric operates on the projection coefficients.

1-NN + hand-crafted features One-nearest-neighbor algorithm using HSV color histograms as features. See [8] for more detail.

ANN + hand-crafted features Artificial neural networks algorithm using HSV color histograms as features.

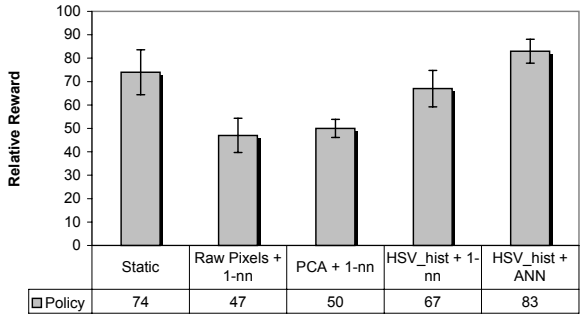
A set of 20 plantation images (see Figure 1 for an example) was randomly split into four sub-sets of five images each. In a cross-validation study, to test each of the five policies, one sub-set of images was used a training set leaving the other three sub-sets for testing. The experimental results are shown in Table 1.

Clearly the one-nearest-neighbor algorithm, using raw pixels and PCA coefficients, does not perform nearly as well as the best static sequence. The following three reasons are thought to account for the poor performance: (a) There are not enough training samples to enable the 1-NN algorithm to approximate rewards well. (b) The 1-NN algorithm coupled with the Euclidean distance metric[†] simply cannot adequately approximate the reward of a hypothesis. (c) The features used by the 1-NN algorithms are not relevant.

In order to establish which of the three aforementioned reasons caused the poor performance, we examine the the performance of policies using 1-NN and artificial neural networks (ANN) together with hand-crafted features. Since the ANN algorithm using hand-crafted features was able to achieve an accuracy of 83%, a lack of training samples cannot explain the poor performance of the 1-NN algorithm coupled with PCA coefficients or raw pixels as features. On the other hand, compared to ANN coupled with hand crafted features, the drop in performance of the 1-NN algorithm using the same hand-crafted features implies that nearest-neighbor algorithm is a poor reward approximator. Finally, compared to the performance of the 1-NN algorithm using hand-crafted features with PCA coefficients or raw pixels, it is clear that the automatic features we have defined need to be developed further in order for machine learning algorithms to approximate rewards. In summary, the inferior performance of the 1-NN algorithm coupled with automated feature creation techniques, such as raw pixels or PCA projection coefficients, is a function of both the features and the approximation algorithm used. While changing the machine learning mechanism is relatively easy, automat-

[†]The results shown in Table 1 represent the best results obtained by using a Euclidean distance metric. Additionally we have used Manhattan, Angle and Maholonobis distance metrics, which did not outperform the Euclidean distance metric. Finally, we varied the number of nearest neighbors from 1-5 with 1-NN outperforming all other algorithm settings.

Table 1: Performance results of policies using PCA coefficients and raw Pixels as features compared to static policy and policies using hand-crafted features. For all experiments the 1-NN algorithm used the Euclidean distance metric to approximate rewards. The results shown are relative to off-line optimal performance. In other words the table represents $\frac{\text{policy reward}}{\text{optimal off-line reward}} * 100$.



ically determining relevant features or removing features altogether remains a difficult problem.

4 Future Directions

The analysis in the previous section implies that very few labeled training examples are needed in order for a machine learned function approximator to evaluate a segmentation hypothesis. On the other hand, the k-nearest-neighbor algorithm should perform better (unless all possible input images have been seen) as the number of training examples increases. Initial experiments with a $34:1$ train/test ratio using 1-NN and hand-crafted features did not produce a significant increase in performance. Perhaps this counterintuitive result is a clue to the poor performance of the 1-NN algorithm. Our conjecture is that the input images used exhibit far too much internal variation that cannot not captured by the machine learning algorithms. Figure 1 clearly shows several target objects present in one input image. By splitting the images into several, possibly overlapping, sub-images, each focused on a single object, the performance of the system may be improved by allowing the scene variance to be learned by the function approximators. In order to do so focus-of-attention mechanisms will need to be added to the system. Incremental PCA methods, proposed in [15], will need to be implemented in order for a larger training corpora, created by splitting the input images, to be used in the future experiments. In addition much more efficient KNN algorithms need to be used to effectively deal with the increase in the training data. Approaches such as finding approximate neighbors [16] or using kd-trees [17] as efficient search structures offer prospective starting points. In addition, adaptive distance metrics need to be designed that would enable a more accurate reward approximation. For example, the use of a weighted Euclidean [18] metric may improve the performance of the KNN algorithm by removing the influence of irrelevant features (in our case removing the influence of PCA coefficients that are not relevant to the task of calculating rewards).

5 Conclusions

Conventional ways of developing image interpretation systems often require that system developers posses significant subject matter and computer vision expertise. The resulting knowledge-engineered systems are expensive to upgrade, maintain, and port to other domains.

More recently, second-generation image interpretation systems have used machine learning methods in order to (i) reduce the need for human input in developing an image interpretation system or port it to a novel domain and (ii) increase the robustness of the resulting system with respect to noise and variations in the data.

This paper presented a state-of-the-art adaptive image interpretation system called MR ADORE. We then reported on the performance of feature extraction methods aimed at completely eliminating the need to hand-craft features. The performance of instance-based function approximators using automatically constructed features (raw pixels or PCA coefficients) was shown to be inferior compared to the policies using hand-crafted features. A possible solution to the feature extraction problem may perhaps come in the form of feature extraction libraries, similar in nature to the operator libraries currently used by MR ADORE. Just as MR ADORE learned efficient operator selection, the next generation object recognition systems may need to *select* which features are relevant [19] for a given processing level through off-line trial-and-error processes based, perhaps, on the very same MDP framework used today to efficiently select operator sequences. Clearly if such an approach is to be employed a number of feature selection algorithms, such as filter-based Relief [20, 21] or wrapper-based [22] methods could be readily employed to remove the last vestiges of human intervention from the outlined framework for automatically constructing object recognition systems.

Acknowledgements

Bruce Draper participated in the initial MR ADORE design stage. Lisheng Sun, Yang Wang, Omid Madani, Guanwen Zhang, Dorothy Lau, Li Cheng, Joan Fang, Terry Caelli, David H. McNabb, Rongzhou Man, and Ken Greenway have contributed in various ways. We are grateful for the funding from the University of Alberta, NSERC, and the Alberta Ingenuity Center for Machine Learning.

References

[1] Bruce A. Draper. From knowledge bases to Markov models to PCA. In *Proceedings of Workshop on Computer Vision System Control Architectures*, Graz, Austria, 2003.

[2] B. Draper, A. Hanson, and E. Riseman. Knowledge-directed vision: Control, learning and integration. *Proceedings of the IEEE*, 84(11):1625–1637, 1996.

[3] R. Rimey and C. Brown. Control of selective perception using bayes nets and decision theory. *International Journal of Computer Vision*, 12:173–207, 1994.

[4] B. Draper, J. Bins, and K. Baek. ADORE: adaptive object recognition. *Videre*, 1(4):86–99, 2000.

[5] B. Draper, U. Ahlrichs, and D. Paulus. Adapting object recognition across domains: A demonstration. In *Proceedings of International Conference on Vision Systems*, pages 256–267, Vancouver, B.C., 2001.

[6] R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2000.

[7] A. G. Barto, S. J. Bradtke, and S. P. Singh. Learning to act using real-time dynamic programming. *Artificial Intelligence*, 72(1):81–138, 1995.

[8] I. Levner. Multi resolution adaptive object recognition system: A step towards autonomous vision systems. Master's thesis, Department of Computer Science, University of Alberta, 2003.

[9] Vadim Bulitko and Ilya Levner. Improving learnability of adaptive image interpretation systems. Technical report, University of Alberta, 2003.

[10] I. Levner, V. Bulitko, G. Lee, L. Li, and R. Greiner. Towards automated creation of image interpretation systems. In *Australian Joint Conference on Artificial Intelligence (To appear)*, 2003.

[11] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Macmillian College Pub. Co., 1994.

[12] S. Agarwal and D. Roth. Learning a sparse representation for object detection. In *7th European Conference on Computer Vision*, volume 4, pages 113–130, Copenhagen, Denmark, 2002.

[13] Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York, 1973.

[14] Michael Kirby. *Geometric Data Analysis: An Emprical Approach to Dimensionality Reduction and the Study of Patterns*. John Wiley & Sons, New York, 2001.

[15] P.M.Hall, R.R.Martin, and A.D. Marshall. Merging and splitting eigenspace models. *IEEE Trans. on Pattern Analysis & Machine Intelligence*, 22(9):1042–1050, 2000.

[16] Sunil Arya, David M. Mount, Nathan S. Netanyahu, Ruth Silverman, and Angela Y. Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM*, 45(6):891–923, 1998.

[17] Dan Pelleg and Andrew Moore. Accelerating exact k-means algorithms with geometric reasoning. In Surajit Chaudhuri and David Madigan, editors, *Proceedings of the Fifth International Conference on Knowledge Discovery in Databases*, pages 277–281. AAAI Press, aug 1999.

[18] R. Kohavi, P. Langley, and Y. Yun. The utility of feature weighting in nearest-neighbor algorithms, 1997.

[19] Vadim Bulitko, Greg Lee, and Ilya Levner. Evolutionary algorithms for operator selection in vision. In *Proceedings of the Fifth International Workshop on Frontiers in Evolutionary Algorithms*, 2003.

[20] 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.

[21] J. Bins and B. Draper. Feature selection from huge feature sets. In *Proceedings of International Conference on Computer Vision*, volume 2, pages 159–165, 2001.

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