Efficient Interpretation Policies

Ramana Isukapalli
101 Crawfords Corner Road
Lucent Technologies
Holmdel, NJ 07733 USA
ramana@research.bell-labs.com

Russell Greiner
Department of Computing Science
University of Alberta
Edmonton, AB T6G 2E8 Canada
greiner@cs.ualberta.ca

Abstract

Many imaging systems seek a good interpretation of the scene presented — *i.e.*, a plausible (perhaps optimal) mapping from aspects of the scene to real-world objects. This paper addresses the issue of finding such likely mappings *efficiently*. In general, an "(interpretation) policy" specifies when to apply which "imaging operators", which can range from low-level edge-detectors and region-growers through high-level token-combination–rules and expectation-driven object-detectors. Given the costs of these operators and the distribution of possible images, we can determine both the expected cost and expected accuracy of any such policy. Our task is to find a maximally effective policy — typically one with sufficient accuracy, whose cost is minimal. We explore this framework in several contexts, including the eigenface approach to face recognition. Our results show, in particular, that policies which select the operators that maximize *information gain per unit cost* work more effectively than other policies, including ones that, at each stage, simply try to establish the putative most-likely interpretation.

Keywords: vision, decision theory, real time systems

1 Introduction

Interpretation, *i.e.*, assigning semantically meaningful labels to relevant regions of an image, is the core process underlying a number of imaging tasks, including recognition ("is objectX in the image?") and identification ("which object is in the image?"), as well as several forms of tracking ("find all moving objects of typeX in this sequence of images"), etc. [PL95; HR96]. Of course, it is critical that interpretation systems be *accurate*. It is typically important that the interpretation process also be *fast*: For example, to work in real-time, an interpreter examining the frames of a motion picture will have only 1/24 of a second to produce an interpretation. Or consider a web-searcher that is asked to find images of, say, aircraft. Here again speed is critical — and most searchers do in fact sacrifice some accuracy to gain efficiency (*i.e.*, they quickly return a large number of "hits", only some of which are relevant). This paper addresses the challenge of producing an interpreter that is both sufficiently accurate and sufficiently efficient.

Section 2 provides the framework, showing how our framework generalizes the standard (classical) approaches to image interpretation, then providing a formal description of our task: given a distribution of possible images and an inventory of "operators", produce a "policy" that specifies when to apply which operator, towards optimizing some user-specified objective function. It describes three different policies that could be used, using a simple blocks-world example to illustrate these terms. The rest of this paper demonstrates that one of these policies, "INFOGAIN" (which uses information gain to myopically decide which operator is most useful at each step), is more effective than the other obvious contenders. Section 3 provides an empirical comparison of these approaches in the context of the simple blocks-world situation. Section 4 extends these ideas to deal with face recognition, using the modern eigenvector approach. (This complements Section 3's classical approach to interpretation.) Section 5 quickly surveys some related work. While the results in this paper demonstrate the potential for this approach, they are still fairly preliminary. Section 6 discusses some additional issues that have to be addressed towards scaling up this system.

2 Framework

2.1 Standard Approaches

There are many approaches to scene interpretation. A strictly "bottom-up" classical approach performs a series of passes over *all* of the information in the scene, perhaps going first from the pixels to edgels, then from edgels to lines, then to regions boundaries and then to descriptions, until finally producing a consistent interpretation for the entire scene. Most "top down", or "model driven", systems likewise begin by performing several bottom-up "sweeps" of the image — applying various low-level processes to the scene to produce an assortment of higher-level tokens, which are then combined to form some plausible hypothesis (*e.g.*, that the scene contains a person, etc.). These systems differ from strictly bottom-up schemes by then switching to a "top-down" mode: given sufficient evidence to support one interpretation, they seek scene elements that correspond to the parts of the proposed real-world object that have yet to be found [LAB90].

Notice the model-based systems have more prior knowledge of the scene contents than the strictly bottom-up schemes — in particular, they have some notion of "models" — which they can exploit to be more efficient. We propose going one step further, by using additional prior knowledge to further increase the efficiency of an interpretation system. Consider a trivial situation in which we only have to determine whether or not a "red fire-engine" appears in an image; and imagine, moreover, we knew that the *only* red object that might appear in our images is a fire-engine. Here it is clearly foolish to worry about line-detection or region-growing; it is sufficient, instead, to simply sweep the image with an inexpensive "red" detector. Moreover, if we knew that the fire-engine would only appear in the bottom third of the image, we could apply this operator only to that region.

This illustrates the general idea of exploiting prior knowledge (*e.g.*, which objects are we seeking, as well as the distribution over the objects and views where they might appear) to produce an effective interpretation process. In general, we will assume our interpretation system also has access to the inventory of possible imaging operators. Given this collection of operators, an "interpretation policy" specifies when and how to apply which operator, to produce an appropriate interpretation of an image.

Our objective is to produce an *effective* interpretation policy — *e.g.*, one that *efficiently* returns a *sufficiently accurate* interpretation, where accuracy and efficiency are each measured with respect to the underlying task and the distribution of images that will be encountered. Such policies must, of course, specify the details: perhaps by specifying exactly which bottom-up operators to use, and over what portion of the image, if and when to switch from bottom-up to top-down, which aspects of the model to seek, etc. These policies can include "conditionals"; *e.g.*, "terminate on finding a red object in the scene; otherwise run procedure X". They may also specify applying a *particular* operator only to *specified* regions of the image (*e.g.*, seek only near-horizontal edges, only in the upper left quadrant of the image). Based on the information available, this interpretation policy could then use other operators, perhaps on other portions of the image to further combine the tokens found.

2.2 Input

We assume that our interpretation system "*IS*" is given the following information:

⋆ The **distribution** of images that the *IS* will encounter, encoded in terms of the distribution of objects and views that will be seen, etc. (Here we assume this information is explicitly given to the algorithm; we later consider acquiring this by sampling over a given set of training images.)

As a trivial example, we may know that each scene will contain exactly 25 sub-objects, each occupying a cell in a 5×5 grid; see Figure 1. Each of these cells has a specified "color", "texture" and "shape", and each of these properties ranges over 4 values. (Hence, we can identify each image with a $5 \times 5 \times 3 = 75$ tuple of values, where each value is from $\{1, 2, 3, 4\}$.) Moreover, our *IS* knows the distribution over these 4^{75} possible images; see below.

⋆ The **task** includes two parts: First, what objects the *IS* should seek, and what it should return — *e.g.*, "is there an airplane in image" or "is there a DC10 centered at $[\,43, 92\,]$ in the image", etc. *In our trivial blocks-world case, we simply want to know which of the images is being examined.*

Second, the task specification should also specify the "evaluation criteria" for any policy, which is based on both the expected "accuracy" and its expected "cost". In general, this will be a constrained optimization task, combining both hard constraints and an optimization criteria (*e.g.*, minimize some linear combination of accuracy and cost, or perhaps maximize the likelihood of a correct interpretation, for

$c_{1,1}, t_{1,1}, s_{1,1}$	$c_{2,1}, t_{2,1}, s_{2,1}$			
		$c_{3,3}, t_{3,3}, s_{3,3}$		
				$c_{5,5}, t_{5,5}, s_{5,5}$

Figure 1: A simple image of 25 sub-objects

a given bound on the expected cost).

For this blocks-world task, we want to minimize the expected cost and also have 100% *correctness, assuming the operators are perfect.*

⋆ The **set of possible "operators"** includes (say) various edge detectors, region growers, graph matchers, etc. For each operator, we must specify

- its input and output, of the form: "given a set of pixel intensities, returns tokens representing the regions of the same color";
- its "effectiveness", which specifies the accuracy of the output, as a function of the input. This may be a simple "success probability", or could be of the form: "assuming noise-type W, can expect a certain ROC curve" [RH92];
- its "cost", as a function of (the size of) its input and parameter setting.

When used, each operator may be given some arguments, perhaps identifying the subregion of the image to consider.

Here, we consider three operators: O_1 *(resp.,* O_2*,* O_3*) for detecting the "value" of color (resp., "texture", "shape"); each mapping a* $[\,x, y\,]$ *location of the current image to a value in* $\{1, 2, 3, 4\}$. *(Note that location is an argument to the operator.) We assume that each operator, when pointed at a particular "cell", will reliably determine the actual value of that property at the specified location, and will do so with unit cost.* (Section 4 considers less trivial operators.)

For each situation, we assume our interpreter will be given a series of scenes, but will always have the same objective and criteria; *e.g.*, it is expected to look for the same objects in each image, and has a single objective function. (It is easy to generalize this to deal with environments that can ask different questions for different images, and impose different costs.)

At each stage, our *IS* will use its current knowledge (both prior information — *e.g.*, associated with the distribution and the operators — and information obtained by earlier probes) to decide whether (1) to terminate, returning some interpretation; or (2) to perform some operation, which involves specifying both the appropriate operator and the relevant arguments to this operator, and then recur.

2.3 Policies

In general, we could represent an *IS* explicitly as a large decision tree, whose leaf nodes each represent a complete interpretation (which is returned as the result of the *IS*), and whose internal nodes each correspond to a sequence of zero or more operator applications, followed by a test on the original data and/or some set of inferred tokens. Each arc descending from this node is labeled with a possible result of this test, and descends to new node (containing other operators and tests) appropriate for this outcome.

$IS_\chi(T = \langle C_{max}, P_{min} \rangle$: TaskSpecification,
$P(\ )$: Distribution, $O = \{o_i\}$: Operators, I: Image)
Initialize cost $C := 0$ Evidence $\vec{O} = \langle\rangle$
While $[C < C_{max}$ & $P_{min} > \max_x P(\text{Obj} = x \mid \vec{O})]$ do
Select $[o :$ operator; $a :$ arguments$]$ (based on policy χ, $P(\cdot)$)
(Note a may specify the region to consider)
Apply $o(a)$ to I, yielding v
Extend $\vec{O} := \vec{O} + \langle o(a), v \rangle$
Update $P(\cdot \mid \vec{O})$, based on result
Update $C := C + \text{Cost}[\,o(a)\,]$
Return Best Interpretation: $\text{argmax}_x P(\text{Obj} = x \mid \vec{O})$

Figure 2: Identification algorithm,
for policy $\chi \in$ {RANDPOL, BESTHYP, INFOGAIN}

Given that such explicit strategy-trees can be enormous, we instead represent strategies *implicitly*, in terms of a "*policy*" that specifies how to decide, at run-time, which operator to use. Figure 2 shows a general interpretation strategy using any of the policies. We will consider the following three policies:[1]

Policy RANDPOL**:** selects an operator[2] randomly.

Policy BESTHYP**:** first identifies the object that is most likely to be in the scene (given the evidence seen so far, weighted by the priors, etc.) and then selects the operator that can best verify this object [LHD+93, p370]: That is, after gathering information from k previous operators $\vec{O} = \langle o_1 = v_1, \ldots, o_k = v_k \rangle$, it computes the posterior probabilities of each possible interpretation s_i, $P(S = s_i \mid \vec{O})$. To select the next operator, BESTHYP will first determine which of the scenes is most likely — *i.e.*, $s^* = \text{argmax}_s\{P(S = s \mid \vec{O})\}$ — and then determines which operator has the potential of increasing the probability of this interpretation the most: Assume the operator o returns a value in $\{v_1, v_2 \ldots, v_j\}$; then o might increase the probability of s^* to $\text{best}(s^*, o) = \max_j\{P(S = s^* \mid \vec{O}, o = v_j)\}$. Here, BESTHYP will use the operator

$$o^*_{BH} \quad = \quad argmax_o\{\ \text{best}(s^*, o)\ \}$$

Policy INFOGAIN**:** selects the operator that provides the largest information gain (per unit cost) at each time. This policy computes, for each possible operator and argument combination o, the expected information gained by performing this operation: $EIG(S, \vec{O})o =$

$$H(S|\vec{O}) - \sum_j P(o = v_j \mid S, \vec{O})\, H(S|\vec{O}, o = v_j)$$

[1]We view RANDPOL as a baseline; clearly we should not consider any system that does worse that this. These empirical studies are designed to test our hypothesis that INFOGAIN will actually work best in practice — and in particular, work better than BESTHYP, which is actually being used in some deployed applications [LHD+93].

[2]We will use the term "operator" to refer to the operator *instantiated with the relevant arguments*. Also, we will further abuse notation by writing $o_i = v_i$ to mean that the value v_i was obtained by applying the (instantiated) operator o_i to the image.

Figure 3: Tail lights of Chevrelot

where $H(S|E) = \sum_i P(S = s_i \mid E) \log P(S = s_i \mid E)$ is the entropy of the distribution over the interpretations S given the evidence E, for $E = \vec{O}$ or $E = \{\vec{O}, o = v_j\}$.

INFOGAIN then uses the operator

$$o^*_{IG} \quad = \quad argmax_o\{\ EIG(\vec{O}, o)\ /C(o)\ \}$$

that maximizes $[EIG(\vec{O}, o)/C(o)]$, where $C(o)$ is the cost of applying the operator.

3 Simple Experiments: Blocks World

This section presents some experiments using the simple blocks world situation presented above. They are designed simply to illustrate the basic ideas, and to help us compare the three policies described earlier. Section 4 below considers a more realistic situation.

We first generated a set of 1000 images, each with 25 sub-objects, by uniformly assigning values for color, texture and shape to each of the sub-objects randomly, for each of 1000 images; we also assigned each a "prior distribution" p_i to these images (this corresponds to taking an empirical sample, with replacement). For each run, we randomly select one of the 1000 images to serve as a target for identification, then used each of the three policies to identify the image.

After observing $O_k = \{o_1 = v_1, o_2 = v_2, \ldots, o_k = v_k\}$ from the operators in the first k iterations, RANDPOL randomly selects a cell $C[i,j]_{RP}$ and an operator o_{RP} to probe the value for a property (color, texture etc), insisting only that o_{RP} was not tried earlier on $C[i,j]_{RP}$ in any previous iterations of this run. BESTHYP chooses a cell $C[i,j]_{BH}$ and an operator o_{BH} to maximize the posterior probability of the most likely image, as explained earlier. Finally, INFOGAIN chooses $C[i,j]_{IG}$ and o_{IG} such that $EIG(\vec{O}, o_{IG})/C(o_{IG})$ is the maximum over all possible cell and operator combinations. For each of these policies, the posterior probability is updated after applying the chosen operator on the cell. The process is repeated until the image is identified — *i.e.*, all other contenders are eliminated.

We considered 10 set-ups (each with its own objects and p_i's), and performed 5 runs for each set-up. Over these 50 runs, RANDPOL required on average 5.82 ± 0.27 probes, BESTHYP 5.44 ± 0.32 probes and INFOGAIN 5.32 ± 0.13. INFOGAIN is statistically better than the other two policies, at the $p < 0.1$ level.

We then performed a variety of other experiments in this domain, to help quantify the relative merits of the different policies — *e.g.*, in terms of the "average Hamming distances" between the images. See [IG01] for details.

While this particular task is quite simplistic, we were able to use the same ideas for the more interesting task of identifying the make and model of a car (*e.g.*, Toyota Corolla, Nissan Sentra, Honda Civic, etc.) given an image of the

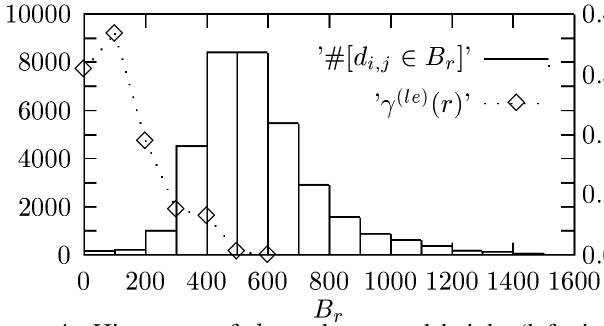

Figure 4: Histogram of $d_{i,j}$ values; total height (left-size scale) reflects number of $\langle i, j\rangle$ pairs in bucket B_r. (le feature; $k = 25$.) Also shows $\gamma^{(le)}(r)$, using right-size scale.

car that shows its "rear tail lights assembly"; see Figure 3. Again see [IG01] for details.

4 Scaling Up: Face Recognition

We next investigate the efficiency and accuracy of the three policies in the more complicated domain of "face recognition" [TP91; PMS94; PWHR98; EC97]. This section first discusses the prominent "eigenface" technique of face recognition that forms the basis of our approach; then presents our framework, describing the representation and the operators we use to identify faces; then presents our face interpretation algorithm; and finally shows our empirical results.

4.1 Eigenface, and EigenFeature, Method

Many of today's face recognition systems use *Principal Component Analysis* (PCA) [TP91]: Given a set of training images of faces, the system first forms the covariance matrix Σ of the images, then computes the k main eigenvectors of Σ, called "eigenfaces". Every training face h_i is then projected into this coordinate space ("facespace"), producing a vector, $\Omega_i = [\omega_1^{(i)}, \omega_2^{(i)}, \ldots, \omega_k^{(i)}]$.

During recognition, any test face h_{test} is similarly projected into the facespace, producing the vector Ω_{test}, which is then compared with each of the training faces. The best matching training face is taken to be the interpretation [TP91].

Following [PMS94], we extend this method to recognize facial features — eyes, nose, mouth, etc. — which we then use to help identify the individual in a given test image: We first partition the training data into two sets, $T = \{h_{1,i}\}$ (for constructing the eigenfeatures) and $S = \{h_{2,i}\}$ (for collecting statistics — see below), which each contains at least one face of each of the n people. Using id(h) to denote the person whose face is given by h, we have $\mathrm{id}(h_{1,i}) = i = \mathrm{id}(h_{2,i})$ for $i = 1..n$; each remaining $h_{1,j}$ and $h_{2,j}$ ($j > n$) also maps to $1..n$.

We use PCA on, say, the mouth regions of each T image, to produce a set of eigenvectors; here *eigen-mouths*. For each face image h_i, let $\Omega_i^{(m)}$ be "feature space" encoding of h_i's mouth-region. We will later compare the feature space encoding $\Omega_{test}^{(m)}$ of a new image h_{test} against these $\{\Omega_i^{(m)}\}$ vectors, with the assumption that $\Omega_{test}^{(m)} \approx \Omega_i^{(m)}$ suggests that h_{test} is really person i — *i.e.*, finding that $\|\Omega_{test}^{(m)} - \Omega_i^{(m)}\|$ is small should suggest that $\mathrm{id}(h_{test}) = i$. (Note $\|\cdot\|$ refers to the L_2 norm, aka Euclidean distance.)


Figure 5: Training images (top); test images (bottom)

To quantify how strong this belief should be, we compute $M = |T| \times |S|$ values $\{d_{i,j}\}$ where each $d_{i,j} = \|\Omega_{1,i}^{(m)} - \Omega_{2,j}^{(m)}\|$ is the Euclidean distance between the "eigen-mouth encodings" of T's $h_{1,i}$ and S's $h_{2,j}$. We considered 16 buckets $B_r \subset \Re$ for these $d_{i,j}$ values: $B_0 = [0, 100)$, $B_1 = [100, 200)$, ..., $B_{14} = [1400, 1500)$, $B_{15} = [1500, \infty)$. Then, for each bucket B_r, we estimate $P(\, d_{i,test} \in B_r \,|\, \mathrm{id}(h_{test}) = i\,)$ as

$$\gamma^{(m)}(r) \;=\; \frac{\#[\, d_{i,j} \in B_r \;\&\; \mathrm{id}(h_{1,i}) = \mathrm{id}(h_{2,j})\,]}{\#[\,\mathrm{id}(h_{1,i}) = \mathrm{id}(h_{2,j})\,]}$$

where $\#[\,\mathrm{id}(h_{1,i}) = \mathrm{id}(h_{2,j})\,]$ is the number of $\langle i, j\rangle$ pairs where $h_{1,i} \in T$ is the same person as $h_{2,j} \in S$. (We used the obvious Laplacian correction to avoid using 0s here [Mit97].) We also compute

$$\rho^{(m)}(r) \;=\; \frac{\#[\, d_{i,j} \in B_r\,]}{|S| \times |T|}$$

to estimate $P(\, d_{i,test} \in B_r\,)$. (Note this is the average over *all* images i in the test set T.) Figure 4 shows a histogram of the $d_{i,j}$ values, using the 16 buckets for the left eye feature (see below); it also shows the values of $\gamma^{(le)}(r)$ for $r = 0..6$.

We use these $\{\gamma^{(m)}(r)\}_r$ and $\{\rho^{(m)}(r)\}_r$ values to interpret a new test image of a person's face h_{test}. (While the specific image h_{test} is not in $T \cup S$, it is another face of someone who has other faces in $T \cup S$).) We first project h_{test}'s mouth region onto the "eigen-mouth"'s space, forming the vector $\Omega_{test}^{(m)}$, then compare $\Omega_{test}^{(m)}$ with the stored eigen-mouth projections (from T) — computing the values $d_{i,test} = \|\Omega_i^{(m)} - \Omega_{test}^{(m)}\|$ for each i in T. This $d_{i,test}$ will be in some bucket, say $B_r = [r, r + 100)$. We then use Bayes Rule to compute the probability that this face is person i:

$$\begin{aligned}
& P(\,\mathrm{id}(h_{test}) = i \,|\, \Omega_{test}^{(m)}, \{\Omega_j^{(m)}\}_j\,) \\
&= P(\,\mathrm{id}(h_{test}) = i \,|\, d_{i,test} \in B_r\,) \\
&= \frac{P(\, d_{i,test} \in B_r \,|\, \mathrm{id}(h_{test}) = i\,)\; P(\,\mathrm{id}(h_{test}) = i\,)}{P(\, d_{i,test} \in B_r\,)} \\
&\approx \gamma^{(m)}(r) \times \tfrac{1}{n} / \rho^{(m)}(r)
\end{aligned} \tag{1}$$

(Here, we assume the faces are drawn from the n individuals uniformly; hence $P(\,\mathrm{id}(h_{test}) = i\,) = 1/n$.)

So far, we considered only a *single* feature — here, "mouth projections", as indicated by the (m) superscript. We similarly compute $\gamma^{(n)}(r)$, $\gamma^{(le)}(r)$, $\gamma^{(re)}(r)$, values associated with the nose, left-eye and right-eye, as well as the $\rho^{(n)}(r)$, $\rho^{(le)}(r)$, $\rho^{(re)}(r)$ values.

We then used the Naïve-Bayes assumption [DH73] (that features are independent, given the specified person) to essentially simply multiply the associated probabilities: Assume we observed, for each feature f, $\|\Omega_{test}^{(f)} - \Omega_i^{(f)}\| \in B_{r_f}$, then

$$\begin{aligned} & P(\,\mathrm{id}(h_{test}) = i \,|\, \Omega_{test}^{(m)},\, \Omega_{test}^{(le)},\, \Omega_{test}^{(n)},\, \{\Omega_i^{(f)}\}_{f,i}\,) \\ &\quad = \alpha \cdot P(\,\Omega_{test}^{(m)}, \Omega_{test}^{(le)}, \Omega_{test}^{(n)} \,|\, \mathrm{id}(h_{test}) = i\,) \\ &\quad = \alpha \cdot P(\,\Omega_{test}^{(m)} \,|\, \mathrm{id}(h_{test}) = i\,) \cdot \\ &\quad\quad P(\,\Omega_{test}^{(le)} \,|\, \mathrm{id}(h_{test}) = i\,) \cdot P(\,\Omega_{test}^{(n)} \,|\, \mathrm{id}(h_{test}) = i\,) \\ &\quad \approx \quad \alpha' \cdot \frac{\gamma^{(m)}(r_m)}{\rho^{(m)}(r_m)} \cdot \frac{\gamma^{(le)}(r_{le})}{\rho^{(le)}(r_{le})} \cdot \frac{\gamma^{(n)}(r_n)}{\rho^{(n)}(r_n)} \qquad (2) \end{aligned}$$

where α, α' are scaling constants (as the prior is uniform).

(Note: we had also tried computing individual $\gamma_i^{(m)}(r)$ values, specific to each training face $h_{1,i}$. However, we found this was too noisy, as the number of relevant instances was too small.)

4.2 Framework

The **distribution** is the set of all people who can be seen, which varies over race, gender and age, as well as poses and sizes; we approximate this using the images given in the training set. We assume that any test face-image belongs to one of the people in the training set, but probably with a different facial expression or in a slightly different view, and perhaps with some external features not in the training image (like glasses, hat, etc.), or vice versa. Figure 5 shows three training images (top) and four test images (bottom).

Our **task** is to identify the person from his/her given test image h_{test} (wrt the people included in the training set), subject to the minimum acceptable accuracy (P_{min}) and the maximum total cost of identification (C_{max}).

We use four classes of **operators**, $O = \{\, o_{le}(k),\ \ o_{re}(k),\ \ o_n(k),\ \ o_m(k)\,\}$ to detect respectively "left eye", "right eye", "nose" and "mouth". Each specific operator also takes a parameter k which specifies the size of the feature space to consider; here we consider $k \in \{25, 30, 35, 40, 45\}$. As discussed above, each instantiated operator $o \in O$ takes an input the test image of a face h_{test}, and returns a probabilistic distribution over the individuals.

Each operator $o_f(k)$ (associated with the feature $f \in \{le,\ re,\ n,\ m\}$) performs three subtasks: SubTask#1 locates the feature f_{test} from within the entire face h_{test}. Here we use a simple template matching technique in which we search in a fixed "window" of size $P \times Q$ pixels in h_{test} for any given feature, of size $p < P$ by $q < Q$ pixels. SubTask#2 then projects the relevant region f_{test} of the test image into the feature space — computing $\Omega_{test}^{(f)}$ of dimension k. SubTask#3 uses this $\Omega_{test}^{(f)}$ to compute first the values $d_{i,test} = \|\Omega_i^{(f)} - \Omega_{test}^{(f)}\|$ for each person i, then place each $d_{i,test}$ value into the appropriate B_r bucket, and finally compute the probability $P(\,\mathrm{id}(h_{test}) = i \,|\, \Omega_{test}^{(f)}\,)$ for each person i, using Equation 1 (possibly augmented with Equation 2 to update the distribution when considering the 2nd and subsequent features); see Section 4.1. For each eigenspace dimension k, we empirically identified the cost (in seconds) of the four classes of operators — $C(o_{le}(k)) = 0.65+(k-25)\times 0.021$, $C(o_{re}(k)) = 0.87+(k-25)\times 0.015$, $C(o_n(k)) = 1.54 + (k-25) \times 0.025$ and $C(o_m(k)) = 1.23 + (k-25) \times 0.04$. While increasing the dimensionality k of the feature space should improve the accuracy of the result, here we see explicitly how this will also increase the cost.

4.3 Interpretation Phase

During interpretation, each current policy (RANDPOL, BESTHYP and INFOGAIN) iteratively selects an operator $o(k) \in O$. RANDPOL chooses an operator o_{RP} and a value k_{RP} randomly, subject only to the condition that o_{RP} had not been tried before on the image; BESTHYP first identifies the most likely person $i = \mathrm{argmax}\, P(\,\mathrm{id}(h) = i \,|\, \cdot\,)$, then choses the instantiated $o_{BH}(k_{BH})$ operator that best confirms this hypothesis (that the image belongs to i person), provided this o_{BH} had not been used earlier for this image; and INFOGAIN chooses an instantiated operator $o = o_{IG}(k_{IG})$ that has the maximum $EIG(\,\cdot\,, o\,)/C(o))$ value. In each case, the operator is applied to the appropriate region in the given test face image and the distribution is updated... until one face is identified with sufficiently high probability ($> P_{min}$) or the system fails (by exhausting all the possible operators, or having cost $> C_{max}$); see Figure 2.

4.4 Face Recognition Experiments

We used 534 face images of 102 different people, each 92×112 pixels, from which we placed 187 images into T, another 187 images into S (T and S are used in the training phase to collect statistics) and used the remaining 260 as test images. As shown in Figure 5, the faces are more or less in the same pose (facing front), with some small variation in size and orientation.[3] We considered all 20 operators based on the four features listed above and $k \in \{25, 30, 35, 40, 45\}$ for each feature.

Basic Experiment: We set $P_{min} = 0.9$ and $C_{max} = \infty$ (*i.e.*, no upper limit on identification cost). In each "set-up", we assigned a random probability to each person. On each run, we picked one face randomly from the test set as the target, and identified it using each of the three policies. We repeated this for a total of 25 runs per set-up, then changed the probability distribution and repeated the entire process again, for a total of 8 set-ups. The cost of recognition on the average was 8.764 ± 0.586, 7.674 ± 0.702 and 6.811 ± 0.702

[3](1) All these faces were downloaded from the web sites `whitechapel.media.mit.edu` and `www.cam-orl.co.uk`. (2) This work assumes the head has already been located and normalized; if not, we can use standard techniques [TP91] first. (3) All the experiments reported in this paper were run on a Pentium 200 MHz. PC with 32 MB. RAM running Linux 2.0.35 OS.

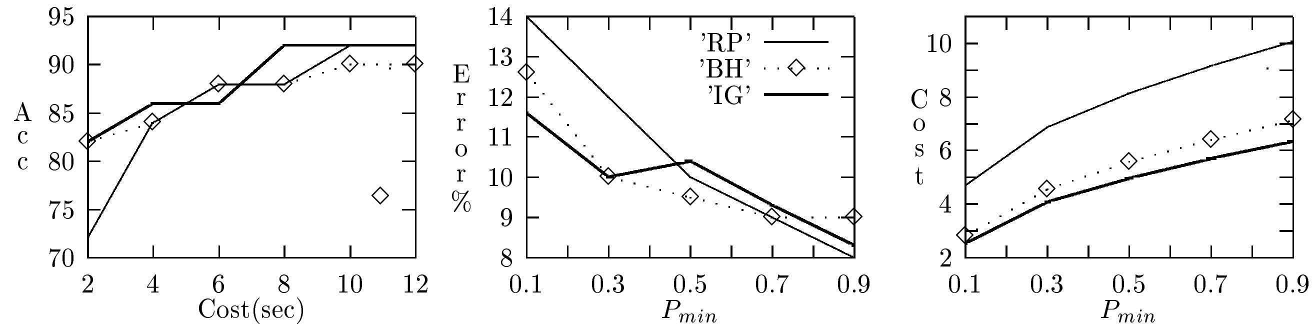

Figure 6: (a) Cost vs. Accuracy (b) Min. Accuracy vs. Error (c) Min. Accuracy vs. Cost

seconds for RANDPOL, BESTHYP and INFOGAIN respectively. INFOGAIN is statistically better than the other two policies, at the $p < 0.1$ level. (As expected, these policies had comparable identification accuracy here: 89.16%, 90.84% and 90.84%, respectively.)

Bounding the Cost: In many situations, we need to impose a hard restriction on the total cost; we therefore considered $C_{max} \in \{2, 4, \ldots, 12\}$ seconds. We then picked one face randomly from the test set, and identified the test image for each of these maximal costs, using each of the three policies. As always, we terminate whenever the probability of any person exceeds P_{min} or if the cost exceeds C_{max}, and return the most likely interpretation.

We repeated this experiment for a total of 10 set-ups (each with a different distribution over the people) and with 25 random runs (target face images) per set-up. The accuracy (the percentage of correct identifications) for each policy is shown for various values of C_{max} in Figure 6(a). INFOGAIN has better accuracy than both BESTHYP and RANDPOL. RANDPOL trailed these two policies significantly for low (≤ 4 seconds) cost.

Varying the Minimum Accuracy: In this experiment, we varied P_{min} from 0.1 to 0.9. For each of these values, we chose a face randomly from the test set as the target and identified it using each of the three policies. During the process, the first person i in the training set for which $P(\,\mathrm{id}(h) = i \,|\, \cdot\,) > P_{min}$ is returned (or if cost $> C_{max}$, the most probable face is returned). We repeated this for 25 different target faces (runs) per set-up, and repeated the entire process for a total of 8 different set-ups.

We evaluated the results in two different ways. First, Figure 6(b) compares the percentage of wrong identifications of each policy, for each P_{min} value. INFOGAIN has fewer wrong identifications than BESTHYP and RANDPOL for low accuracy. As expected, for sufficiently high accuracy, all three policies have comparable number of wrong identifications. Secondly, Figure 6(c) compares the average cost of each policy, for each P_{min} value. Again, INFOGAIN has lower cost than BESTHYP and RANDPOL, while RANDPOL trails the other two policies significantly.

5 Literature Survey

Our work formally investigates the use of decision theory in image interpretation, explicitly addressing accuracy versus efficiency tradeoffs [GE91; RSP93]. Geman and Jedynak [GJ96] used information theoretic approaches to find the "optimal sequence of questions" for recognizing objects — hand-written numerals (0-9), and highways from satellite images. Our work also uses information theoretic methods to compute the expected information gain, but we differ as we are seeking the most cost-effective sequence of interpretation operators, rather than the shortest sequence of questions; this forces us to explicitly address the cost vs accuracy tradeoffs. Sengupta and Boyer [SB93] presented a hierarchically structured approach to organizing large structural model-bases using an information theoretic criterion. We do not have explicit model-bases, but consider various interpretation policies that decide, at run time, which operators to apply to what regions of an image, based on expected information gain of the operators.

We used the domain of face recognition to test our approach. While there are several approaches to face recognition [TP91; PMS94; PWHR98; EC97], none *explicitly* address the issues of efficiency. We used one of the popular and successful methods as the basis to our approach and performed a systematic study of the various efficiency and accuracy related issues.

Levitt et al. [LAB90; BLM89; LHD$^+$93] have applied Bayesian inference methods and influence diagrams to image interpretation. We however provide a way to adjust the optimization function. (Our work also further motivates the use of "maximum expected utility" in such systems.)

As our system is seeking a policy that maps the state (here current distribution over possible interpretations) to an appropriate action, it can be viewed as solving a Markov decision problem (MDP), which puts it in the realm of reinforcement learning; *cf.*, [Dra96]. Our research objective differs as we are considering a range of reward functions, which can be various combinations of accuracy and efficiency (some of which may be difficult to word within a MDP framework). We anticipate being able to use many of the Reinforcement Learning techniques as we begin to consider interactions between the actions, and going beyond our current myopic approach.

Finally, there is a growing body of work on providing precise characteristics of various imaging operators, which quantify how they should work [Har94; RH92]. We hope to use these results to quantify the effectiveness of our operators, to help our algorithms decide when to use each. There is also work on building platforms that allow a user to *manually* assemble these operators [Fua97; PL94], often using an expert-system style approach [Mat89].

Here, we are taking a step towards automating this process, wrt some given task. In particular, our approach suggests a way to *automatically* assemble the appropriate imaging operators (*i.e.*, without human intervention), as required to effectively interpret a range of images.

6 Conclusions

Future Work: While our face recognition results show that our ideas can be applied to a complex domain, there are a number of extensions that would further scale up our approach. Some are relatively straightforward — *e.g.*, extending the set of operators to cover more features; this will help deal with larger number of faces in the training set, with better accuracy and lower interpretation costs. In other contexts, we will need to deal with thornier issues, such as operators that rely on one another. This may be because one operator requires, as input, the output of another operator (*e.g.*, a line-segmenter produces a set of tokens, which are then used by a line-grower — notice this precondition-situation leads to various planning issues [CFML98]), or because the actual data obtained from one operator may be critical in deciding which next operator (or parameter setting) to consider next: *e.g.*, finding the fuselage at some position helps determine where to look for the airplane's wings.

Clearly we will need to re-think our current myopic approach to cope with these multi-step issues; especially as we expect heuristics will be essential, as this task is clearly NP-hard [Sri95]. Finally, all of this assumes we have the required distribution information. An important challenge is more efficient ways to acquire such information from a set of training images — perhaps using something like Q-learning [SB98].

Contributions: This paper has three main contributions: First, it provides a formal foundation for investigating *efficient* image interpretation, by outlining the criteria to consider, and suggesting some approaches. Secondly, our implementation is a step towards *automating* the construction of effective image interpretation systems, as it will automatically decide on the appropriate policies for operator applications, as a function of the user's (explicitly provided) task and the available inventory of operators. Finally, it presents some results related to these approaches — in particular, our results confirm the obvious point that information gain (as embodied in the INFOGAIN policy) is clearly the appropriate measure to use here — and in particular, it is better than the BESTHYP approach. This observation is useful, as there are deployed imaging systems that use this BESTHYP approach [LHD+93].

Acknowledgements

RG gratefully acknowledges support from NSERC for this project. RI thanks Sastry Isukapalli for several discussions, general comments and help on the paper. Both authors thank Ramesh Visvanathan, and the anonymous reviewers, for their many helpful and insightful comments.

References

[BLM89] T Binford, T Levitt, and W Mann. Bayesian inference in model based machine vision. In *UAI*, 1989.

[CFML98] S Chien, F Fisher, H Mortensen, and E Lo. Using ai planning techniques to automatically reconfigure software modules. In *Lecture Notes in CS*, 1998.

[DH73] R. Duda and P. Hart. *Pattern Classification and Scene Analysis*. Wiley, New York, 1973.

[Dra96] B. Draper. Learning control strategies for object recognition. In Ikeuchi and Veloso, editors, *Symbolic Visual Learning*. Oxford University Press, 1996.

[EC97] K Etemad and R Chellappa. Discriminant analysis for recognition of human faces. *J. Optical Society of America*, 1997.

[Fua97] P Fua. *Image Understanding for Intelligence Imagery*. Morgan Kaufmann, 1997.

[GE91] R. Greiner and C. Elkan. Measuring and improving the effectiveness of representations. In *IJCAI91*, pages 518–524, August 1991.

[GJ96] G Geman and B Jedynak. An active testing model for tracking roads in satellite images. *IEEE PAMI*, 1996.

[Har94] R Haralick. Overview: Computer vision performance characterization. In *ARPA IU*, 1994.

[HR96] R Huang and S Russell. Object identification: A bayesian analysis with application to traffic surveillance. *Artificial Intelligence*, 1996.

[IG01] R. Isukapalli and R. Greiner. Efficient car recognition policies. In *ICRA*, 2001.

[LAB90] T S Levitt, J M Agosta, and T O Binford. Model-based influence diagrams for machine vision. In *UAI*, 1990.

[LHD+93] T. Levitt, M. Hedgecock, J. Dye, S. Johnston, V. Shadle, and D. Vosky. Bayesian inference for model-based segmentation of computed radiographs of the hand. *Artificial Intelligence in Medicine*, 1993.

[Mat89] T. Matsuyama. Expert systems for image processing: knowledge-based composition of image analysis processes. *Computer Vision, Graphics, and Image Processing*, 48(1):22–49, 1989.

[Mit97] T. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[PL94] A. Pope and D. Lowe. Vista: A software environment for computer vision research. In *IEEE CVPR*, 1994.

[PL95] A. Pope and D. Lowe. Learning object recognition models from images. In *Early Visual Learning*, 1995.

[PMS94] A Pentland, B Moghaddam, and T Starner. View-based and modular eigenspaces for face recognition. In *IEEE CVPR*, 1994.

[PWHR98] P Phillips, H Wechsler, J Huang, and P Rauss. The feret database and evaluation procedure for face recognition algorithms. *Image and Vision Computing*, 1998.

[RH92] V Ramesh and R M Haralick. Performance characterization of edge operators. In *Machine Vision and Robotics Conference*, 1992.

[RSP93] S. Russell, D. Subramanian, and R. Parr. Provably bounded optimal agents. In *IJCAI93*, August 1993.

[SB93] K Sengupta and K Boyer. Information theoretic clustering of large structural modelbases. In *IEEE CVPR*, 1993.

[SB98] R. Sutton and A. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[Sri95] S Srinivas. A polynomial algorithm for computing the optimal repair strategy in a system with independent component failures. In *UAI*, 1995.

[TP91] M Turk and A Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 1991.