Search-based Evolutionary Operators for Extensionally-defined Search Spaces: Applications to Image Search

Colin G. Johnson
School of Computing
University of Kent
Canterbury, Kent, UK
http://www.cs.kent.ac.uk/people/staff/cgj/

***Abstract*—This paper explores the idea of applying evolutionary algorithms to those search spaces that are defined *extensionally*, i.e. by listing every item in the space. When these spaces are with a function that returns similar elements given a key element, analogies of mutation and crossover can be defined. This idea is discussed in general, and specific examples are given where the search is for images, in particular where image search is carried out using an interactive genetic algorithm.**

I. Introduction

The search space that evolutionary algorithms explore has traditionally been defined *intensionally* [1]—that is, a definition is given of what objects belong in the search space, but no explicit list of those objects is created. Contemporary web search technology and web services are beginning to make possible the idea of an *extensionally* defined search space possible. That is, the search space is a large collection of objects: the text from a large collection of web pages, a large collection of images or sounds.

The aim of this paper is to explore the role that evolutionary computing methods might play in exploring such a search space. In particular, can a large-scale search task (e.g. finding an image with particular characteristics in a vast search space of images) be carried out by an evolutionary algorithm that uses extensionally-defined local search algorithms, potentially available as web services, such as requesting similar images? This idea of image search will be used as the motivating example for this paper, in particular image search through interactive genetic algorithms, though the ideas are generic and could readily be applied elsewhere.

The technology for implementing such systems is still in its infancy, for a variety of technical and economic reasons. Therefore, many of the ideas in this paper rely on services that are only available in limited or experimental forms—yet, which seem likely to become readily available in the next few years. Nonetheless, a couple of simple examples have been implemented, which are presented towards the end of the paper.

The paper is structured as follows. Section II provides a review of relevant literature and technology. Section III describes the core ideas, and then Sections IV and V provide simple. implemented examples.

II. Literature and Technology Review

This section reviews the relevant research literature and technology in two areas: the idea of navigating search spaces with interactive genetic algorithms, and the web-based technology available for accessing large image databases.

A. Navigating Images with Interactive Genetic Algorithms

Interactive genetic algorithms (IGAs) have been applied to navigate spaces of images. An IGA is a genetic algorithm where the fitness is allocated by a human user: the population is presented to the user, and this user assigns a score, rank or simple binary yes/no choice to each item in the population [2]. The remainder of the GA then progresses as in the standard GA—individuals are selected for mutation and/or crossover based on this fitness evaluation, and this is then used to construct a new population that is then presented to the user, and the cycle begins again.

This has been used in two main ways. Firstly, as a way of allowing a user to explore a complex space, perhaps with no "end goal" in mind. This is typified by one of the earliest examples, Dawkins's *Biomorphs* program [3], where simple pixel-based images are generated from a list of parameters (see Figure 1). A more complex example is given by the recent terrain-generation experiments by Walsh and Gade [4], where landscape are generated using fractal terrain-generation algorithms. Similar ideas have been applied to a number of visual and audio-based explorations, particularly where the aim is artistic or subjective.

Secondly, these are used to explore spaces where there is an end goal in mind, but closeness to that end goal is difficult to define up front and place in a fixed fitness function and so human feedback is the best way of providing the fitness. For example, such systems have been used for the creation of "facial composites", images that are used in the detection of criminals [5] (see Figure 2). This is particularly effective in this application because individuals are good at recognising faces but less good at describing them, so such a navigation

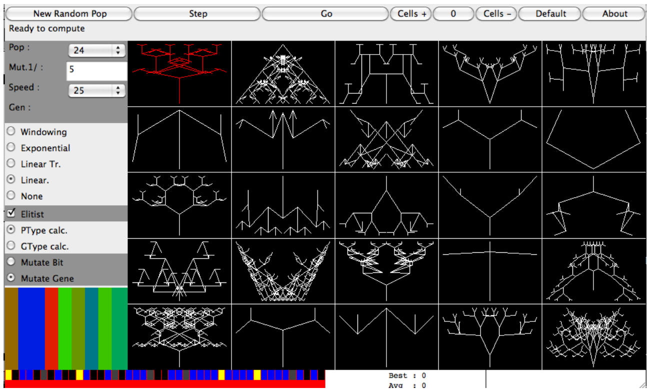

Fig. 1. An implementation of the Biomorphs program: screenshot of a program available from http://www.rennard.org/alife/english/gavgb.html.

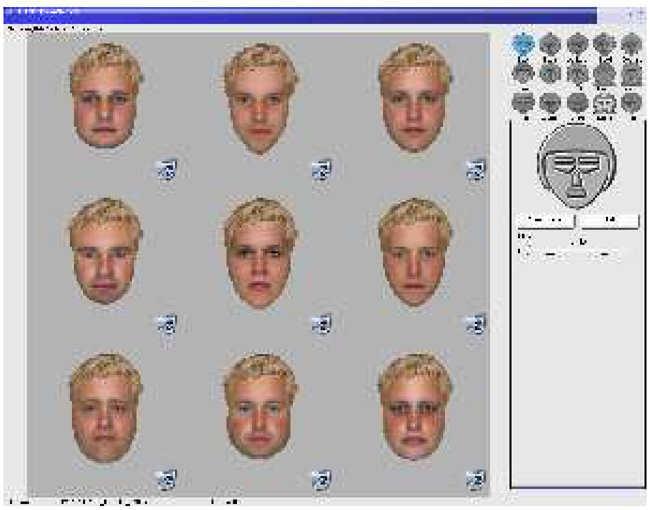

Fig. 2. User interface to a facial composite system—from www.visionmetric.com.

system works well with this feature of the human perceptual system.

In all of the above systems, the images were constructed from a list of parameters. In the biomorphs system, the images were constructed via some simple, but surprisingly fecund, developmental rules. In the face exploration system faces were constructed via the combination of "eigenfaces", which were derived from a corpus of facial images.

In this paper we would like to explore the use of search-based operators. That is, instead of mutation carrying out a parameter change with concomitant changes to the image generated by that individual, population members will consist of images drawn from a large collection, and mutation will consist of swapping the image for another one close by according to some algorithm. Before we can discuss this further, we need to give some background on image collections and image search, which is given in the next part of the paper.

B. *Image Collections*

Large collections of images are now readily available. These are of two types. The first is the intentionally-created image collection, as typified by the *Flickr* website. Around half a million images are uploaded to this every day [6], and there are currently over 6 billion images available to the public on the site [7]. The second type of collection is that obtained by web-crawling and made available through a search engine: a canonical example here is *Google Images*, which indexes around 10 billion images [8].

Finding an image in such a collection can be carried out in two main ways. The first is where the user inputs one or more words to be used as the basis of the search. These are then compared with lists of words that have been associated with the image, which can happen through a number of processes: the creator of the image might tag the image, other users might add tags, or words might be added automatically e.g. by extracting nearby words from a web page containing the image.

The second main way is via a *content-based image retrieval* (CBIR) or *visual search* system [9], [10]. The idea of such a system is that it takes an image and provides images from the collection that are visually similar to that image. Quantifying a subjective quality such as image similarity is difficult—a large number of systems exist, but none are regarded as canonical. Visual similarity algorithms typically consist of a combination of several aspects: overall colour distribution within the image, the presence/absence of certain textures or shapes within the image, measures of image complexity. All of these can furthermore be applied to similarity between a grid of *patches* within the image.

There are currently a number of web-based systems available: an overview of a sample of them is given in Table I. Some explore a wide range of images, other a smaller set. Some allow the user to upload images, some will only begin from an image chosen by a word-based search. With one, very preliminary, exception, none of these have an API (application programming interface) allowing it to be programmed directly. The development of programmable-based systems will depend on the emergence of a meaningful economic model for such services, which currently depend on the incidental viewing of advertising as their source of revenue, which would not happen in a programmable system.

This lack of programmability is a significant hurdle to the development of substantial applications of this kind. There is a bootstrapping issue here: until a market for applications of this kind exists, search companies will have little motivation to allow (paid) API-based access to their services; without ready access to such services, the motivation of developers to develop this kind of application is small. One way (as exampled below) is to prototype systems using so-called *screen-scraping/web-scraping* methods [11], which take material intended to be interpreted by a human reader via a web browser (for example, the results of a dynamic web query) and processes this with a program that is designed to extract

System Name	Location	Properties
Google *similar images* search	www.google.com/imghp?hl=en&tab=wi	Initially starts from a word-based input, but can then do a similarity-based input. As an alternative, a very experimental (and only available from time-to-time) image-based input is available where the user uploads their own image. Searches a vast collection of images: around 10 billion.
GaZoPa	www.gazopa.com	No longer available. Allowed the upload of images for use as the starting point for comparison. Appeared to access a very large number of images.
Tiltomo	www.tiltomo.com	Begins with a word-based search, after that allows search based either on visual (texture/colour) based similarity or on word-based similarity using Flickr tags. Database consists of c. 140,000 Flickr images.
Macroglossa	www.macroglossa.com	Searches within a number of specialised collections (animals, panoramas, etc.) Rather slow. An experimental API is available.
Tineye	www.tineye.com	Allows the uploading of images by the user as the starting point for image similarity search. Only finds very close matches to the original image—designed primarily to find web-pages that reproduce the supplied image or minor variants of it such as cropping and resizing.
Flickr	www.flickr.com	No visual search option available, but images are often heavily tagged, which facilitates a different kind of search.
Live	www.live.com	Requires an initial word-based search, after that allows a search based on similarity.
BYO Image Search	labs.ideeinc.com/upload	Allows the upload of an initial image by the user. Has a database of around 3 million images.
Incogna	www.incogna.com	Requires an initial word-based search, after that allows a search based on similarity.

TABLE I
A SAMPLE OF WEB-BASED IMAGE SEARCH SYSTEMS.

the relevant information.

III. Evolutionary Algorithms on Extensionally-defined Search Spaces

This section explains how evolutionary algorithms could be defined on extensionally-defined search spaces. The first part describes the difference between intensionally- and extensionally-defined spaces; definitions of evolutionary operators for extensionally-defined spaces are given, and then made more concrete by being discussed in the specific context of image based search.

A. Intensionally and Extensionally Defined Spaces

There are two main ways in which a (natural) language defines a space of objects. One of these is the *intensionally* [1]. This is where necessary and sufficient conditions for something to be part of the space are given. The alternative is an *extensional* definition, where all elements in the space are individually listed.

Evolutionary algorithms are typically described as exploring an intensionally-defined space. Typically, this definition is given by specifying a number of *slots* and a range of values that could be taken by each of those slots—this is the *genotype* of the algorithm. For example, in an n-dimensional function optimization algorithm, there might be n slots each of which takes a floating point number in a particular range. The modification operators—typically mutation and crossover—are then defined as functions on those genotypes. A fitness function will sometimes take this genotype more-or-less directly, or it might form the basis of some complex processing (e.g. in developmental systems [12] or genetic programming [13]) to convert it into a phenotype that is used for fitness evaluation.

B. Evolutionary Operators on Extensionally-defined Spaces

This section explores how evolutionary operators can be defined on extensionally-defined spaces. Let us assume that we have a space $E = \{e_1, e_2, \ldots, e_{n_e}\}$ of possible solutions in the search space. As an example, consider all of the photographs on *Flickr* or *Google Images* as an example of E. These individuals can be retrieved from E in a number of ways: by a unique ID; by a content-based similarity search where an object o of the same type to those in E but not necessarily in E is presented and a subset of E returned that are similar, for some problem specific definition, to o are returned; or, by some kind of keyword-based search, where every element in E has a list of tags from some vocabulary associated to it and elements of E can be retrieved that match those tags. Depending on the data source, some of these retrieval methods might not be available.

A concept of *search-based mutation* can now be defined. Consider a population of n_p objects $P = [p_1, p_2, ..., p_{n_p}]$ where $P \subset E$. Imagine that we have a *content-based retrieval* system that interrogates E. This algorithm—call it C—returns a set of objects that are, in some problem-specific way, similar to the input object. So $C(p_i) \rightarrow \{c_1, c_2, \ldots, c_n\}$. A mutation M of a population member p can be defined as a random choice from that set, that is, $M(p) = \text{rnd}(C(p))$, where rnd() is a function that chooses uniformly at random from a set.

A more nuanced form of this could be achieved if C provided some kind of similarity measure or ranking rather than just a set of "similar" objects; or, if C could be provided with a "tolerance" for how much similarity is required before items are returned. This could allow the extensional-equivalent of adjustments of mutation rate.

A search-based crossover is more difficult. One way to implement this would be to implement a search algorithm that looked for objects that had a reasonable level of similarity to both of the parent objects. Another approach would be to use the two parent objects directly to create a rough "proxy" object, which would then be used as the starting point for the search.

C. An Example: Image Similarity Search

This part of the paper explores the practicalities of realizing these kinds of operators for a specific example: that of image search, i.e. where E is a large collection of images. There are two main approaches, depending on the kind of information that can be obtained from the image database being used.

1) Image-similarity Approaches: Some image sources—see Table I for details—can provide a set of similar images from a target images. Image similarity is a complex issue, as humans use many different aspects of images when deciding whether two images are similar. Image similarity algorithms [14] use many features of images as a proxy for this: some of these are global features, e.g. the distribution of colours in an image; some are local features, e.g. approximate similarity of pixels in similar positions; other approaches identify specific features, e.g. shapes or textures, within images and match images based on the amount of overlap in such feature vectors.

For the purposes of this paper, the details of such similarity algorithms are not relevant; we assume that an online service is able to serve such similar image sets on demand from a target image. Using the scheme above, a mutation operator can readily be defined. Defining crossover is considerably more complex. A *proxy* image could be created by a rough combination of the two parent images, e.g. by cutting out features of the two images and combining them together, and then this searched for.

2) Metadata-based Approaches: An alternative is to use the metadata that is frequently found in image databases. Most collections of images annotate each image with a number of *tags*, words that say something about the image and which are provided by human who either created or who viewed the images. Systems such as *Flickr* place most of the responsibility of this onto the creator of the image, who has to generate a list of such tags. Other systems, such as Google Images, rely on a mixture of contextual information (e.g. the text that is found near to the image on the web page where the image was found) or human annotation by people other than the creator of the image. In particular, Google used a *game with a purpose* [15] in the form of the Google Image Labeller, which paired random pairs of people to provide labels for images.

Given such tag lists for images, we can define mutation in a number of ways. For example, we might simply re-search for images with the same list of tags, or we might remove tags so that the mutated image is drawn from a broader set of possible images. The obvious analogy with traditional mutation—taking one tag at random and replacing it with another tag—is unlikely to work, as there is too much variety in tags so the chance of finding an image with that particular tag list is minimal. Crossover can be defined similarly, by sampling a number of tags from both parent images to find a new tag list for search. Again, this has problems: certainly if too many tags are taken from each parent then the chance of successfully finding a picture tagged by that specific mixture of tags is very low. However, using some notion of *similarity* between tags might make this more feasible.

There are a number of problems with the use of linguistic tags. The first is the issue of *polysemy*: a single tag can refer to very different things, as a single word can have multiple distinct meanings. The second is concerned with *level of abstraction*. Tags might be too specific: for example, a picture might be tagged with *Labrador* rather than the more generic *dog*. One strategy to combat this is to use some kind of thesaurus to bring all terms up to single level of abstraction. Indeed, choice of level of abstraction could act as a kind of "mutation rate" adjustment, where abstracting all terms up to a high-level description is useful during a very exploratory phase whereas at a more exploitational phase of the search it might be desirable to use very specific terms. A third issue is that in some systems, e.g. Flickr, users make use of tags to indicate issues relating to the production of the image—most commonly, the kind of camera used—rather than relating to the image itself.

IV. Explorations with Interactive Mutation

The first way in which the system was applied was in an exploratory mode. A program was written based on the Tiltomo image similarity search engine; this allows the user to select an image and find images that are similar to that image either by visual similarity or via similarity of tag-based descriptions. At present, this is a prototype system, using a database of around 140,000 images. Furthermore, no API is available so the program was written using screen-scraping.

The algorithm used was as follows:

```
Initialise population with 9 random images
LOOP (until user decides to stop):
  display the images
  allow the user to select an image for mutation
  pass this image ref to Tiltomo
  receive set of 30 similar images from Tiltomo
  select 9 images at random from those images
END LOOP
```

Initialisation was straightforward as Tiltomo can provide random images. The user interface is illustrated in Figure 5. An example of the search process can be seen in Figure 3; a video of this can be seen at http://www.youtube.com/watch?v=SK53wxBmwTk.

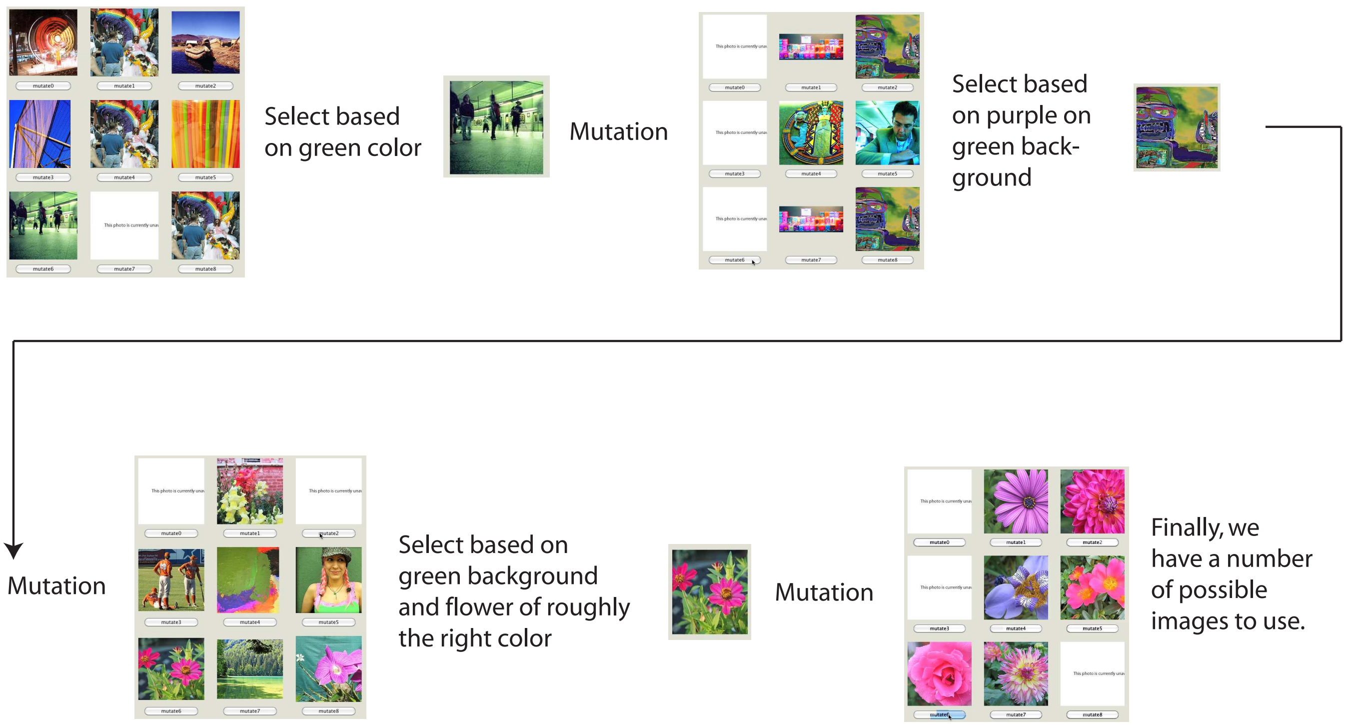

Fig. 3. Searching for a particular image type using the mutation-based search interface. The target that was in mind was a purple flower on a background of green leaves.

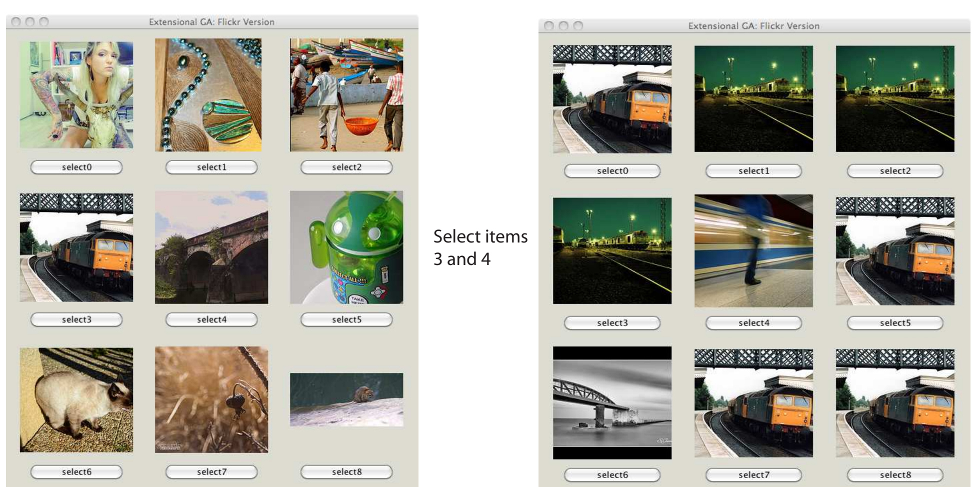

Fig. 4. A meaningful crossover? A train and the arch of a bridge are selected, one of the results is a railway bridge.

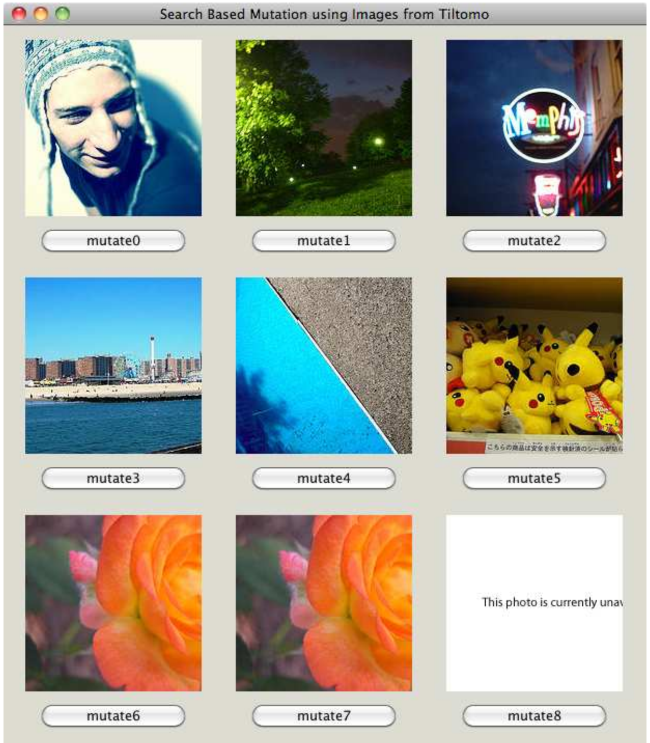

Fig. 5. The user interface to the image similarity based extensional search algorithm.

Overall, this serves as a basic illustration of the capacity of such extensionally-based systems. However, it can be argued that little has been added here to what could be achieved directly from a website such as Tiltomo or Google Images. We have reframed the way of acting with such a system by calling it a "genetic algorithm" and the similarities "mutations", which encourages users to think of their interactions with a system such as this as an iterative process rather than a one-off process, but have not really added to the capabilities of such websites.

V. Experiment with Tag-based Crossover

A second experiment explored the idea of tag-based crossover. This was based on the Flickr API (www.flickr.com/services/api), in particular the FlickrJ Java interface (flickrj.sourceforge.net). These allow, amongst many other things, for the user to search for images by tags.

The algorithm used was as follows:

```
Initialise population with 9 random images
LOOP (until user decides to stop):
  display the images
  allow the user to select two images
  get the tag lists for these from Flickr
  LOOP: for each member of the population
    create a child list by choosing a tag
      from each list
    look up images matching these tag lists
      on Flickr and add to new population
  END LOOP
  replace old population with new population
END LOOP
```

Random images were created by searching for a random tag drawn from the 200 "picturable words" from *Basic English* [16]. If an image with the full tag list is not able to be found, then the Flickr API defaults to searching for a subset of the tags submitted. This is typical of how many such user-focused APIs work: if the task cannot be completed, then an approximation to the task is carried out, rather than the system giving up entirely.

On the whole this was less successful than the image similarity based system. Even with just two tags, it was often the case that no exact match could be found, and so the system would default to a single tag and therefore carry out no crossover at all. An example of a successful crossover is illustrated in Figure 4, where a crossover where the two images selected are a train and the arch of a bridge, and the final result is a railway bridge.

To make such a system as this work well, a much more sophisticated approach to the language of tags is needed, perhaps based on the idea above of pulling all tags up to a particular level of abstraction, removing irrelevant tags such as camera-type, and so on.

VI. Conclusions and Future Work

This paper has introduced the idea of search based on *extensional* search spaces, where the space being searched consists of a large collection of individual objects. We have used the example of large image databases in this, and shown how the availability of online image similarity search, and the use of metadata such as image tags,

It is difficult to create decent-quality systems of this kind at present, and therefore to evaluate them, as there is little availability of reliable APIs for such databases. However, it is important to develop prototype systems of the kind discussed in this paper at this stage, in order encourage the development of APIs.

Ways to improve systems such as this include:

- The reporting of similarity measures as well as such a list of similar objects. This would allow the adjustment of "mutation rates" and similar.
- The development of a more sophisticated way of handling the tags to allow mutation and crossover to work with language at a single level of abstraction.

Such techniques could also work with a "target image". For example, the user could provide a rough sketch of the desired image, and an algorithm could search by using an image similarity algorithm between the population members and the sketch rather than in an interactive mode. These ideas might also be capable of being applied to text and sound applications as well as images.

References

[1] R. Carnap, *Meaning and Necessity*. Chicago: University of Chicago Press, 1947, enlarged edition 1956.

[2] H. Takagi, "Interactive evolutionary computation: Fusion of the capacities of EC optimization and human evaluation," *Proceedings of the IEEE*, vol. 89, no. 9, pp. 1275–1296, 2001.

[3] R. Dawkins, *The Blind Watchmaker*. Penguin, 1990.
[4] P. Walsh and P. Gade, "Terrain generation using an interactive genetic algorithm," in *Proceedings of the 2010 IEEE Congress on Evolutionary Computation*. IEEE Press, 2010, pp. 1–7.
[5] S. Gibson, C. Solomon, M. Maylin, and C. Clark, "New methodology in facial composite construction: from theory to practice," *International Journal of Electronic Security and Digital Forensics*, vol. 2, pp. 156–168, 2009.
[6] Flickr Blog. (2010, September) 5,000,000,000. Viewed November 2011. [Online]. Available: http://blog.flickr.net/en/2010/09/19/5000000000/
[7] ——. (2011, August) 6,000,000,000. Viewed November 2011. [Online]. Available: http://blog.flickr.net/en/2011/08/04/6000000000/
[8] Official Google Blog. (2010, July) Ooh! Ahh! Google Images presents a nicer way to surf the visual web. Viewed November 2011. [Online]. Available: http://googleblog.blogspot.com/2010/07/ooh-ahh-google-images-presents-nicer.html
[9] M. S. Lew, *Principles of Visual Information Retrieval*. Springer, 2001.
[10] M. S. Lew, N. Sebe, C. Djraba, and R. Jain, "Content-based multimedia information retrieval: State of the art and challenges," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 2, no. 1, pp. 1–19, 2006.
[11] M. Schrenk, *Webbots, Spiders, and Screen Scrapers*. No Starch Press, 2007.
[12] S. Kumar and P. Bentley, Eds., *On Growth, Form and Computers*. Academic Press, 2003.
[13] R. Poli, W. B. Langdon, and N. F. McPhee, *A Field Guide to Genetic Programming*. Published via `http://lulu.com` and freely available at `h ttp://www.gp-field-guide.org.uk`, 2008, (With contributions by J. R. Koza). [Online]. Available: http://www.gp-field-guide.org.uk
[14] A. Shrivastava, T. Malisiewicz, A. Gupta, and A. A. Efros, "Data-driven visual similarity for cross-domain image matching," *ACM Transaction of Graphics (TOG) (Proceedings of ACM SIGGRAPH ASIA)*, vol. 30, no. 6, 2011.
[15] L. von Ahn, "Games with a purpose," *IEEE Computer Magazine*, pp. 96–98, June 2006.
[16] C. K. Ogden, *Basic English: A General Introduction with Rules and Grammar*. London: Paul Treber, 1930.