Least Committment Graph Matching by Evolutionary Optimisation

Richard Myers and Edwin R. Hancock

Department of Computer Science
University of York
York YO1 5DD, UK.

Abstract. This paper presents a method of matching ambiguous feature sets extracted from images. The method is based on Wilson and Hancock's Bayesian matching framework [1], which is extended to handle the case where the feature measurements are ambiguous. A multimodal evolutionary optimisation framework is proposed, which is capable of simultaneously producing several good alternative solutions. Unlike other multimodal genetic algorithms, the one reported here requires no extra parameters: solution yields are maximised by removing bias in the selection step, while optimisation performance is maintained by a local search step. An experimental study demonstrates the effectiveness of the new approach on synthetic and real data. The framework is in principle applicable to any multimodal optimisation problem where local search performs well.

1 Introduction

Graph matching problems have pervaded computer vision since the early 1970s, when Barrow and Popplestone used graphs in [2] to represent spatial relationships between scene components. Minsky's "frames" extended this idea using a hierarchical representation with image features at the bottom level and scene knowledge [3]. These ideas have given rise to representations such as aspect graphs and part-primitive hierarchies [4,5] have found favour. Comparing relational representations is a central task in many areas of computer vision, among which are stereo matching [6], image registration [1], object recognition [5] and image labelling [7]. The main problem is that the two graphs are often different, so that it is not possible to locate an isomorphism. The inexact graph matching problem is to locate the best match between the graphs. Early attempts at inexact matching used heuristics to reconcile dissimilar structures [8,9]. These heuristics have been augmented by information theoretic and probabilistic criteria [10,5,11].

Over the last ten years, there has been more interest in using statistical evidence from the scene, instead of purely structural matching [12,13]. Wilson and Hancock have recently developed a Bayesian formulation which naturally combines the structural constraints imposed by the graphs with evidence from the scene [1]. This formulation provides a distance measure which can be optimised with gradient ascent, relaxation or genetic algorithms [14,15,16]. However, the

D. Vernon (Ed.): ECCV 2000, LNCS 1842, pp. 203–219, 2000.

classification technique used by Wilson and Hancock makes the assumption that there is a single best match. This approach works best where each of the features is distinctly located in the feature space, when it is possible to make a single minimum risk assignment. However, when the features are poor in the sense that there is considerable overlap in the feature space, restricting attention to the most probable assignments incurs a substantial risk of ignoring alternatives which are only slightly less good. For example, in a situation where there are two possible assignments with probabilities close to 0.5, it would be unwise to ignore the less likely one. This paper will demonstrate how to overcome this difficulty for graph matching in the context of Wilson and Hancock's Bayesian framework.

The standard method of enumerating a set of good solutions to an optimisation problem is to restart the optimiser, possibly with modifications to avoid revisiting optima [17]. However, this implies discarding valuable information uncovered in the optimisation process. An alternative is to use population based optimisation techniques, of which the genetic algorithm [18,19] is the most well-known.

The idea that genetic algorithms can be used to simultaneously find more than one solution to a problem was first mooted by Goldberg and Richardson in [20]. They attempted to prevent the formation of large clusters of identical individuals in the population by de-rating the fitness function. Other techniques include crowding [21], sequential niching [17], and distributed genetic algorithms [22]. A common feature of these approaches has been the necessity for extra parameters. Niching and crowding strategies typically require two or three extra parameters to be controlled. These parameters are needed, for example, to determine when to de-rate the fitness of an individual, by how much, and the distance scale of the de-rating function. In distributed algorithms, it is necessary to decide how to arrange the sub-populations, their sizes, and under what conditions migration between them may occur. In [23], Smith and co-workers demonstrated a situation in which niching could occur in a standard genetic algorithm, without the need for any extra parameters. The authors have demonstrated elsewhere that unmodified genetic algorithms are capable of finding many solutions to line labelling problems [24]. This paper will obtain similar behaviour for graph matching, and show how suitable algorithm modifications can improve solution yield without introducing any new parameters.

This paper will show that using a genetic algorithm as an optimisation framework for graph matching could allow a vision system to follow Marr's principle of least commitment [25]. The outline of this paper is as follows. The next section reviews and extends Wilson and Hancock's Bayesian graph matching criterion. Section 3 considers the implementation of a genetic algorithm for ambiguous graph matching problems. An experimental study is presented in section 4. Finally, section 5 draws some conclusions and suggests directions for future research.

2 Bayesian Matching Criterion

The problems considered in this paper involve matching attributed relational graphs [26]. An attributed relational graph is a triple, $G = (\mathbf{V}, \mathbf{E}, \mathbf{A})$, where $\mathbf{V}$ is the set of vertices or nodes, $\mathbf{E} \subseteq \mathbf{V} \times \mathbf{V}$ is the set of edges, and $\mathbf{A} \subset \mathbf{V} \times \Re^k$ is the set of measurement $k-$vectors relating to the original scene. Graph matching is the problem of establishing a correspondence between a data graph, $G_D = (\mathbf{V}_D, \mathbf{E}_D, \mathbf{A}_D)$, and a model graph, $G_M = (\mathbf{V}_M, \mathbf{E}_M, \mathbf{A}_M)$. This correspondence, $f : V_D \mapsto V_M \cup \{\phi\}$, is a labelling of the nodes in $\mathbf{V}_D$ with nodes from $\mathbf{V}_M$ or a special null label, ϕ, for unmatchable nodes.

In [1], Wilson and Hancock described a framework in which both neighbourhood structure and node attributes were combined in a single measure of matching consistency. The goal is to optimise the *a posteriori* probability of the match given the measurements:

$$P(f|\mathbf{A}_D, \mathbf{A}_M) = \frac{p(\mathbf{A}_D, \mathbf{A}_M|f)}{p(\mathbf{A}_D, \mathbf{A}_M)} P(f) \tag{1}$$

where $P(f)$ is the structural component of the criterion. The joint measurement density, $p(\mathbf{A}_D, \mathbf{A}_M)$, only depends on the measurements and is thus a static property of the problem which can be ignored when comparing matches. The conditional measurement density, $p(\mathbf{A}_D, \mathbf{A}_M|f)$, depends on both the current match and the measurements. Wilson and Hancock showed that, assuming conditional independence of these measurements given the current match, the conditional measurement density can be factorised over the tuples in f to give

$$p(\mathbf{A}_D, \mathbf{A}_M|f) = \prod_{(u,v)\in f} P(u, v|x_u, x_v) \frac{p(x_u, x_v)}{P(u, v)} \tag{2}$$

where the posterior matching probability, $P(u, v|x_u, x_v)$, is the probability of node u from the data graph matching node v in the model graph given their measurements, x_u and x_v. Like $p(\mathbf{A}_D, \mathbf{A}_M)$, the unconditional density, $p(x_u, x_v)$, is independent of the current match, f. Assuming that the matching priors, $P(u, v)$, are uniformly distributed, equation 1 can be written

$$P(f|\mathbf{A}_D, \mathbf{A}_M) \propto \left[\prod_{(u,v)\in f} P(u, v|x_u, x_v) \right] P(f) \tag{3}$$

Wilson and Hancock used this relationship to formulate the MAP update rule given in [1] for iteratively improving the match, f. The mapping of data graph node u, was chosen from the union of the model graph node set, $\mathbf{V}_m$, with the null label, $\{\phi\}$, according to:

$$f(u) = \arg \max_{v\in \mathbf{V}_m \cup \{\phi\}} P(u, v|x_u, x_v) P(f) \tag{4}$$

The structural component of the criterion, $P(f)$, although an essential ingredient, is not a primary concern in this paper. To summarise, the structural

constraints on matching can be explicitly captured by defining a dictionary of legal assignments, Θ_j, over the neighbourhood of each node, j, in the data graph. Wilson and Hancock then applied Hancock and Kittler's Bayesian formulation of dictionary based relaxation [27], in which a memoryless error process is assumed to have produced the current assignments of the j^{th} neighbourhood, Γ_j, by corrupting the dictionary item, S_i. This error model leads to a structural consistency criterion which is exponential in the distance between assignments and dictionary items, D.

$$P(f) = \frac{1}{|\mathbf{V}_D|} \sum_{j \in \mathbf{V}_D} \frac{K_{C_j}}{|\Theta_j|} \sum_{S_i \in \Theta_j} \exp\left[-k_e D(\Gamma_j, S_i)\right] \tag{5}$$

where $K_{C_j} = (1 - P_e)^{|\Gamma_j|}$ and $k_e = \ln\left(\frac{1-P_e}{P_e}\right)$. The probability of assignment corruption, P_e, can be used as a control variable in a manner analogous to the temperature in simulated annealing [1]. In [28], Myers, Wilson and Hancock have shown that the Levenshtein distance was the most appropriate choice for D [29].

Wilson and Hancock also used a feasibility heuristic to screen out highly unlikely mappings. If the difference in the neighbourhood sizes of a data graph node, u, and a model graph node, v, is above some threshold, the mapping, $f(u) = v$, is considered infeasible. This heuristic has been very successful in reducing the size of the search space.

2.1 Measurement Ambiguity

The measurement information contributes to the matching criterion via the posterior matching probability, $P(u, v|x_u, x_v)$, which has yet to be defined. In [1], Wilson and Hancock defined it in terms of the Euclidean distance between attribute pairs for non-null mappings:

$$P(u, v|x_u, x_v) = \begin{cases} P_\phi & \text{if } v = \phi \\ (1 - P_\phi) \dfrac{\exp\left[\frac{-(x_u - x_v)^2}{2\hat{\sigma}_v^2}\right]}{\sum_{w \in V_M} \exp\left[\frac{-(x_u - x_w)^2}{2\hat{\sigma}_w^2}\right]} & \text{otherwise} \end{cases} \tag{6}$$

where P_ϕ is the prior probability of a null match, $f(u) = \phi$, which may be taken as $2\left\|\frac{|\mathbf{V}_D| - |\mathbf{V}_M|}{|\mathbf{V}_D| + |\mathbf{V}_M|}\right\|$, and $\hat{\sigma}_v^2$ is the estimated variance of x_v. This effectively regards the model graph node measurement, x_v, as a mean about which the data graph node measurement, x_u, varies with estimated variance $\hat{\sigma}_v^2$, under the null hypothesis that the two measurements are the same (because the nodes match). This approach requires the assumption that a data measurement is only likely to be statistically close to **one** of the model measurements. This is ideal when there is little overlap between classes, e.g. for possible angles of line-fragments segmented from a radar image. However, if there is significant overlap, e.g. in the average intensities of regions, such a scheme will not reflect these ambiguities in its classification of features.

The alternative is to compare the data measurements to the model measurements using an artificial scale. This can be done by considering the number of standard deviations separating the data measurement from its class mean under the null hypothesis that the nodes match. Table 1 gives an example of such a scale for the arbitrary classes "similar", "comparable", and "different".

Table 1. Example Scale for Measurement Comparisons.

Class	Range of standard deviations from x_v
Similar	[0,1.0]
Comparable	(1.0,2.0]
Different	(2.0,∞]

Consider the standardised distance, $\Delta_{uv} = \|x_u - x_v\|/\hat{\sigma}_v$. The probability that x_u lies within $[a, b]$ standard deviations on either side of x_v is twice the standard Normal integral from a to b:

$$P(a \leq \Delta_{uv} \leq b) = \sqrt{\frac{2}{\pi}} \int_a^b e^{-\frac{1}{2}z^2} dz \qquad (7)$$
$$= \text{erf}\left(\frac{b}{\sqrt{2}}\right) - \text{erf}\left(\frac{a}{\sqrt{2}}\right)$$

Each of the classes in table 1 corresponds to a separate interval which must be considered. Rather than introduce so many extra parameters, it is better to simplify the classification to "similar" if $\Delta_{uv} \in [0, a]$ and "dissimilar" otherwise. Thus, $P(u, v|x_u, x_v)$ can be defined as follows

$$P(u,v|x_u,x_v) = \begin{cases} P_\phi & \text{if } v = \phi \\ (1-P_\phi)\dfrac{\exp\left[\frac{-(x_u-x_v)^2}{2\hat{\sigma}_v^2}\right]}{\sum_{w\in V_M}\exp\left[\frac{-(x_u-x_w)^2}{2\hat{\sigma}_w^2}\right]} & \text{if } a = 0 \\ (1-P_\phi)P[\Delta_{uv} \leq a] & \text{if } \Delta_{uv} \leq a \\ (1-P_\phi)(1-P[\Delta_{uv} \leq a]) & \text{otherwise} \end{cases} \qquad (8)$$

For convenience, the original unambiguous definition is used when $a = 0$. At the cost of an extra parameter, a, ambiguous measurements can now be handled. The important property of equation 8 is that when $a > 0$, it assigns the exact same probability to sets of mappings, thus enabling different alternatives to be considered.

The ambiguity parameter, a, has a direct interpretation as the number of standard units beyond which data measurements are considered dissimilar from the model. However, it has a more useful indirect interpretation in terms of "matching tolerance". The matching tolerance, T, is defined as the average proportion

of model measurements similar to each data measurement, and is a function of a for any particular graph pair as shown in equation 9. The feasibility of mappings, feasible(u, v), is determined according to the neighbourhood size difference constraint mentioned at the end of section 2.

$$T(a) = \frac{1}{|\mathbf{V}_D||\mathbf{V}_M|}|\{(u, v) \in \mathbf{V}_D \times \mathbf{V}_M | \text{feasible}(u, v) \wedge \Delta_{uv} \leq a\}| \qquad (9)$$

Figure 1 shows T as a function of a for several synthetic graphs. The tolerance reaches a plateau of between 0.4 and 0.7 at values of a higher than about 2. This plateau is the limit imposed by the feasibility constraint, feasible(u, v). These graphs suggest that it should be possible to determine an appropriate value of a by estimating the proportion of "similar" features in the data set. For example, if this proportion is estimated to be 10%, a should be about 0.5.

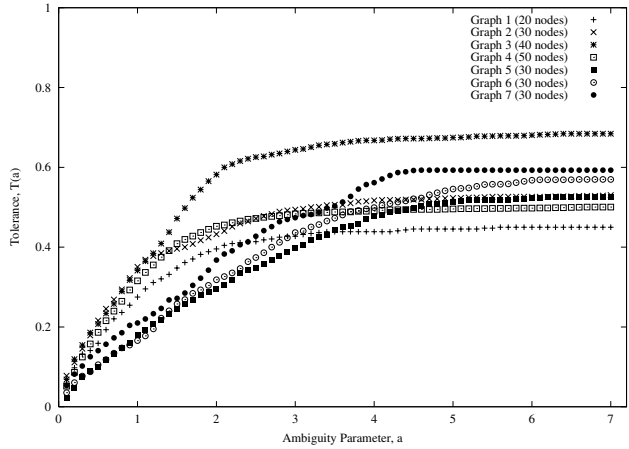


Fig. 1. Matching Tolerance as a Function of the Ambiguity Parameter.

3 Genetic Algorithms

The feasibility of using a genetic algorithm as an optimisation framework for graph matching in general was established by Cross, Wilson and Hancock in [16]. That work showed that the algorithm outperformed other optimisation techniques such as gradient ascent and simulated annealing. This section complements the previous work, adapting it with a view to enhancing solution yield. The algorithm used in this study is a variant of the simple genetic algorithm. Rather than restrict mating to the fittest individuals, it is allowed to occur at random. This models more closely the panmitic populations observed in nature.

After reproduction, both parents and offspring are subject to selection to produce the parent population for the next generation. This process is designed to better exploit the diversity in the population. The algorithm is augmented with a gradient ascent step: optimisation of the MAP criterion in equation 4. Such a step would seem ideal for labelling problems since gradient ascent is a standard technique for solving them in an optimisation framework [30,31,27,1]. The tendency of gradient ascent to get stuck in local optima should be mitigated by the global optimisation properties of the algorithm.

As in [16], equation 3 can be used directly as the fitness function. The constant of proportionality depends on the joint measurement density, $p(\mathbf{A}_D, \mathbf{A}_M)$, the unconditional density, $p(x_u, x_v)$, and the matching priors, $P(u, v)$, which are assumed to have a uniform distribution. Since selection is based on the ratio of the fitness of a particular individual to the the total fitness of the entire population, this constant need not be explicitly calculated.

A detailed consideration of crossover and population size cannot be given here due to lack of space. Suffice to say that uniform crossover [32] was found to perform best with the hybrid algorithm. Appropriate population sizes can be determined by considering how good an initial guess has to be before the gradient ascent step can successfully repair it. It turns out that the MAP update rule of equation 4 can locate optimal solutions from an initial guess in which only 10% of the mappings are correct. For a 50-node problem and an initial population size of 10, the probability that this will be the case is 0.98. So even for moderately large graphs, relatively small populations should be adequate.

3.1 Selection

In a standard genetic algorithm, selection is crucial to the algorithm's search performance. Whereas mutation, crossover and local search are all "next-state" operators, selection imposes a stochastic acceptance criterion. The standard "roulette" selection algorithm, described by Goldberg in [19], assigns each individual a probability of selection, p_i, proportional to its fitness, $\mathcal{F}_i$. The genetic algorithm used here allows the population, $\mathbf{\Psi}$, to grow transiently and then selects the next generation from this expanded population. Denoting the expanded population by $\mathbf{\Psi}_e$, the selection probability of the i^{th} individual, p_i, is given by

$$p_i = \frac{\mathcal{F}_i}{\sum_{j \in \mathbf{\Psi}_e} \mathcal{F}_j} \tag{10}$$

The algorithm then holds selection trials for each "slot" in the new population, for a total of $|\mathbf{\Psi}|$ trials. Since selection is with replacement, the constitution of the new population is governed by the multinomial distribution, and the copy number of a particular individual, $N(i)$, is distributed binomially:

$$P(N(i) = r) = \begin{pmatrix} |\mathbf{\Psi}| \\ r \end{pmatrix} p_i^r (1 - p_i)^{|\mathbf{\Psi}| - r} \tag{11}$$

and so the expectation of $N(i)$, is $\mathrm{E}[N(i)] = |\mathbf{\Psi}| p_i$, and its variance is $\mathrm{Var}[N(i)] = |\mathbf{\Psi}| p_i (1 - p_i)$.

The search power of the standard genetic algorithm arises from the fact that if the individual in question is highly fit, p_i will be much larger than the average, and hence the expectation will be that the copy number will increase. This approach has two disadvantages. The first is that for small populations, sampling errors may lead to copy numbers very much higher or lower than the expected values. This can lead to premature convergence of the algorithm to a local optimum. In [33], Baker proposed "stochastic remainder sampling", which guarantees that the copy number will not be much different from the expectation by stipulating that $\lfloor \mathrm{E}[N(i)] \rfloor \leq N(i) \leq \lceil \mathrm{E}[N(i)] \rceil$. However, the larger the population, the less need there is for Baker's algorithm [34]. The second disadvantage is that less fit individuals have lower expectations, and that the lower the fitness, the lower the variance of the copy number. In other words, less fit individuals are increasingly likely to have lower copy numbers. When $E[N(i)]$ falls below 1, the individual will probably disappear from the population. In general, the copy number variance decreases with decreasing fitness. Only when $p_i > 0.5$ does the variance decrease with increasing fitness. This occurs when the fitness of one individual accounts for at least half the total fitness of the population, i.e. when it is at least $|\mathbf{\Psi}_e| - 1$ times as fit as any other individual.

In short, the problem with roulette selection is that it imposes too strict an acceptance criterion on individuals with below average fitness. Several alternative strategies have been proposed to avoid this problem. "Sigma truncation", rank selection and tournament selection [19,35] all seek to maintain constant selection pressure by requiring individuals not to compete on the basis of their fitness, but on some indirect figure of merit such as the rank of their fitness, or the distance between their fitness and the average in standard units. Taking rank selection as a typical example of these strategies, the selection probabilities are assigned by substituting the rank of the individual for its fitness in equation 10, with the best individual having the highest rank. The implication of this is that the expected copy numbers of the best and worst individuals are given by:

$$\left.\begin{aligned} \mathrm{E}[N(\mathrm{best})] &= \frac{2|\mathbf{\Psi}|}{(|\mathbf{\Psi}_e|-1)} \\ \mathrm{E}[N(\mathrm{worst})] &= \frac{2|\mathbf{\Psi}|}{|\mathbf{\Psi}_e|(|\mathbf{\Psi}_e|-1)} \end{aligned}\right\} \tag{12}$$

So, the expected copy number of the fittest individual differs from that of the least fit by a factor of $|\mathbf{\Psi}_e|$. Moreover, if $|\mathbf{\Psi}_e|$ is even moderately large, $\mathrm{E}[N(\mathrm{worst})]$ will be much less than 1. Indeed, $\mathrm{E}[N(i)]$ will be less than 1 for about half the population. Thus, under rank selection, less fit individuals are highly likely to disappear, even if they are quite good.

A second alternative to roulette selection is Boltzmann selection [36,37]. This strategy borrows the idea from simulated annealing, that at thermal equilibrium the probability of a system being in a particular state depends on the temperature and the system's energy. The idea is that as the temperature is lowered, high energy (low fitness) states are less likely. The difficulty with this analogy is that it requires the system to have reached thermal equilibrium. In simulated annealing, this is achieved after very many updates at a particular temperature. However, in a genetic algorithm this would require many iterations at each

temperature level to achieve equilibrium, coupled with a slow "cooling". Within the 10 or so iterations allowed for hybrid genetic algorithms, equilibrium cannot even be attained, let alone annealing occur.

It would appear, then, that there is a tradeoff between premature convergence and the strength of the selection operator. The problem arises from the fact that expected copy numbers of fit individuals may be greater than one, while those of unfit individuals may be less than one. One way of preventing the increase in copy number of highly fit individuals is to use "truncation selection", as used in Rechenberg and Schwefel's evolution strategies [38,39]. Truncation selection would simply take the best $|\mathbf{\Psi}|$ individuals from the expanded population, $\mathbf{\Psi}_e$, to form the new population. The copy number of each individual is simply 1 or 0, depending on its rank. Although no individual may increase its copy number, the selection pressure might still be quite severe, since for the algorithm used in this paper, $|\mathbf{\Psi}_e|$ can be as large as $3|\mathbf{\Psi}|$. In other words, less fit individuals still disappear at an early stage. The fact that individuals never increase their copy number makes this a relatively weak search operator, and probably unsuitable for a standard genetic algorithm. However, the gradient ascent step is itself a powerful optimiser [1], and may be mostly responsible for the optimisation performance of the algorithm. If this is so, selection would be a much less important search operator for this hybrid algorithm than it is for standard genetic algorithms. It may therefore be beneficial to trade search performance for greater diversity.

Neutral Selection The benefits of stochastic selection can be combined with the evenness of truncation selection by selecting without replacement. This strategy can be called "biased selection without replacement", since it is biased first in favour of fitter individuals, although it may also favour less fit ones.

The alternative is to abandon fitness based selection altogether, and rely on the local search step to do all the optimisation. If the genetic algorithm's rôle is explicitly limited to assembling a good initial guess for the local search operator, the selection probabilities can be assigned uniformly, i.e. $\forall_{i \in \mathbf{\Psi}_e} p_i = \frac{1}{|\mathbf{\Psi}|}$. This operator is called "neutral selection". Neutral selection without replacement can be implemented very efficiently by shuffling $\mathbf{\Psi}_e$ and choosing the "top" $|\mathbf{\Psi}|$ individuals. This strategy shares the advantage with truncation selection, that the minimum number of individuals are excluded from the new population, but also maintains the global stochastic acceptance properties of standard selection operators.

Elitism Elitist selection guarantees that at least one copy of the best individual so far found is selected for the new population. This heuristic is very widely used in genetic algorithms. In [40], Rudolph showed that the algorithm's eventual convergence cannot be guaranteed without it. The elitist heuristic can be modified in two ways to help maintain diversity. First, it seems natural that if the goal is to simultaneously obtain several solutions to the problem in hand, several of the fittest individuals should be guaranteed in this way. This is called "multiple

elitism". Second, if one wishes to avoid losing too many unfit individuals, the *worst* individual can also be granted free passage to the new population. This is called "anti-elitism". These heuristics, together with the selection strategies discussed earlier, are evaluated at the end of section 4.

4 Experiments

This experimental study establishes the suitability of the hybrid genetic algorithm for ambiguous graph matching, and compares the selection strategies discussed in the prevous section. The algorithm was tested on 30-node synthetic graphs The point sets were generated at random, and then triangulated by connecting each point to six of its nearest neighbours. Data graphs were generated by randomly perturbing the node attributes, and then duplicating 10% of the nodes and perturbing their attributes. The intention was to simulate segmentation errors expected of region extraction, such as the splitting of one region into two similar ones.

4.1 Comparative Study

A comparative study was performed to determine the best algorithm for ambiguous matching. The algorithms used were the hybrid genetic algorithm with and without mutation, crossover or both (hGA, hGA-m, hGA-x and hGA-xm),[1] a hybrid version of Eshelman's CHC algorithm [41] (hCHC), and plain gradient ascent (HC). The experimental conditions are summarised in table 2.

Table 2. Algorithms for Graph Matching. Each algorithm, apart from HC, made approximately 700,000 fitness evaluations. Abbreviations: hGA = hybrid genetic algorithm, hGA-m = hGA without mutation, hGA-x = hGA without crossover, hGA-xm = hGA with neither mutation nor crossover, hCHC = hybrid CHC, and HC = gradient ascent (hillclimbing).

	hGA	hGA-m	hGA-x	hGA-xm	hCHC	HC
Population	50	50	120	120	100	1
Iterations	5	5	5	5	5	10
Crossover	Uniform	Uniform	Uniform	Uniform	HUX	n/a
Cross rate	0.9	0.9	0.0	0.0	1.0	n/a
Mutate rate	0.3	0.0	0.3	0.0	0.35	n/a

[1] These should be regarded as different algorithms, not merely different parameter sets for a genetic algorithm, because a genetic algorithm with no crossover or mutation is fundamentally different from one which has these operators. For example, the hGA-xm algorithm is really just multiple restarts of gradient ascent with a selection step.

Each of the algorithms listed in table 2, except HC, was run 100 times. Since HC is deterministic, it was only run once per graph. The results for the different graphs were pooled to give 400 observations per algorithm. Algorithm performance was assessed according to two criteria. The first was the average fraction of correct mappings in the final population. The second was the proportion of distinct individuals in the final population with more than 95% correct mappings. The results are reported in table 3.

Table 3. Graph Matching Results. Standard errors are given in parentheses. Abbreviations: hGA = hybrid genetic algorithm, hGA-m = hGA without mutation, hGA-x = hGA without crossover, hGA-xm = hGA with neither mutation nor crossover, hCHC = hybrid CHC, and HC = gradient ascent (hillclimbing).

Algorithm	Average Fraction Correct	Average Fraction Distinct
hGA	0.90 (0.0044)	0.078 (0.0019)
hGA-m	0.88 (0.0051)	0.040 (0.0012)
hGA-x	0.84 (0.0052)	0.044 (0.00094)
hGA-xm	0.76 (0.0068)	0.013 (0.00036)
hCHC	0.92 (0.0042)	0.012 (0.00033)
HC	0.97 (n/a)	n/a

At first sight, pure gradient ascent appears to outperform all the other algorithms. The reason for this is partly that the gradient ascent algorithm starts from an initial guess in which about 50% of the mappings are correct, whereas the other algorithms start with random initial guesses. More importantly, the final population of a genetic algorithm typically contains solutions much better and worse than the average. Thus, this comparison is not really fair: a fairer comparison of optimisation performance comes from considering hGA-xm, which is multiple random restarts of gradient ascent. Furthermore, gradient ascent is deterministic, and therefore always gives the same result, but the genetic algorithm is stochastic and may do significantly better or worse than gradient ascent. Indeed, the genetic algorithm occasionally found matches with 100% correct mappings. However, the performance of gradient ascent alone suggests that for unambiguous problems, genetic algorithms may not necessarily be the method of choice. Apart from pure gradient ascent, the best optimiser was hCHC, which is only slightly better than hGA. The results for hGA-m and hGA-x indicate that crossover and mutation are playing an active part in the optimisation process. Turning to the fraction of distinct individuals with over 95% correct mappings, it is clear that pure gradient ascent is incapable of finding more than one solution. The hCHC algorithm appears to converge to fewer solutions than the hGA algorithm. In all, the hybrid genetic algorithm (hGA) combines strong optimisation performance with the highest solution yield, and it is this algorithm which will be the subject of the remainder of this study.

4.2 Selection

Two sets of experiments were conducted to evaluate different selection strategies with and without elitism. In each case, a hybrid genetic algorithm was used, with a population size of 20, and uniform crossover was used at a rate of 1.0. The mutation rate was fixed at 0.4. The first set of experiments used 20, 30, 40 and 50 node graphs, and for these the population size was set to 10, and the algorithm run for 5 iterations. The second set of experiments used four 30 node graphs, with a population size of 20 and 10 iterations. Five different selection strategies were compared: they were standard roulette, rank, and truncation selection, and neutral and biased selection without replacement. Five combinations of elitist heuristics were considered: they were no elitism, single elitism, multiple elitism, anti-elitism, and a combination of multiple and anti-elitism. The experimental design was therefore a 5x5x4 factorial with 100 cells. The first set of experiments had 40 replications for a total of 4000 observations; and the second set had 50 replications for 5000 observations. Figures 2 and 3 summarise the results.

Both plots show that neutral selection without replacement produced the best yields, and that truncation selection produced the worst. Biased and roulette selection strategies gave similar results, and were both outperformed by rank selection. Linear logistic regression analysis of both data sets confirmed this ranking of selection strategies. The results for elitism heuristic were not so convincing. It is questionable whether elitism has any overall effect: the regression analysis of the second data set found no significant effect of varying the elitism strategy. The analysis of the first data set did show that either standard (single) or multiple elitism gave significantly better yields, but that the effect was small.

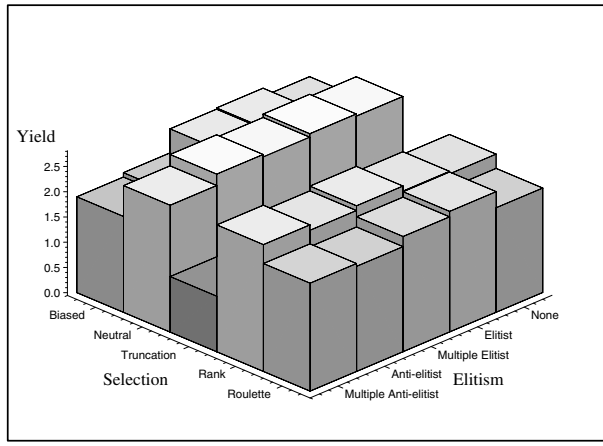


Fig. 2. Average Yields versus Selection and Elitism I. Data from all four graphs has been pooled.

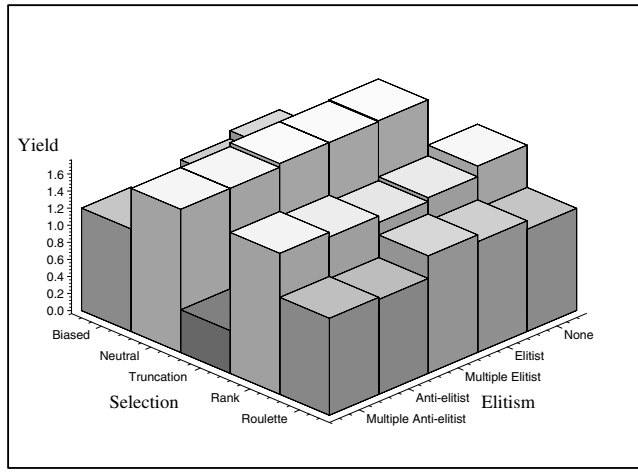
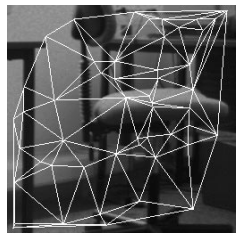


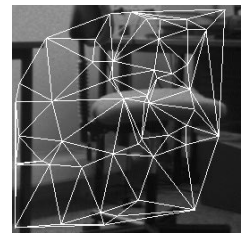
Fig. 3. Average Yields versus Selection and Elitism II. Data from all four graphs has been pooled.

4.3 Real Images

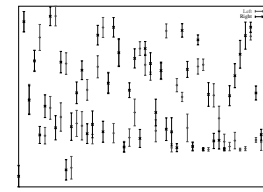
The problem motivating this paper is the registration of point sets derived from image features. Unlike Wilson and Hancock's point sets [1], the problems considered here do not generally produce unambiguous measurements. Take for example the case of the stereogram in figure 4. Regions were extracted from the greyscale image pair (panels (a) and (b), an office scene taken with an IndyCam) using a simple thresholding technique. Each image contained 50 regions. The region centroids were Delaunay triangulated using Triangle [42]. The average grey level over each region was used for the attribute information, shown in panel (c). The overlap between left and right image measurements indicates that the images are matchable. However, the overlap between left and left, and right and right, image attributes suggests that unambiguous assignments will not be possible. The Delaunay triangulations were matched using a hybrid genetic algorithm with neutral selection. The population size was set to 5, and 5 iterations were allowed. The crossover and mutation rates were 1.0 and 0.5 respectively. Panel (d) shows an initial guess in which none of the mappings is correct. Panels (e) to (g) show the three distinct solutions found. There were 50 regions in the left image of which 42 had feasible correspondences in the right. The amount of relational corruption between the two triangulations was estimated at around 35% by counting the number of inconsistent supercliques given the ground truth match. Despite the significant relational corruption, the three solutions had 98%, 93% and 95% correct mappings.



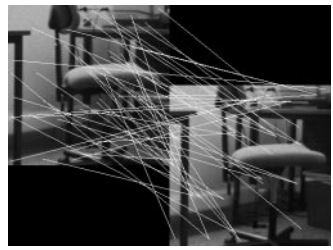
(a) Left Image



(b) Right Image

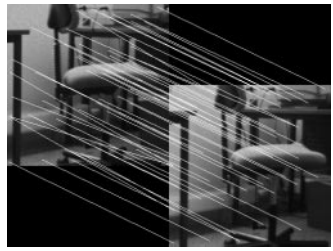


(c) Average Region Intensities



(d) Initial Guess (0%)

(e) Final Match (98%)



(f) Final Match (93%)

(g) Final Match (95%)

Fig. 4. Uncalibrated Stereogram 1. The camera positions are not known.

5 Conclusion

This paper has presented a method of matching ambiguous feature sets with a hybrid genetic algorithm, which does not require any additional parameters to achieve multimodal optimisation. This allows the principle of least commitment [25] to be applied to such problems. The first contribution made was to adapt the Bayesian matching framework, due to Wilson and Hancock [1], to handle

ambiguous feature measurements. Rather than finding the most probable assignment, the new framework appeals directly to the underlying measurement distributions to classify data features as similar or dissimilar to model features, and to assign probabilities to those classes.

If most of the optimisation is undertaken in the gradient ascent step, the tradeoff between effective search and maintenance of diversity, which must be made in choosing a selection operator for standard genetic algorithms, can be abandoned. Neutral selection without replacement maximises the diversity in the next generation with no regard to individuals' fitnesses. This operator was shown in section 4.2 to provide the highest solution yields.

There are a number of interesting directions in which this work could be taken. First, the ambiguity parameter or matching tolerance, defined in section 2.1, must at present be estimated graphically by considering what proportion of features are statistically similar. It should be possible to infer suitable values of this parameter more robustly, given a set of training examples. A final observation is that in figure 4, the variations in the solutions tend to concern assignments which are incorrect. This raises the possibility of directing a vision system's focus of attention to those parts of a scene which are most problematic.

References

1. R. C. Wilson and E. R. Hancock. Structural matching by discrete relaxation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:634–648, 1997.
2. H. G. Barrow and R. J. Popplestone. Relational descriptions in picture processing. In B. Meltzer and D. Michie, editors, *Machine Intelligence*, volume 6. Edinburgh University Press, 1971.
3. M. Minsky. A framework for representing knowledge. In P. H. Winston, editor, *The Psychology of Computer Vision*. McGraw-Hill, 1975.
4. J. Koenderink and A. van Doorn. The internal representation of solid shape with respect to vision. *Biological Cybernetics*, 32:211–216, 1979.
5. S. J. Dickinson, A. P. Pentland, and A. Rosenfeld. 3-D shape recovery using distributed aspect matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:174–198, 1992.
6. R. Horaud and T. Skordas. Stereo correspondence through feature grouping and maximal cliques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:1168–1180, 1989.
7. A. C. M. Dumay, R. J. van der Geest, J. J. Gerbrands, E. Jansen, and J. H. C. Reiber. Consistent inexact graph matching applied to labeling coronary segments in arteriograms. In *Proceedings of the* 11$^{\text{th}}$ *International Conference on Pattern Recognition*, volume C, pages 439–442, 1992.
8. A. Sanfeliu and K. S. Fu. A distance measure between attributed relational graphs for pattern recognition. *IEEE Transactions on Systems, Man and Cybernetics*, 13:353–362, 1983.
9. L. G. Shaprio and R. M. Haralick. A metric for comparing relational descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7:90–94, 1985.
10. A. K. C. Wong and M. You. Entropy and distance of random graphs with application to structural pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7:599–609, 1985.

11. J. Liang, H. I. Christensen, and F. V. Jensen. Qualitative recognition using Bayesian reasoning. In E. S. Gelsema and L. S. Kanal, editors, *Pattern Recognition in Practice IV*, pages 255–266, 1994.
12. K. W. Boyer and A. C. Kak. Structural stereopsis for 3-D vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10:144–166, 1988.
13. J. Kittler, W. J. Christmas, and M. Petrou. Probabilistic relaxation for matching problems in computer vision. In *Proceedings of the* 4th *IEEE International Conference on Computer Vision*, pages 666–673, 1993.
14. R. C. Wilson and E. R. Hancock. A Bayesian compatibility model for graph matching. *Pattern Recognition Letters*, 17:263–276, 1996.
15. R. C. Wilson and E. R. Hancock. Graph matching by discrete relaxation. In E. S. Gelsema and L. N. Kanal, editors, *Pattern Recognition in Practice*, volume 4, pages 165–176. Elsevier, 1994.
16. A. D. J. Cross, R. C. Wilson, and E. R. Hancock. Inexact graph matching using genetic search. *Pattern Recognition*, 30:953–970, 1997.
17. D. Beasley, D. R. Bull, and R. R. Martin. A sequential niche technique for multimodal function optimisation. *Evolutionary Computation*, 1:101–125, 1993.
18. J. H. Holland. *Adaptation in Natural and Artificial Systems.* MIT Press, 1975.
19. D. Goldberg. *Genetic Algorithms in Search, Optimisation and Machine Learning.* Addison-Wesley, 1989.
20. D. E. Goldberg and J. Richardson. Genetic algorithms with sharing for multimodal function optimization. In *Proceedings of the* 2^{nd} *International Conference on Genetic Algorithms*, pages 41–49, 1987.
21. W. Cedeño, V. R. Vemuri, and T. Slezak. Multiniche crowding in genetic algorithms and its application to the assembly of DNA restriction-fragments. *Evolutionary Computation*, 2:321–345, 1995.
22. M. Gorges-Schleuter. ASPARAGOS: A parallel genetic algorithm for population genetics. In *Parallelism, Learning, Evolution. Workshop on Evolutionary Models and Strategies*, pages 407–418, 1991.
23. R. E. Smith, S. Forrest, and A. S. Perelson. Searching for diverse, cooperative populations with genetic algorithms. *Evolutionary Computation*, 1:127–149, 1993.
24. R. Myers and E. R. Hancock. Genetic algorithms for ambiguous labelling problems. *Lecture Notes in Computer Science (EMMCVPR 97)*, 1223:345–360, 1997. Extended version to appear in Pattern Recognition.
25. D. Marr. *Vision.* Freeman, 1982.
26. K. S. Fu. A step towards unification of syntactic and statistical pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5:200–205, 1983.
27. E. R. Hancock and J. Kittler. Discrete relaxation. *Pattern Recognition*, 23:711–733, 1990.
28. R. Myers, R. C. Wilson, and E. R. Hancock. Efficient relational matching with local edit distance. In *Proceedings of the* 14^{th} *International Conference on Pattern Recognition*, pages 1711–1714, 1998.
29. V. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics – Doklady*, 10:707–710, 1966.
30. R. A. Hummel and S. W. Zucker. On the foundations of relaxation labeling processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5:267–287, 1983.
31. O. D. Faugeras and M. Berthod. Improving consistency and reducing ambiguity in stochastic labeling: An optimisation approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3:412–424, 1981.

32. G. Syswerda. Uniform crossover in genetic algorithms. In *Proceedings of the* 3rd *International Conference on Genetic Algorithms*, pages 2–9, 1989.
33. J. E. Baker. Reducing bias and inefficiency in the selection algorithm. In *Proceedings of the* 2nd *International Conference on Genetic Algorithms*, pages 14–21, 1987.
34. M. Mitchell. *An Introduction to Genetic Algorithms.* MIT Press, 1996.
35. J. E. Baker. Adaptive selection methods for genetic algorithms. In J. J. Grefenstette, editor, *Proceedings of the* 1st *International Conference on Genetic Algorithms*, 1985.
36. D. E. Goldberg. A note on Boltzmann tournament selection for genetic algorithms and population-based simulated annealing. *Complex Systems*, 4:445–460, 1990.
37. A. Prügel-Bennett and J. L. Shapiro. An analysis of genetic algorithms using statistical physics. *Physical Review Letters*, 72:1305–1309, 1994.
38. I. Rechenberg. *Evolutionsstrategie - Optimierung Technischer Systeme nach Prinzipien der biologischen Information.* Fromman Verlag, 1973.
39. H-P. Schwefel. *Numerical Optimization of Computer Models.* Wiley, 1981.
40. G. Rudolph. Convergence analysis of canonical genetic algorithms. *IEEE Transactions on Neural Networks*, 5:96–101, 1994.
41. L. J. Eshelman. The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. In G. J. E. Rawlins, editor, *Foundations of Genetic Algorithms*, volume 1, pages 265–283. Morgan Kaufmann, 1991.
42. J. R. Shewchuk. Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator. In *Proceedings of the* 1st *Workshop on Applied Computational Geometry*, pages 124–133, 1996.