Random Search Methods Speed Up Object Recognition

Harvey A Cohen
School of Computer Science and Computer Engineering
La Trobe University Bundoora 3083Australia
Alan L Harvey
Electrical Engineering Department
RMIT Melbourne 3000 Australia

Abstract
Random search methods are long established mathematical techniques for multivariate optimization, and have been applied to such optimization tasks as the training of neural nets, but are not an established method in image analysis. In this paper we present a new approach to the problem of object location within an image, with mismatch function determined via a template, by regarding this task, as an optimization problem and applying a random search method based on the Matyas algorithm. This elaboration of the Matyas algorithm, called random cluster search, involves a simple pattern search about random image locations. The paper reports experiments in which random cluster search was compared with both conventional exhaustive search and sparse data based fast search methods. Speed-ups as much as 40% compared to the most efficient deterministic search strategies were recorded in situations where a single sought object was located in. the image.

1. Introduction

The recognition and location of an object within an image has long been a central task of computer vision. In machine vision object recognition is vital to robotic assembly, and is an essential element of automated surveillance. With the recent emergence of massive image databases requiring automated means of classification, the recognition of objects within an image will play a significant role. Thus improvements to classical methods of object recognition and location remain of interest.

The key idea of this paper is to treat the problem of object recognition using templates as an optimization task using a well-established method of numerical analysis, random search that has not been previously applied to this problem. The optimization task is the location of the minima of the template image match function evaluated at all datapoint locations within an image, that is, at all pixel locations.

In *random search* algorithms for multivariate optimization, some, not necessarily all, co-ordinates of search locations are randomly generated. Random or stochastic search methods are an established method of numerical optimisation, [2][3][7] and have been used for a wide class of minimisation problems. These optimization methods of numerical analysis have been found to have significant advantages when the objective function is difficult to compute or a global minimum is required in the presence of many local minima. [8] Random search methods have been used in the training of neural nets, and were shown by Baba to be much faster than the gradient descent. Benke [22] and Cohen and You [23] have used random search in the "tuning" of 5x5 convolution masks to meet objective functions. However random search has not previously been applied to object location.

. Random search algorithms were written in order to investigate firstly the applicability of these methods and secondly their effectiveness in terms of computational cost. Important search variables were the number of short range calculated searches relative to the number of random searches and the range of the random step in the calculated steps. After a series of investigations it was found that for a given template size, a strategy defined below termed a guided cluster search of appropriate scale converged after less iterations than a systematic sparse grid search. A 100x100 Pebbles image from Brodatz was used for the investigation to give spatially constant image properties, particularly texture or the auto correlation function

In this object recognition study, objects were located within an image by finding the point of minimal object template mismatch after a series of studies carefully established the correct range for the local search referred to here as a *cluster search*. The random search strategy used consisted of an objective function evaluation at a randomly generated image location followed by several function evaluations at a short distance from the point of best match so far. In this application the point of best match means the image point where the mismatch function is lowest so far. The ratio of random searches to local searches and the method of selection of the local search incremental steps are of fundamental importance to finding the global minimum for template mismatch in a minimal number of search steps. The efficiency of

various strategies was compared in terms of the average number of searches to find the object.

1. Random Search Review

In the search for a data item at an unknown location, there is a basic distinction between deterministic and random search methods. In a deterministic search, indicative signs of near match may be used to modify the pattern of search. For example, in most coarse-fine deterministic search strategies, indications of near-match are use to switch from the coarse search pattern to the fine one. In random search, although the basic pattern of search is random, here, too, the parameters of search may be modified by indications of near-match. In the work described here a regular pattern of search is performed about a randomly determined location.

This section provides an overview of random search**,** following closely the formulation of Solis and Wets, then presents a new random search algorithm that is an extension of the Matyas random search algorithm. This modified algorithm, called the cluster algorithm, includes Matyas as a special case.

It is important to note that in any search algorithm, random or deterministic, the location of the next location to be examined is a function of the location of the current search location, the results obtained there, together with some the results of immediate search algorithm. Hence a search algorithm is specified well in terms of the step, and its direction, from the current location to the next position to be examined.

Before turning to what is called in this paper and elsewhere random search, it is useful to consider n an exhaustive search, all entirely random search, what could be called a "pure"random search, random search locations are generated and the objective function is evaluated. No data is retained regarding image locations for previous searches. For a purely random search in a NxN image, the required number of searches will not change with the object position. A truly random search can be expected to take N^2 searches on average. In practise, a purely random search would not be used. In practical random search methods, search positions need to be biased towards areas of low mismatch in order to find the optimum position more quickly.

1.1 The Matyas Algorithm

The classic random search algorithm for multivariate optimization is that of Matyas [8]. In the Matyas algorithm, a Gaussian random step from the previous best position is taken and the matching function evaluated. If the match is better at the new coordinates, this position is adopted. Otherwise the next random step is taken from the old x,y position.

In mathematical terms we have the following, where we formulate for the case of multivariate minimization of a function f of the vectorial quantity x:

An initial value of the coordinate vector x^0 is chosen and f(x) evaluated for search number k = 0.

For search k>0

Generate a Gaussian Random vector ξ

Check that $x^k+\xi \in X$ .is within the search space X..

If so, calculate $f(x^k+\xi)$.

If $f(x^k+\xi) < f(x^k$, then $x^{k+1} = x^k+\xi$

Continue until f(x) is less than a threshold value or k is equal to the maximum search number M.

Baba and Solis and Wets [1]-[11] have given proofs of convergence for random search methods. They stipulate various conditions for the search method. These conditions are easily satisfied in practice. The basic condition these proofs use is that the best so far is selected. Formally, if x^k is the best so far at iteration k, and ξ^k is a new estimate in iteration k, then the value to be selected is given by the function $L(x^k, \xi^k)$, satisfying:

$L(x^k, \xi^k) = \xi^k$, if $f(x^k) > f(\xi^k)$

$= x^k$ otherwise

Note that x^k and ξ^k are vectors; in this case of object location in conventional 2D images they are 2D vectors.

These proofs, subject to credible conditions, state that there is zero probability of repeatedly missing the point of minimum provided enough searches are made. Of greater practical significance is how many searches need to be made to be reasonably certain of obtaining the global minimum? This can be estimated in the case of a 2D function from the ratio of the area of convergence to the area of the search space. This point is discussed more fully in the following sections.

Solis and Wets [11] improved the convergence of the Matyas random search by adding a second search step at the "mirror image" of the random search location ξ at x(k)-ξ. If the search step at x(k)+ξ does not reduce f(x) then compare with x(k)-ξ and accept this position if f(x) is less at that position. Solis and Wets also calculate a bias vector b around which the random vector is generated. $b^{(k+1)} = 0.4.\xi + 0.2b^{(k)}$

If the reflected step gives a better result then $b^{(k+1)} = 0.4.\xi - 0.2b^{(k)}$

1.2 Simulated Annealing

In the Matyas search algorithm detailed above, the search proceeds via the "best-so_far" location invariably. In contrast, there are a range of stochastic search algorithms

based on thermodynamic analogies where the search proceeds according to an inherently more complex algorithm. The earliest such papers by Metropolis et al [9] introduced what is now termed "simulated annealing". [10][13]

The basis of simulated algorithm is as follows:
At search point x^k a random step ξ is generated.
The random step is accepted if $f(\xi) < f(x^k)$.
However if $f(\xi) > f(x^k)$ still accept ξ with probability:
$p = \exp\{-(f(\xi)- f(x^{(k)})/T_k\}$. Thus as the value of f(x) decreases, the probability of acceptance of the random step also decreases. The temperature T_k is reduced during the search causing the acceptance probability to decrease also.
Recently Bilbro et al [23] showed that simulated annealing was serviceable in the global optimization of functions with many minima.

1.3 Guided Random Search

In the study reported here a new version of random search is introduced, called guided random search, that generalises the Matyas Algorithm. The key idea for this modification is to select a new search location at random, evaluate the mismatch function at this new location, then search at computed locations about the best so far location.

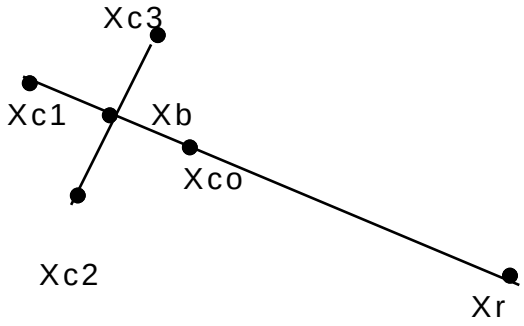

Figure 2 Guided Random Search
Locations of search locations about the best-so-far.

In the new algorithm the computed search locations are clustered about the best-so-far location. Two points (pixels) on the cluster are on the line between the best-so-far and the previous best-so-far, and both at the same distance from the best-so-far, exactly as for Solis and Wets version of Matyas algorithm; but there are a further two search locations at the same distance from the best-so-far, on a line perpendicular to the same line. See Fig 2, above. To simplify programming, the Gaussian search probability used by Matyas has been replaced by simple uniform distribution (see below).

2 - Search Strategies

The purpose of this section is to outline the search methods which are used in an experimental study comparing XXX with what can be considered state of the art conventional approaches to object location via the use of templates. The algorithm discussed is as follows:

1. Sequential Search
2. Sparse/Fine Sequential Search
3. Purely Random Search
4. Convergent Random Search

These are detailed in the following sub-sections.

2.1 Sequential Search

For a NxN region, the area is searched point by point giving a maximum number of N^2 search points. For a sequential or exhaustive search, the number of searches to find the object will vary with the object position. Taking the average position as the centre, the average number of searches will be $N^2/2$. For an object template of size TxT pixels, there are actually $(N-T+1)^2$ search positions for the full image but for small templates, N^2 points is a close approximation.

2.2 Sparse/Fine Sequential Search

The sparse/fine search methodology consists of searches at a sparse set of image locations in a raster pattern switching to a fine search at all locations if the mismatch starts to converge. In other words, a purely sequential search can be speeded up by using a sparse set of points providing the mismatch error is monitored well enough not to miss the correct match. For a NxN image, with an 8 pixel step, the number of sparse grid searches will be $N^2/8$ for a line by line sequential search. This does not include the number of fine steps taken when the mismatch error indicates that the error is reducing significantly and fine steps of one pixel must be taken. Studies show that the overall average step will be equal to the coarse or sparse step less one, a sparse step of 8 for example producing an average step of seven when the fine steps are included.

2.3 Purely Random Search

In an entirely random search, random search locations are generated and the objective function is evaluated. No data is retained regarding image locations for previous searches. For a purely random search in a NxN image, the required number of searches will not change with the object position. A truly random search can be expected to take N^2 searches on average. In practise, a purely random search would not be used. In practical random search methods, search positions need to be biased towards areas of low mismatch in order to find the optimum position more quickly.

2.4 Convergent Random Search

In a convergent random search, random steps are followed by local searches around the best result so far. For a local mismatch area of A pixels in a NxN region, the expected number of random searches required is N.N/A from a first principles approach before the search

will start to converge. The total number of searches will then be the above plus the number of local searches, to converge on the object. Search types vary from random increment local searches with fixed relative locations to random direction searches with a fixed radius from the last best position. The size of the step must be comparable to the size of the mismatch region, which depending on template size and configuration, is of the order of 5-15 pixels in width.

For the 2D image matching problem, the problem formulation is as follows: [11] Given a function f of dimension R^2 , and a search region S which is a subset of R^2, we seek a point x in S which minimises f(x), or at least gives a close approximation to the minimum for the mismatch function of f on S.
The Algorithm given therein is as follows:

Step 0. Find x(0) in S and set k = 0. (K is the search counter.)

Step 1. Generate E_k from sample space (Rn, B, uk).

Step 2. Set x(k+1) = D(x(k),E_k), choose u(k+1), set k=k+1. Goto 1.

uk is a probability variable related to biasing the search to regions of low error.

The map D with domain S x Rn and range S satisfies the following condition:

$f(D(x,E)) \leq f(x)$ and if $E \in S$, $f(D(x,E)) \leq f(E)$.

uk is a probability measure corresponding to distance function defined on R^2.
By Mk we denote support of uk ie Mk is the smallest subset of R^2 of measure 1.

To converge, random search methods must be adaptive, ie uk depends on certain quantities, in particular the value of the objective function at x0,x1,...x(k-1) generated by preceding iterations. uk is regarded as a conditional probability measure.

2.4.1 Random Search Termination

Some information on the matching function is required to determine a reasonable figure for the total number of searches in order to terminate the search (a) if the object is present or (b) if the object is not present. If the size of the mismatch region is known, an estimate of the most probable number of searches may be made and the maximum search number can be set to say four or five times this figure for a failure criterion. Alternatively an acceptable threshold for the mismatch function may be used as a termination criterion for success.

2.4.2 Final Search Speed Up

When the search is within the convergence region of the correct match point, a large number of methods may be used to find the best point of match as quickly as possible. Some methods require function evaluations only, others require gradient calculations to indicate the direction of maximum gradient descent. For searches with a large search space a very small amount of time will be spent on finding the minimum as distinct from finding the region where the point of match is. Thus the overall gain in reducing search time by using elaborate "hill climbing" or descent methods for object recognition is unlikely to be very significant. A method that can quickly dismiss an area as unproductive will be of greater benefit.

RANDOM SEARCH STRATEGIES.
For a search area of considerable size where it is reasonable to look for ways to avoid an exhaustive search of all locations, a random search will eventually succeed and as long as there is a finite correlation area the search can be terminated relatively early assuming the method converges reasonably once the mismatch dip is found. This is the same criterion as the sparse search using an adaptive step size requires for it's success in reducing x,y calculation points.[6] If the number of calculation points is still large, meaning the maximum coarse step is small but the area is large, a random search may reduce the number of x,y calculation points but the burden of finding the point of minimum error from local error measure calculations must be low to make the method competitive.

A *cross search,* is one in which searches are carried out in a cross shaped pattern and then the search is relocated at the location of the best result. Then value of w, the step size is then halved and another cross search performed.[5] Such a search, even if aborted after one cross search due to lack of convergence requires five calculations at locations, at ± w in the x and y directions and one at the original random location. A cross search must be made at each randomly generated x,y location as a minimum is equally likely in any particular location. Thus a random search using a local cross search must be at least 5 times as efficient as an ordered sparse step x,y raster scan search to give comparable performance. Other local search strategies, such as the simplex method, require only three points, so that potentially it can give a reduced search overhead over the cross search method.

Figure1 Mismatch Function Plot, Inverted.

SPARSE TEMPLATE CALCULATION

The mismatch calculation at each location may be reduced by using a sparse grid of template points by taking, for example every second x,y template sample. This reduces the mismatch calculations by a factor of four at each image location. Other reduction methods may be used, for example taking every second complete row or column. Alternatively, one line and one row in the form of a cross may be used. For greater precision, a switch to a full template may be used when warranted, for example if the mismatch drops to a very low value. For very noisy data, full templates must be used to give greater discrimination..

PURELY RANDOM SEARCH RESULTS

As the limiting case of a random search, a set of searches was made at purely random row and column co-ordinates. The row and column random co-ordinates were 0-99 for a 100x100 pixel textured image with the template taken at location 50,50. The average number of searches was 8,501 with a standard deviation of 9001 for a 12/5 pixel sparse template at an average time of 3.24 seconds averaged over 10 data sets.

The method is comparable with an exhaustive search at 10,000 locations. Search numbers varied wildly from 2,121 to 25,604 due clearly to the fact that no attempt was made to obtain convergence. Note that 25,604 searches is about five times the number of search positions required by a sequential, exhaustive search terminated at position 50,50. By keeping the template small, the time per search is not large but the number of searches required is excessive. Note that there is nothing to stop search coordinates being duplicated. Also without any local searches being made, search results close to a match will be ignored.

Random Versus Calculated Step Search Strategies

The following discussion is for flat distribution random search steps rather than the Gaussian search steps used by Matyas and Solis and Wets. In random search techniques, searches are made at coordinates that are purely random, ie have no connection whatever with previous search results or other steps which are made close to the previous best match in order to find the minimum close to the point of best match found so far. The number of calculated local searches per random search to find the point of best match is of considerable interest. The following results show the effect of varying the number of calculated steps from 1 per random search to 4 calculated steps per random search step.

See the diagram below for search location positions.

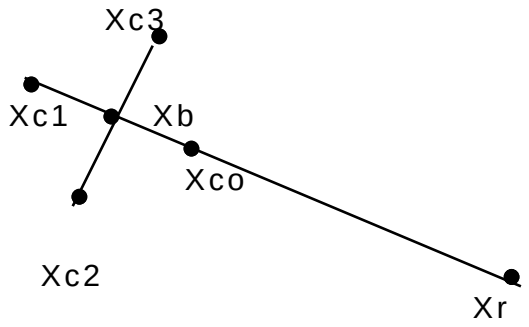

Figure 2 Random Search Locations.

In the diagram, Xb is the point of best match so far and Xr is a purely random image location. A computed step is then taken which is position Xco on the Xb-Xr line. Further function evaluations can be made at the reflection of Xco at Xc1 and at right angles to the Xb-Xr line giving image locations Xc2 and Xc3. Note that the calculated step size varies with Xr, so the "radius" of the cluster search around Xb varies.

$$Xc = Xb + (Xr-Xb).\beta/100$$

In the above equation, the calculated step Xc is equal to the best so far location plus a fraction β/100 of the distance between the latest random location and the best so far location. The following figures are for a 100x100 section of the 256x256 pixel image pebbles. Template was located at image row, column of 30, 30. A 12 x 12 pixel template with a sparse grid of 5x5 pixels (12/5 template) gave the following results:

One calculated step Xco Best of Xb, Xco and Xr retained.

No of Successes	Avg Search	STD	Time secs.
17	5764	6099	2.63
34	2780	2631	1.20
20	4537	3592	2.06

Two calculated steps Xco,Xc1 Best of Xb, Xc's & Xr is retained.

No of Successes	Avg Search No	Std Dev'n	Time secs.
83	1202	1060	0.58
68	1446	1195	0.71
80	1244	1146	0.56

Three calculated steps Xco, Xc1,Xc2 Best of Xb, Xc's & Xr retained.

No of successes	Avg Search No	Standard Dev'n	Time secs.
93	1064	1015	0.68
75	1311	1156	0.58

Four calculated steps Xc0,Xc1,Xc2,Xc3 Best of Xb, Xc's & Xr retained.

No of successes	Average Search No	Standard Deviation	Time secs.
89	1114	1000	0.52
86	1127	944	0.47
96	1015	948	0.45

It is clear from these results that the two or more calculated step searches are superior to the one step search and that three calculated steps did, on one run, give a slightly better result than the two calculated step search.

Random Search Calculated Step Size Considerations

The size of the step from the location of the best so far match position depends largely on the characteristic scale of the mismatch function. If the function slowly moves to a minimum, the scale and hence step size will be large compared with a function that rapidly drops to a minimum over a few pixels. An investigation of the step size for a 12/5 sparse template for *two* calculated steps is as follows: Image pebbles Row, column location of 30.

Data used to evaluate calculated step size for fastest guided random search convergence

Step	No finds	Avg search	Std Dev'n	Time
3/100	23	4314	4199	1.68s
5/100	80	1244	1146	0.56s
6/100	114	876	779	0.46s
7/100	113	881	759	0.39s
8/100	90	1098	794	0.48s
10/100	51	1958	2063	0.80s
20/100	22	4376	3579	1.68s
50/100	10	9841	12447	3.58s

Very clearly step fractions between 5/100 and 8/100 give good results with 7/100 giving the best results with this 12/5 sparse template for the two calculated step search strategy. At a fraction of 50/100 the result is essentially the random search result, namely NxN or 10,000 locations in this case for a 100x100 image search space.

The evaluation of the optimum value of the computed step ratio, ie the ratio of the distance Xb-Xc to the distance Xb-Xr shown above was done for computed steps Xc and Xc1, the reflected step. In other words how close should the short range steps be to the point of best match so far to give maximum speed of convergence? The previous table shows the results of this study.

Random Search Template Size Effect

As the template size of the image object increases, the mismatch dip size increases and therefore the probability of finding the object also increases. However for a full template, the calculation load will increase as the square of the template size so a small number of template pixels is desirable since the template mismatch burden is large since it is calculated at every search location. By using a sparse template, in which we take a sub sample of pixels from the template, studies showed that a larger mismatch region for a given number of pixels is available. This result is very important as it applies to all search methods and gives a very significant reduction in search times.
The following results are for the pebbles image with two calculated steps, a search multiplier alpha of 7/100 and a correct template position at row and column 30.

Results Sparse Template Size

#of Loc'ns	Avg. Search	Std Dev'n	Time	Template
14	6028	4145	2.64	3x3/1
99	1006	829	0.47	12x12/5

Since alpha is reduced for the smaller template, a fairer comparison is with an appropriate value of alpha. For alpha reduced to 4, and a 3x3 dense template, results were as follows

18	5211	4459	2.16	3x3/1
24	3791	3843	1.54	3x3/1

Clearly, even though the success rate increased with alpha reduced appropriately, while the number of successes has increased, there is still a dramatic reduction from 1.54 seconds to 0.47 seconds due to the sparse grid, namely the pixels distributed on a 5x5 grid rather than a 1x1 spacing, 3x3 template. Note that there is the same number of pixels in each template, 3x3=9 pixels and integer 12/5 ie 0,5 and 10 giving a 3x3, 9 pixel sparse template also.
Other templates using vertical and/or horizontal lines of pixels are also possible.

1.3 Fixed Step Cluster Search

As an alternative to the cluster search step size varying with Xr and Xb, the step size K may be fixed and the direction of the cluster step only varies with Xr

$$Xc = Xb + K$$

where all symbols are as discussed and K is the step size. The vector K is varied in direction in the same way as the varying step cluster search. This methodology also converges quickly if K is chosen well.

For 3 calculated steps: Image Pebbles

K	# Searches
2	3,753
3	986
4	1755

3.4 Concentric Square Search

An ordered square search strategy was developed by searching in a square pattern and then increasing the size of the square until the image area is covered. The search

starts at the centre of the image and ends when a square equal to the image size is searched. The square search may be speeded up by essentially the same methods as the ordered "raster scan" search where the search starts proceeds row by row from the top left corner and finishes at the bottom left corner. These speed up methods are a sparse/fine search of the sides of the squares and a system where squares are skipped over if the previous search square shows no sign of a match. Sparse templates were used to reduce matching calculations at each search point.

In the square search variation, the mismatch function is evaluated in a consecutive manner at points on the edge of a square starting in the centre. In the sparse/fine square search, the mismatch function is evaluated at sparse locations spaced at 4 to 10 pixels depending on the template size. If the mismatch sum starts to drop, a fine search at steps of one pixel is made. The square search has the advantage of having the lowest average distance from any point since it starts from the centre.

Results for a concentric square search for the pebbles image are shown in the following tables

RESULTS CONCENTRIC SQUARE SEARCH
Forward Search Square spacing = 1, or all squares.

Template	Step	No Searches	Time sec.	col/row
12/5	1	9609	2.26	1
	2	4901	1.18	1
	3	3361	0.88	1
	4	2497	0.66	1
24/5	1	9609	4.40	1
	2	4901	2.45	1
	3	3367	1.67	1
	4	2513	1.33	1

Above results are for contiguous squares, ie no sparse searches on squares
By keeping a check on the minimum sum for each square, and comparing it with the previous square's minimum, an increment above the minimum may be made without missing the point of match. For the 12/5 template, this was not possible but with the larger template 24/5, the following results were obtained: :

Sparse Template	Step	# Searches	Time sec.	Col/row	Square Incr't
24/5	1	6270	2.83s	1	4
pixels	2	3360	1.78	1	"
	3	2370	1.32	1	"
	4	1591	0.93	1	"
	5	1377	0.83	1	"
	6	1104	0.72	1	"
	7	Fail			

It is apparent that the square search is not inherently faster than the raster scan search but it has the advantage of a lower average distance to reach a random position on the image.

A graphical interface was used to check the shape of the concentric square search and to check the effect of coding changes.

3.5 Raster -ordered Search Comparison

Ordered searches were made with the following variations:
1. Forward Sparse/Fine Column Steps.
2. Forward and Reverse Sparse/Fine Column Steps. rowstep = 1.
3. Above Column options plus sparse/fine row steps.
Forward Sparse/Fine Column Steps.
A forward search using sparse column steps step sizes in the range four to eight were made. The maximum successful step depends on template size and sparseness. Results are as follows for search times and numbers of searches for the image pebbles:

Forward Search Rowstep = 1 Program blokrow.c

Template	Step	No Searches	Time sec.	col/row
12/5	4	2056	0.44s	49
	5	1742	0.38	"
	6	Fail		"

Template	Step	No Searches	Time sec.	col/row
24/5	4	1995	0.94	49
	5	1658	0.80	49
Rowstep=1				
	6	1431	0.69	49
	7	1251	0.61	49
	8	1140	0.52	49

Search averages start point varied as above to stop accidental matches.
2. Forward and Reverse Sparse/Fine column steps. row step = 1 bloksize.c

Template	Step	# Searches	Time sec.	col/row
12/5	4	5655	1.51	49
	5	4764	1.26	49
	6	4122	1.12	49
	7	Fail		

3. Forward and Reverse Sparse/Fine Column and Row Steps
For the small 12/5 template, with row steps greater than one, the search failed to locate the template.
All Results are averages of two searches.

Template	4	2299	1.57	49	4
24/5	5	1926	1.29	49	4
	6	1616	1.07	49	4
	7	1336	0.88	49	4
	8	Fail			

3.5 Binary Template Random Search

Random search strategies were also evaluated using binary templates and images instead of grey level data. Binary data is important as many object recognition systems, especially character recognition systems, binarise the data to remove brightness variations. Random search strategies were successful with binary templates but it was found that 3x3 contiguous templates or smaller (non-sparse), could not be used as a match was found at other than the correct position of the template. This is to be expected for binary templates since the binarisation process throws away all but the most basic shape information. Larger templates or sparse templates were more successful although a single pixel error from the correct location occasionally occurred. Also searches did not converge as quickly as grey scale templates. Searches required two to three times more searches to converge on the correct location due to the smaller correlation region. The search mismatch dip is less than half the width in most cases. For the same template and image, the following results were found:

Image peb256, Template 31/5 [ie 30x30 with a pixel spacing of 5x5, Sparse Pixels 49, 100x100 space].

Binary, Grayscale Template Search Time Comparison for Various Match Positions.

Data	X,Y Position	Time	# Searches
Bin		30	0.73s 858
Grey		"	0.35 412
Bin		50	0.41 479
Grey		"	0.19 223
Bin		70	1.24 1443
Grey		"	0.40 456

3.6 Binary Images

A random search followed by two local searches was the strategy used in this investigation as it worked well for the grey scale data. The calculated step size was varied in order to speed up the convergence.
The following results were obtained. Note that the step is actually the percentage step of the random search minus the calculated search.

Step	Avg Search
6	443
9	402
11	371
12	332
13	343
15	425

Clearly there is a minimum at around 12 pixels although the minimum is not very sharp. Summarising the work on binary templates, random search methods can be used on binary templates using a single random and two calculated step strategy. For this textured image the calculated step was optimised at 12/100 for a 30x30 template with a sparse grid spacing of 5x5 pixels. These results show the comparison between grey scale and binary images and are indicative of results for the general case of binary images.

4 Discussion

This section is intended to provide a more technical and detailed overview of the experimental study, wiith general conclusions to be separately presented in the conclusion. The experimental results detailed above are summarised in Table II.

A first point to be discussed is how general are the test images. For a mathematically complete evaluation of random search methods compared with sequential search methods, it is clear that the object of interest should be located at all possible positions and comparisons made of both search methods. A summation could then be made to give a "global" figure of merit for each method. For any finite sized search area, the amount of computation required is huge and also hugely unnecessary as results for random searches would give essentially the same result and it is also axiomatic that a systematic search will give a result proportional to the distance of the sought object along the trajectory of search. Therefore the comparison was made with the object in the centre of the search space for comparison purposes as a fair and reasonable case for the purpose of comparing search methods for a given template size.

In the previous paragraph, it was stated that the random search, different (unknown) object location “would essentially give the same results”. This is clearly the case except when the object is located close to an image boundary, as then two effects come into play
(a) The distribution used for computation of random step may admit search locations outside the image, so that deleting such possibilities results in skewed search distrubutions.
(b) The sought object could be located totally within the image, but so close to image boundaries that the mismatch dip area overlaps the boundary, so that

essentially this mismatch region is reduced in size. Such a (almost on edge) location will also reduce the efficiency of coarse to fine search strategies,.

Table 2. Search Method Result Summary Search Space 100x100 Template 12/5 Pebbles Object at 50,50 for sequential search and 30,30 for random search.

Search Method	# Searches	SDV	Time sec
Sequential Search	5,051	0	1.70
Sparse/Fine Seq'l. 5/1	1,742	0	0.38
Simple Random Search	8,540	9001	3.24
1 Cluster Random Search	3989	4129	1.78
2 Cluster Random Search	1288	4129	0.61
3 " " " "	1174	1088	0.64
4 " " " "	1083	964	0.48
Fixed Step Cluster search			
3 Cluster Steps "	986	895	0.48

5 Conclusion

Object recognition and location via the use of templates is akin to a numerical optimization problem, where the function minima being sought are the bottom of rather narrow dips, of the order of 9 (pixel) units for the target/images examind here. In this paper we examined the applicability of random search algorithm, called the cluster algorithm, based on the Matyas algorithm. In the Matyas algorithm, a random step is taken from the current "best so far" to determine a new location. In the (original) Matyas algorithm, a second position is then searched at a determinstic location along the line joining these two locations. The (original) Matyas algorithm corresponds to our cluster algorithm, with only two (of four possible) points in the cluster. Essential our results indicate that cluster search based algorithm with just 1, or 2 (original Matyas) cluster points offer little advantage, but that the full cluster or four yields notable speed ups for object location in greyscale images,, compared to the quite extensive range of determinstic search strategies.

It is important to note that random search inherently has a random element, so that the computational cost has a high standard deviation, which has been computed for the experimental data tabled. Because of this high standard deviation, the number of locations that need to be searched to be confident that no object, or no further object(s) will be located, is significantly increased compared to an average search time for successful recognition/location.

What this paper essentially shows is that over a range of images, random search methods can be applied with advantage. For the images of this study, we found (near) optimal values for the parameters of random search. In searching for objects located within the images of a large and varied database, or a stream of real-time images from a mobile camera, the parameters of the random search would need to be adjusted.

References

[1] Baba, N. "A New Approach to Finding the Global Minimum of Error Function of Neural Networks" Neural Networks Vol 2 pp 367-373. 1989

[2] Brooks, S.H. "A Discussion of Random Methods for Seeking Minima" Oper. Res. Vol 6 pp 244-251, 1958

[3] Brooks,S.H. "A Comparison of Maximum Seeking Methods", Oper. Res. Vol. 7 pp 430-457, 1959

[4] Cohen, H.A. and Harvey,A.L., "Stochastic Search Approach to Object Location", Proc. IEEE Int'l Conf Systems, Man and Cybernetics, San Antonio, Texas, Vol 3, pp 2237-2241, 1994.

[5] Ghanbari, M. "The Cross-Search Algorithm for Motion Estimation" IEEE Transaction on Communications Vol 38, No.7 pp 950-953 July 1990

[6] Harvey, A.L. and Cohen. H.A. "Speed Up Methods for Image Object Recognition" ICARCV-90 Singapore pp 965-968, 1990.

[14] Harvey, A.L. "Sparse Data Methods in Image Engineering" Ph.D. Thesis, School of Computer Science and Computer Engineering, La Trobe University, August, 1994.

[7] Himmelblau, D. "Applied Non linear Programming". McGraw-Hill, New York, 1972.

.

[13] Kirkpatrick, S., Gelatt,C.D. and. Vechi, M.P., "Optimisation by Simulated Annealing", Science Vol 220 pp 671-680, 1983.

[8] Matyas. J., "Random Optimisation" Automation and Remote Control, Vol 24 pp 1031-1043, 1965.

[9] Metropolis N., Rosenbluth, A., Rosenbluth, M.,Teller A., Teller E., J., "Equation of state calculations by fast computing machines", Chem. Phys. Vol 21 pp 1087, 1953.

[10] Press W.H., Flannery, B.P., Teukolsky, S.A., Vetterling, W.T., "Numerical Recipes in C" Cambridge University Press pp 673-680, 1988.

[11] Solis, F.J. and Wets, R. J-B., "Minimization by Random Search Techniques" Mathematics of Operations Research Vol 6 # 1 pp19-30, February 1981

[23] Bilbro, G.L. et al, "Global optimization of functions with many minima using simulated annealing: INTEROPT program", Trans IEEE SMC, 840-849, 1991.

[12] Wismer, D.A. and Chattergy, R., "Introduction to Nonlinear Programming" North Holland, New York, 1978

[?] Cohen, H.A., and You, J., "A multi-scale texture classifier based on multi-resolution "tuned" masks", Pattern Recognition Letters, Vol 13 No 8 August 1992, 599-603.

[?] Benke,K.K. and Skinner,D.R., "Segmentation of visually similar textures by convolution filtering", Aust. Computer J, Vol 19, pp 134-139, 1987.

[13] Reiss, T.H., "Recognizing Planar Objects Using Invariant Image Features", Springer-Verlag, Lecture Notes in Computer Science, No. 676, Berlin, 1993.

DROPPED

Random search methods have been shown to be applicable to template based object recognition. We have shown that for a well-chosen value of the calculated step factor or range for the calculated searches, random search methods converge quickly. For a 12x12 template and using pixels on a 5x5 sparse grid, and an object in the centre of a 100x100 search space, a random variable step cluster search averaged 1083 steps to find the object whereas a sparse/fine search took with a sparse step of five took 1742 searches.

The use of sub sampled or sparse templates on a regular grid gives a large reduction in search time for a given template size both for random and sparse/fine searches as long as the number of remaining pixels is not too small..

Results showed that random search methods may be applied to grey scale or binary images but binary image searches take longer to converge than grey scale image searches due to the smaller correlation notch for binary images.

A distinct feature of GRS methods is the high variance of the number of steps required to find the object. If there is a large number of objects to find this effect will average out but for a single object to be found this is a potential disadvantage over a sequential search. Also if the object may not be present, it will take a large number of determinations for a GRS method to be confident the object is not there, of the order of five times the average search.. In summary a comparison showing sequential and random search results is shown in the following table:

A particular strategy for a certain object template averaged approximately 400 searches in a 10,000 position search space, about 4% of the search area.

For random search methods, if the object is located close to the boundaries this will increase the search time. Search time for a boundary location will nominally double compared with a non-extreme position as the mismatch region in which the search will converge, is halved. To find an object in a corner is even slower, with only a quarter of the correlation region available. Average search numbers can be expected to be in the ratio 1:2:4 for the open, edge and corner search positions. As the object moves away from the corners or edges, the correlation region will broaden to a full circle and object location will speed up.

The following discussion is for flat distribution random search steps rather than the Gaussian search steps used by Matyas and Solis and Wets. In random search techniques, searches are made at coordinates that are purely random, ie have no connection whatever with previous search results or other steps which are made close to the previous best match in order to find the minimum close to the point of best match found so far. The number of calculated local searches per random search to find the point of best match is of considerable interest. The following results show the effect of varying the number of calculated steps from 1 per random search to 4 calculated steps per random search step.
See the diagram below for search location positions.

Keywords: Random search, object location, Matyas Algorithm

Image Database References that MIGHT be used:

[?] Wayne Niblack and Myron Flickner, “Find Me the Pictures That Look Like This: IBM's Image Query Projec”,, Advanced Imaging, April 1993, pp 32-35.

[?] A. Pentland, R.W. Picard and S. Sclaroff, “Photobook: Tools for content-base manipulation of image databases”, in Proc Society for Optical Engineers, SPIE Storage and Retrieval of Image and Video Databases II, San Jose, California, Feb 1993.

[?] Rosalind W. Picard and T. Kabir, “Finding Similar Patterns in Large Image Databases”, IEEE ICASSP 93, Minneapolis, Vol V, pp V-161-164.

[?] Rosalind W. Picard and Fang Liu, “A New Wold Ordering for Image Similarity”, IEEE ICASSP 94, Adelaide, SA Vol V, pp V-129-132.

[?] V.E. Ogle, Chabot: “Retrieval from a Relational Database of Images”. IEEE Computer,Vol 28, No 9, Sept 1995, pp 40-48.