Interactive Evolutionary Algorithms for Image Enhancement and Creation

Joseph James Mist

Forensic Imaging Group

School of Physical Sciences

University of Kent, United Kingdom

A Thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Supervised by: Dr Stuart James Gibson

September 2008 - April 2014

Approximate word count: 54,000

Abstract

Image enhancement and creation, particularly for aesthetic purposes, are tasks for which the use of interactive evolutionary algorithms would seem to be well suited. Previous work has concentrated on the development of various aspects of the interactive evolutionary algorithms and their application to various image enhancement and creation problems. Robust evaluation of algorithmic design options in interactive evolutionary algorithms and the comparison of interactive evolutionary algorithms to alternative approaches to achieving the same goals is generally less well addressed.

The work presented in this thesis is primarily concerned with different interactive evolutionary algorithms, search spaces, and operators for setting the input values required by image processing and image creation tasks. A secondary concern is determining when the use of the interactive evolutionary algorithm approach to image enhancement problems is warranted and how it compares with alternative approaches. Various interactive evolutionary algorithms were implemented and compared in a number of specifically devised experiments using tasks of varying complexity. A novel aspect of this thesis, with regards to other work in the study of interactive evolutionary algorithms, was that statistical analysis of the data gathered from the experiments was performed. This analysis demonstrated, contrary to popular assumption, that the choice of algorithm parameters, operators, search spaces, and even the underlying evolutionary algorithm has little effect on the quality of the resulting images or the time it takes to develop them. It was found that the interaction methods chosen when implementing the user interface of the interactive evolutionary algorithms had a greater influence on the performances of the algorithms.

Acknowledgments

I would like to thank the following people for their respective contributions:

- My supervisor Stuart Gibson for providing this opportunity and for his patience through my various difficulties.
- Matthew Maylin for his help with various computing matters.
- My parents Joseph and Linda and my brothers Thomas and Tobias simply for being.
- My late grandmother Flossie Flippence and my uncle Bryan Flippence for providing me with somewhere cheap to live during my studies.
- The various people I have known in the School of Physical Sciences during my studies, in particular: David Pickup for his help on matters relating to Linux, David Clarke and Marianne Riggs for the good times outside the office, and Susan Welford and Agata Makiela for brightening my days.

Publications

Publications with content originating from the work presented in this thesis.

- C.J. Solomon, S.J. Gibson, and J.J. Mist. Interactive evolutionary generation of facial composites for locating suspects in criminal investigations. *Applied Soft Computing* (2013).

- J.J. Mist and S.J.Gibson. Optimization of weighted vector directional filters using an interactive evolutionary algorithm. *Proceedings of the fifteenth annual conference on Genetic and evolutionary computation conference companion 2013* Pages 1691–1694.

Contents

List of Figures

Chapter 1

Introduction

Digital images have become ubiquitous over the past two decades. Infrastructure engineering vice president of social networking website Facebook Jay Parikh said in an interview in 2012 that Facebook gets “300 million photos up every day” [103]. Digital images are encountered in many forms and are created for varying purposes and from various sources. Digital images can be created for utilitarian reasons; two examples of which are medical images to aid the diagnosis and treatment of diseases and architectural models which provide an indication of what a project will look like once it has been completed. Digital images can also be created for aesthetic reasons — to create an image for artistic reasons but using a computer as opposed to traditional artist materials. The majority of the images people see are created with both utilitarian and aesthetic concerns taken into account. For example, people capture photographs in order to make a record of events, but they also try to make such photographs aesthetically pleasing.

Digital images require various degrees of human input in their creation. Digital illustrations and architectural models require a lot of human input as they are given form from the thoughts and ideas of the person creating them. Modern compact cameras, on the other hand, are designed such that in most cases there is no need for human input beyond pointing the camera at the scene to be captured and pressing a button. There are situations in which images require, or could benefit from, some small amount of human input. The amount of human input required for an image enhancement or image creation tool can be as little as setting a few input values. Such a tool may be an image enhancement tool like the contrast enhancement processes of Chapter 6 or a piece of software that creates images using mathematical algorithms such as the facial composites of Chapter 7.

In order for an image enhancement or an image creation tool to take user input there needs to be a user interface. User interfaces are designed to be as simple as possible whilst at the same time delivering an appropriate degree of control. However, it can still be difficult for novice users to identify the tools they need and the user may only wish to use the software in order to perform a single task on only a few occasions. It can be frustrating to spend more time learning how to use a piece of software than actually using the tools to perform the intended task. Alternatively, a user may know how to set values explicitly to achieve a desired result but they do not know what their desired result is; the user may simply wish to explore the options provided by the tool.

The approach explored in this thesis is to relieve the user of the burden of setting the input values of an image creation or image enhancement tool by having a computer algorithm determine multiple sets of input values, creating a number of images from these values, and having the user evaluate the resulting images. The input values which correspond to the images preferred by the user are adjusted in a stochastic manner and the resulting values are then used to create more images. The underlying algorithms used to do this are a special form of *evolutionary algorithms* (EAs). EAs are a group of problem solving algorithms which are:

metaheuristic EAs solve problems using trial and error as opposed to using a deterministic method.

population-based The algorithm maintains a number of potential solutions to the problem which are replaced over time with better solutions.

biologically-inspired The operators used in an EA superficially mimic evolutionary processes found in nature.

More information is given about EAs in Section 2.1.1.

EAs have been used to tackle image enhancement problems. A method for contrast enhancement in monochrome images was described by Munteanu and Rosa [85]. Hoseini and Shayesteh [56] used a hybrid of an EA and other nature-inspired algorithms for the same purpose. In image segmentation Harvey et al. [50] used an EA to develop an automated feature detection/classification system to segment multichannel satellite images. Ghosh and Mitchell [40] built upon the work of Harvey et al. and used an EA to develop a method for segmenting computed tomography (CT) images. Singh et al. [111] used an EA to develop a process for segmenting cells in biological images.

EAs rely on fitness functions (objective measures of goodness) to decide which potential solutions are the best from those created. There are problems in image enhancement and creation for which the use of an objective function may not be suitable. This is likely to be the case when the purpose of an image is aesthetic (such as in art photography) rather than utilitarian (such as a medical scan). It is human evaluation which ultimately determines how good an image is, so it is also reasonable that human evaluation should form a part of the image development process. The EA approach can be modified to include human evaluation by replacing, or at least partially replacing, the fitness function with human evaluation. An EA adapted to use human input in this way is called an *interactive evolutionary algorithm* (IEA).

IEAs have been applied to image enhancement and creation problems. In the field of image enhancement, Poli and Caponi [100] used an IEA to develop a pseudo-colouring scheme for echocardiographic image enhancement. In image segmentation, Otobe et al. [88] used an IEA to segment foreground plant matter from background dirt in photographs of plants growing in a field. Examples of IEAs being used in design tasks include Fons et al. [142] who used an IEA/EA hybrid in the construction of a tool which provides novelty in architectural design. Gong and Guo [42] used an IEA in a tool for designing ladies' outfits. IEAs can also be applied to fractal based art tasks such as the development of virtual landscapes as demonstrated by Walsh and Gade [140].

There is a good deal of variation with regards to the amount of content in papers published in the field of IEAs that is not purely due to paper length restrictions. Ideally, a paper should contain:

- A description of the task to be undertaken using the IEA and details of how the process by which the input values are used to achieve the task can be implemented.

- Details of the IEA used, including the interface, with particular detail to novel aspects of the IEA.

- Comparison of the IEA approach to other methods of performing the same task, or for work in which new algorithmic design options [1] are introduced, comparisons to existing algorithmic design options.

[1] Algorithmic design option is a term used in this thesis to cover the various options available when implementing an IEA. For example, if a paper presents a new mutation operator (see Section 2.1.1) then it is presenting a new algorithmic design option.

- Statistical analysis of the data gathered when making the comparison.

A description of the task and how it can be implemented is desirable because it allows other people to use the task for their own work. In practice the details are usually limited to the required inputs; the process by which the inputs are used to achieve the task is omitted. This is generally because the implementation of the process is too involved to be covered in the space available. For example, Walsh and Gade [140] applied an interactive genetic algorithm (IGA) to set the input values of a process which generated virtual landscapes. Whigham et al. [142] applied an IEA to a flag design task, the details of the process that turned the inputs into flags were too involved to be included. The process that turned the input values into colours in the colour matching task of Breukelaar et al. [10] was simple enough to be fully described in the paper.

Details of the IEA. including the interface used, should be provided because it enables others to reproduce the algorithms used. Work in the field is very strong in this regard and so there is no need to critique its deficiencies here.

If the focus of the work is the application of an IEA to a new task then a comparison to an existing approach to performing the task should be performed. If the focus is the introduction of a new algorithmic design option, then a comparison should be made to equivalent design options. Comparisons are important because it is desirable to know if the new application or algorithmic design option represents an improvement. Comparisons are often omitted if the work presented in the paper is considered preliminary and is only intended to provide proof of concept. This is more commonly the case when an IEA is applied to some new task. The papers by Walsh and Gade [140] and Whigham et al. [142] do not provide comparisons as the intent of the papers is to provide proof of concept. Ueda et al. [133] applied an IEA to an image enhancement task and although they gathered user satisfaction data they did not actually compare the IEA approach with other methods. Poli and Cagnoni [100] applied an IEA to the task of highlighting the differences between two medical images using pseudocoluring. No comparison was made to other methods nor was any evaluation performed beyond a visual inspection of the images by the authors.

When making comparisons between IEAs or between IEAs and alternative methods appropriate measures (ideally more than one) need to be used. Comparisons on multiple measures are important; choosing to compare on only one measure can lead to a comparison between algorithms or approaches that is misleading. For example, if only user satisfaction of the final images created by the participants is measured, the measure may marginally favour Algorithm A over Algorithm B in which case it may be concluded that Algorithm A is preferable. It may be that if time taken was compared also and it was found that this measure favours Algorithm B significantly then the conclusion is likely to be that Algorithm B is preferable.

In most instances, participant satisfaction with the final images and time taken to achieve a satisfactory image are appropriate measures. If the experiment also includes visible differences between the interfaces, such as when an IEA is compared to a direct input approach, then some measure of the usability of the interfaces is also appropriate. Work involving IEAs is generally satisfactory in this regard, though there are examples of experiments that fail to provide comparisons on some measures when the data required for these comparisons could have easily been collected. Lee and Cho [70] developed a smartphone application for image enhancement and compared two IEAs to a piece of commercial image enhancement software. Usability data was gathered and compared but the time taken to enhance the photographs and satisfaction with the final images was not. Yoon and Kim [145] used a photograph effects task to compare the performances of three scales for rating the images. User satisfaction scores (it is not clear if this means satisfaction with the images or usability of the scales) were compared but not the time taken. Oinuma et al. [87] compared four recombination (see Section 2.1.1) methods using a face image beautification task. Time taken was not recorded, nor was participant satisfaction with the images. Time taken is not always an appropriate measure of comparison. This normally the case if an IEA is being compare to some automated approach such as in [83, 62]. Other measures of evaluation may be appropriate. Gong et al. [44] used a fashion design task to compare three surrogate models for users. The number of generations the IEAs required to achieve a satisfactory design and the number of evaluations required by the participants were compared.

If comparisons have been performed stating that, for example, the approach or algorithm with the greatest mean participant satisfaction rating is the best is not satisfactory. It is important to perform statistical analysis of the data to establish whether any observed differences between algorithms and approaches are genuine or if they are due to chance before concluding that one algorithm or approach has outperformed another on some measure. The data collected in experiments involving human participants are generally noisy and so what may appear to be a large difference between mean values could be due to chance. Very little of the work related to the use of IEAs for image enhancement, image creation, or design tasks uses statistical analysis to compare the performances of the algorithms. Most of those that could be found are considered here.

It is common practice in Psychology to apply statistical methods, such as t-tests and ANOVA [116], which are appropriate only for data that meet particular requirements on data that do not. It is likely that authors of the work in which this practice is evident derived their methodology from the field of Psychology. There is also the possibility that the data were in fact suitable for parametric tests and so this practice shall not be criticised. Another practice that was observed was the use of statistical tests designed to compare the data from two treatments being used in a pairwise manner to compare data from more than two treatments. This was done in [66] where three treatments were compared in a pairwise manner. The novel algorithmic design options introduced in [123] and [122] were compared in a pairwise manner to three other algorithmic design options. The correct practice is to perform a multiple comparison test, which tests for statistical significance over all treatments, and then perform an appropriate post hoc test on pairs of treatments. Two examples were found of a multiple comparison test being used when a test for comparing two sets should have been used instead [62, 17]. Two papers were found which used statistical tests correctly: [77] which used the Wilcoxon signed rank test to analyse ratings given by participants to two images and [126] which used Friedman's test with Scheffe's post hoc criterion to analyse the data on four measures used to compare three interface/algorithm combinations.

The consequence of these deficiencies is that conclusions drawn in much of the existing work are not well founded and should be treated with caution, particularly when it is stated that one IEA or method is preferable to another when in fact statistical analysis would reveal no significant difference.

The main contribution of this thesis is to make robust comparisons between algorithmic design options in IEAs, such as choice of mutation operator, and, to a lesser extent, between IEAs and alternative methods. The comparisons are made robust by the use of an adequate number of participants, ensuring that comparisons are made according to multiple appropriate measures (where possible), and performing appropriate statistical analysis on the data. A minor contribution is the introduction (and testing) of a simple method designed to shape search spaces based on user rejection of unfit images.

In Chapter 2 points raised by Takagi [124] and Lewis [72] and observations reported in IEA papers are brought together into a formal summary of the basic parts of an IEA presented in a similar manner to the summary of the parts of an EA in the textbook by Eiben and Smith [29]. Suitability considerations to assess the applicability of EAs laid out by Mitchell [81] are adapted to form a similar list for IEAs. Arguments are laid out as to why some EAs are likely to be better suited for use in IEAs than others. In Chapter 3 a select multiply mutate interactive evolution strategy (SMM-IES), an implementation of an SMM-IEA [41] is used to develop weighted vector directional filters [132] which are compared to one developed using an EA [74] and the basic vector directional filter. In Chapter 4 a novel extension to the SMM-IES, the hyperplane-IES is introduced and compared to the SMM-IES using a colour matching task very similar to those used in [10] and [16]. A virtual user is developed and used to set the parameters of the hyperplane-IES in a similar manner to those used in [91] and [34]. Chapter 5 reports an experiment using the colour matching task in which a hyperplane interactive genetic algorithm (hyperplane-IGA) is compared to a simple IGA based on that used in EvoFIT [34]. Two search spaces are also compared, one implemented for its convenience and the other for its perceptual uniformity in a similar manner to the comparison made by Sugimoto and Honda [121]. The experiment is performed both with the target colour present as in [10] and without. In Chapter 6 the simple IGA is compared to a bespoke slider based interface for setting the input values to two image contrast enhancement processes: an intensity transfer function process [106, 51] and a simple compound process. In Chapter 7 a facial composite creation task based on the same principles as EFIT-V [115] is used to compare the performances of two established mutation and two recombination operators for the simple IGA [29]. The same task is used to compare three search spaces one of which is constructed based on human prioritisation as in [59].

Chapter 2

General theory

2.1 Evolutionary algorithms

As stated in Chapter 1, an evolutionary algorithm (EA) is a population-based metaheuristic nature-inspired optimisation algorithm. EAs are so named because they loosely mimic the processes of biological evolution. EAs are general problem solving algorithms meaning that they can be applied to a wide range of problems, though the use of an EA may not necessarily be the best approach to a particular problem.

2.1.1 Aspects of an EA

A short summary of the basic aspects of an EA is given here. As this thesis is concerned with IEAs, only the parts of an EA which differ from the equivalent parts of an IEA are described in any detail. This summary is adapted from Eiben and Smith [29] and Mitchell [81]. The basic structure of an EA is given in Algorithm 1.

```
INITIALISE population with initial candidate solutions;
EVALUATE each candidate;
while termination condition is not satisfied do
    SELECT parents;
    RECOMBINE pairs of parents;
    MUTATE the resulting offspring;
    EVALUATE new candidates;
    SELECT individuals for the next generation;
end
```

Algorithm 1: Basic structure of an objective evolutionary algorithm (after Eiben and Smith [29])

Representation

If an EA is to be used to solve a problem, the problem needs to be expressed in a form that allows an EA to be used. In an EA candidate solutions to a problem are known as *phenotypes*. What exactly constitutes a phenotype is a little ambiguous in image enhancement and creation. Otobe et al. [88], using an IEA to perform an image segmentation task, referred to the developed segmentation processes as phenotypes. Poli and Cagnoni [100], using an IEA to find a pseudocolouring scheme which highlights the differences between two images, referred to the images themselves as phenotypes. Munteanu and Rosa [86], who used an EA to optimise intensity transfer functions for contrast enhancement of greyscale images, did not refer to phenotypes at all but the term 'individuals' was used to refer to the transfer functions, not the images. Hashemi et al. [51], in similar work, also refer to the transfer functions as individuals. In neither [86] nor [51] were the transfer functions applied to images other than those they were developed on; whether the phenotype was the process or the image was irrelevant. Breukelaar et al. [10], in a colour matching task, referred to the colour panels which were displayed to the user as individuals; not the red, green, and blue values of the colours. There is no standard definition for what constitutes the phenotype when using EAs for image enhancement and creation; whether it is the image process or the resulting image itself. It is for this reason that it is necessary to define what a phenotype is in the context of this thesis. In this thesis, if a process is developed on an image which can be taken and applied, whether successfully or not, to another image then that process is the phenotype. If there is no such process then the output image is defined as the phenotype.For example, the filters of Chapter 3 and the contrast enhancement processes of Chapter 6 are phenotypes. If no such process is created then the image itself is the phenotype. For example, the colours generated in Chapters 4 and 5 and the faces generated in Chapter 7 are phenotypes. Whilst it is the best phenotypes that are if interest, the search is conducted by manipulating *genotype*s in a *search space*. EAs require genotypes to represent the phenotypes during a search, mainly because genotypes provide a means of reducing the search from an infeasibly high number of dimensions to a more manageable number. An element of the genotype such as a single dimension in a vector representing an input value is known as a *gene*. The genotypes of the filters in Chapter 3 are ten-dimensional vectors with nine dimensions to represent the filter weights and the tenth for the step size parameter. The genotypes of the colours in Chapter 4 consist of four-dimensional vectors with three of the dimensions representing colour and one representing the step size parameter. How the phenotypes are represented as genotypes, the *representation*, depends on the EA being used. In an EA each genotype maps to exactly one phenotype, although the reverse is not necessarily true; it is possible for one phenotype to be the phenotype of many genotypes. The genotype, phenotype, or both together (depending on context) are referred to as an *individual*.

Fitness

In the simplest of EAs the genotypes in the search space are mapped to phenotypes which are in turn mapped to a *fitness value* using a *fitness function*. The fitness value of an individual is the measure of 'goodness' of that individual. Fitness values are derived by evaluating phenotypes (not genotypes) on some objective criteria. Fitness values are used to compare individuals during an EA to see which are retained and which are used to create new individuals.

Population

The collection of individuals forms a *population*. A new population of individuals is formed with each iteration of the EA. The population within each iteration is known as a *generation*. The initial population forms the first generation.One of the decisions to be made when implementing an EA is selecting the optimal population size. The most appropriate population size depends on the problem being solved and the specifics of the algorithm being used. A large population is more likely to maintain a high diversity and thus searches more of the search space and has a greater chance of finding a globally optimal solution. However, a large population requires many fitness evaluations per generation. If a single fitness evaluation takes a long time then a search with a large population may take a prohibitively long time. Using a small population means that fewer fitness evaluations are required per generation and, if the problem is amenable, a solution can be found quicker than when a large population is used. However, a small population has a greater risk of converging on a suboptimal solution.

Parent selection

The main point of an EA is that existing individuals are used to generate more individuals, some of which provide better solutions to the problem to be solved. Individuals from which new individuals are derived are called *parents*. Individuals selected as parents are typically added to a *mating pool.* In most EAs it is possible for an individual to be added to the mating pool multiple times in a single generation. There are number of different parent selection methods and each EA has some parent selection methods that are conventionally preferred over others. The IEAs used in this thesis are based on two EAs: *genetic algorithms* (GAs) and *evolution strategies* (ESs). Common selection methods used in these algorithms include *tournament selection*, *roulette wheel selection*, and *stochastic universal sampling.* In tournament selection, k members are drawn at random from the population and the fittest of them is chosen to be a parent. The individuals may or may not be returned to the population afterwards. The process is repeated until the mating pool has as many parents as are needed. A larger value of k means that less fit individuals are less likely to be selected as parents as they will generally lose tournaments to fitter individuals. The term 'Roulette wheel selection' is somewhat inaccurate and requires modification to provide an appropriate analogy for the selection method that bears its name. A roulette wheel has a number of equally sized slots into which the ball can go. When the wheel is spun, the ball has an equal chance of finishing in each slot. If the ball is replaced by a long thin arm which is spun over the slots, which are now fixed, instead of using a ball then a better analogy can be derived. This new wheel allows for slots of uneven sizes in which the probability of the arm pointing to a particular slot is directly proportional to the size of the slot.The wheel is spun as many times as there are parents needed. If one individual has a much higher fitness value than the rest of the population, that particular individual is likely to dominate subsequent generations which can lead to premature convergence on a suboptimal solution. An adjustment which addresses this issue is to sort the individuals and assign proportions of the wheel according to their ranks in a rank based scheme. There is also a chance with roulette wheel selection, particularly in small populations, that the fittest individuals are over or under represented. Stochastic universal sampling addresses this issue by using a number of equally spaced arms on the arm wheel. The wheel is spun once only and each individual is added to the mating pool once for every arm that lands on its corresponding slot.

Variation

Once the mating pool is filled, the parents are used to create more individuals. At the simplest level the parents are paired and one or two new individuals are made by taking parts from each parent, a process known as *recombination.* Each new individual may then be altered in some manner, in a process known as *mutation*. Not all EAs use both recombination and mutation; some use one but not the other. The *operator*s, which define the ways in which the parents are combined and the offspring mutated, depend primarily on the algorithm used and the problem being solved. As no modification is required to the operators to make them suitable for use in an IEA, no details of operators are given here.

Survivor selection

If μ offspring are formed in each generation then a corresponding number of individuals need to be removed from the population in order to maintain a steady population size. The process of choosing which individuals survive is called survivor selection. The simplest approach, known as *elitism*, is to deterministically remove the least fit μ individuals from the new population (parents and offspring combined). The operators used for parent selection: tournament selection, roulette wheel selection, and stochastic universal sampling, can also be used. Age can also be integrated into the survivor selection process so that the oldest individuals are automatically eliminated each generation. Setting an age limit on individuals helps to prevent searches becoming stuck at local optima. Elitism may also be added to stochastic survivor selection methods (those methods in which the fittest individuals are not guaranteed to survive into the next generation) so that the fittest individuals from the previous generation are guaranteed to be carried forward into the next generation along with the stochastically selected individuals.

Initialisation

An EA has to have some individuals with which to start the evolutionary process. The most popular means of determining the initial population is to create individuals at random. This approach has the advantage that there is no chance that the EA neglects parts of the search space due to preconceived ideas concerning the nature of the solution. Conversely, if some prior knowledge exists, or there is some information or heuristic which enables the initial population to have a better average fitness than a random population would have, then the search can be given a ‘head start’. However, as EAs generally make rapid progress in the first few generations the effort required to find a fitter starting position is seldom rewarded.

Termination

An EA has to stop at some point. There are a number of criteria that may be used for termination of a search using an EA. The most obvious terminating condition is that the EA should stop when it obtains a solution that exceeds a particular fitness value. More deterministic finishing criteria could be used: the EA could terminate after a certain number of generations, after a set amount of time has passed, or a particular number of fitness evaluations have been performed. The EA may terminate when the same solution is the fittest for a certain number of generations. The EA may terminate when the diversity of the population drops below a certain threshold. In practice, a combination of these criteria are used to ensure that the EA does not terminate whilst better solutions can be found easily or continue searching when the EA can find no better solutions.

2.1.2 For what problems are EAs suitable?

EAs are robust and can be used to solve many problems of various forms provided that the particular EA and associated operators chosen and the values of any parameters that need to be set are appropriate to the problem. There are problems for which it is better not to employ an EA but to use an alternative approach instead. Mitchell [81] identifies five questions which should be considered before using an EA to solve a problem.

- Is the measure of quality used to evaluate solutions, the fitness function, noisy?
- Is the fitness function unimodal in the search space?
- Is there an approach tailored to the problem available?
- Is the search space large?
- Is a suboptimal solution satisfactory?

A noisy fitness function is one in which repeated evaluation of the fitness of an individual gives multiple fitness values. EAs are generally more robust to noisy fitness functions than deterministic optimisation methods [71, 134]. If the fitness function is noisy then the use of an EA becomes a valid approach to solving a problem, even if the other considerations listed here suggest the use of an alternative approach.

If the fitness function is known to be unimodal, that is, if it has no optima other than the global optimum then an appropriate hill climbing method is guaranteed to find the optimum solution unless the fitness function is noisy.

If the problem is known to not be unimodal but still well understood such that some knowledge exists about the shape of the fitness landscape (the relationship between the genotypes and the fitness values) then a tailored approach will provide a more satisfactory solution.

If the search space is not large then a search which tries every possible solution can be used. Such a search is guaranteed to return the best solution.

EAs are not guaranteed to find globally optimal solutions. An EA's chances of finding a globally optimal solution are increased if the EA is combined with other search methods. An example of this approach is using an EA to explore a search space and then using the best solution found by the EA as the starting point for a hill climbing method.

2.2 Interactive Evolutionary Algorithms

IEAs differ from EAs in one major aspect: human evaluation replaces the fitness function. This human evaluation is not necessarily of the phenotypes directly, but may be of the results of applying the phenotype to the problem to be solved. For example, Chapter 3 deals with the development of filters that remove salt and pepper noise from photographs. The genotypes are vectors of ten real numbers: nine of which are the filter weights and one is the step size mutation parameter. The phenotypes are the weighted vector directional filters, but it is the filtered photographs that are subjected to human evaluation. The term *stimulus* is appropriate for what the users actually evaluate as stimuli need not be still images; for example, IEAs have also been used to develop music [65] and animations [30]. However, this thesis deals only with images (which includes the colour panels used in the experiments of Chapters 4 and 5), and thus the comparison between IEAs and EAs in Section 2.2.3 refers to images only.

2.2.1 Aspects of an IEA

The basic structure of an IEA is similar but not identical to that of an EA and is given in Algorithm 2.

```
INITIALISE population with random individuals;
EVALUATE all individuals using human evaluation;
while termination condition is not satisfied do
    SELECT parents;
    SELECT individuals to survive into the next generation;
    RECOMBINE pairs of parents;
    MUTATE the resulting offspring;
    EVALUATE new individuals;
end
```

Algorithm 2: Basic structure of an interactive evolutionary algorithm

The effects of using human evaluation instead of a fitness function on the various aspects of an EA are summarised below. Much of the following discussion is an expansion on that which precedes the survey of the field of IEAs by Takagi [124]. Some of these points are also touched upon by Lewis [72].

Representation

The representation of the genotypes remains dependent on what is appropriate for the problem and the algorithm to be used; no special adjustments are needed just because human evaluation is used instead of a fitness function.

Fitness

The evaluation process is the principal difference between EAs and IEAs. Fatigue (see Section 2.2.2) is an important consideration when choosing the evaluation process to be used.The way in which fitness values are to be used in parent and survivor selection depends upon how they are assigned during the evaluation process. In the broadest of terms there are two ways of evaluating a population: rating and ranking. In a rating system each image is given a rating on some scale. The advantage of a rating system is that it affords the user the ability to provide information on how much better some images are compared with others. This information gives an IEA's designer more options when it comes to parent selection and survivor selection. Providing a rating for each individual in a population is time consuming and thought must also be given to the scale used. If there are too few graduations then some of the information that could have been provided is lost. For example, two images that would be rated as equal on a five point scale may have different ratings on a ten point scale. If there are too many graduations then the user spends time making fine distinctions which are probably of no use. For example, deciding whether a image warrants a 62 or a 63 on a 100 point scale. Yoon and Kim [145] compared a continuous scale (in the form of a slider), a five point scale, and a two point scale (good/bad) in an IEA used to enhance photographs and found that the two point scale was preferred by the participants of the experiment.This result would indicate that sacrificing the precision afforded by using a number of graduations in favour of simplicity is justified. The other problem with rating systems is that if there is no clear objective means for rating images then an image's rating can change relative to the other images in the population. A user using an algorithm which generates images may give their favoured image a high rating because it is good relative to the other images. A few generations later, after the population as a whole has improved, the same image may receive a poor rating. In this situation the user has effectively ranked the images and used the rankings to assign ratings. Ranking can require less effort than rating as it is only necessary to decide which images are better and not the degree to which they are better. The drawback of ranking compared to rating is that the user is not afforded the ability to, for example, rate the two best images as equally good and the remainder as equally poor. Also, ranking every member of even a moderate sized population requires considerable effort. Partial rating and ranking can be used to make the process easier. For example, rating only the best three in a population (effectively assigning all other members of the population the lowest possible rating) or ranking only the best three (effectively ranking all other members of the population as equal last). It is possible to combine rating and ranking. For example, Frowd [34] developed an evaluation method in which the user selects a single best image (a ranking process) and other images they consider to be good (a rating process with two levels). The lack of guaranteed consistency in ratings due to users being inclined (or required) to rate images based on other images in the population means that the survivors from the previous generation would need to be re-evaluated alongside the offspring. This has a significant effect on the survivor selection step of the interactive evolutionary process and is the reason why in an IEA it is generally performed at the same time as parent selection.

Population

The population size is significantly affected by using human evaluation. If ranking is used as the evaluation method then it is better if the entire population is displayed at once in order to allow convenient comparison between the individuals. If rating is used then evaluating a large population takes a long time as it would be necessary for the user to flip between screens of images. It would also be necessary that a user be able to evaluate images without reference to all of the other images in the population. Another consideration is that if each image takes, for example, a third of a second to create then a population size of nine means that each new generation takes about three seconds to create. To generate a population of 18 individuals would require about six seconds. The longer the user has to wait for each generation to be generated, the more bored they will become and the less likely they are to continue the process to a satisfactory solution.

Parent selection

There is no inherent reason why the parent selection method chosen needs to be dictated by the use of an IEA. However, if the designer has chosen to implement a simple evaluation method in order to minimise fatigue (see Section 2.2.2) then this does affect the choice of parent selection method. A simple rating system of three graduations would likely cause many ties in a $k = 2$ tournament selection system and for larger values of k a single fittest member of the population could dominate the mating pool. Roulette wheel selection may have a problem with a small population size even if a rank based method is used; if few spins of the wheel are being used to fill the mating pool then there is an increased likelihood that some members of the population are under or over represented in the mating pool.

Variation

As with EAs, the ways in which the parents are combined and the offspring mutated depend primarily on the algorithm used and the problem being solved. It has been established that EAs with small population sizes benefit from higher mutation rates than those with larger populations [117, 52]. IEAs typically have small population sizes due to the restrictions imposed by human evaluation. For this reason, IEAs generally depend on mutation to a greater extent than EAs.

Survivor selection

Survivor selection in an IEA is different to that of a EA. Algorithm 2 shows survivor selection as being just after the parent selection stage. Survivor selection in an IEA typically happens in parallel with parent selection. This is because unlike in an EA, in which the fitness values of the previous generation can be reused, it is generally the case in an IEA that the images from the previous generation need to be assessed alongside the new images as the user's opinion of them may have changed. A case can be made for placing survivor selection in the same place in an IEA as in a EA if users are capable of evaluating the quality of images without comparison to other images, that is, users are capable of evaluating the quality of images objectively, but it is unlikely that the users will be capable of doing so. Given that the population μ is likely to already be limited to how many images can fit on the screen, $\mu + \lambda$ images would not all fit on the screen at once and therefore comparison between the new images and the old would become burdensome due to not being able to view all of the images at the same time. A simple way to avoid this problem is to select the $\mu - \lambda$ fittest members of the current population to be carried through into the next generation based on their fitness relative to the current population only. Any individuals that survive from one generation to the next do so because of elitism but they survive not because they are superior to the offspring but because they are superior to the other members of the current generation. The advantage of using this form of elitism is that although it is possible for a generation as a whole to be worse than the previous generation, the best individual(s) from the previous generation will always be present.

Initialisation

Creating the initial population can be done in the same way as is prevalent in EAs — through random generation. However, because of the time each generation takes to evaluate (and perhaps generate) it is worth considering beginning a search in an area of the search space likely to lead to useful solutions. For example, EFIT-V [139], a piece of commercial software for creating facial composites, starts the search in a more likely part of the search space by having the operator ask the witness questions about details of the appearance such as the shape of the chin of the person whose face is to be recreated using the software.Image enhancement tasks may start with genotypes corresponding to processes known to provide generally satisfactory results. It may even be possible to roughly evenly spread the genotypes throughout the search space — enabling a greater exploration of the search space than a random initial population would allow.

Termination

As is the case with an EA, an IEA requires some criteria according to which the IEA should be stopped. The most obvioustermination point for an IEA is simple: the user stops when they choose to. The reasons for a user choosing to stop are similar to those listed for EAs. The user may choose to stop when they are satisfied with one of the images. The user may decide that five minutes is all the time they are willing to spend on developing an image or process. The user may observe that the same image has been the best one for the previous few generations and has decided to settle for that image. The user may decide that the average quality of the images is getting worse and conclude that it is not worth continuing. In reality a user is likely to stop for a combination of these reasons. At the beginning it is likely that the user will not stop for anything other than a good solution. As time passes fatigue becomes a factor and the user becomes more likely to stop due to one of the other reasons.

2.2.2 Fatigue

The most significant limitation imposed by the use of human evaluation is fatigue. The term fatigue is used to cover both mental exhaustion (the tiredness that can be brought on by the cognitive demands of the IEA task and inhibits cognitive functioning)and boredom (the lack of interest in performing the IEA task). They are slightly different factors and it is possible for a user to be affected by one but not the other. Assuming that the EA the IEA is based on is appropriate for the problem being solved, the degree of both boredom and exhaustion experienced depends on the user interface and the user's enthusiasm for the task to be undertaken. An interface in which the user has to repeatedly select the best one of two images will more likely bore the user than exhaust them. An interface in which the user has to rate every member of a population of nine individuals on a scale of 0–100 will exhaust the user more than bore them. The task undertaken also has a bearing on whether it is more exhausting or boring. A witness of a crime using facial composite software such as EFIT-V [139] or EvoFIT [36] which is based on the same principles as the software used for the experiments of Chapter 7to recreate the face of a suspect is more likely to become exhausted than bored as they are likely to be motivated to see the process through to a satisfactory conclusion. A user trying to develop an aesthetically pleasing pattern using generative art software is more likely to become bored than exhausted because the task is not particularly demanding.

The most elementary methods of minimising user fatigue are to present an appropriate number of individuals for evaluation and to use a suitable user evaluation system. There are other approaches which have been used in an attempt to gather more information from the user for the same amount of user effort, though not all of these are appropriate to image enhancement or creation. Gong et al. [45] used a three point scale: ‘promote’, ‘neutral’, and ‘demote’. The user rated all eight individuals in the population. The time it took for a user to promote or demote an individual was also used. The reason behind this approach is that if a user promotes an individual quickly then that individual is likely to be particularly good. Conversely, if a user demotes an individual quickly then that individual is particularly poor. Kamalian et al. [62] developing an IEA to optimise a micromachine resonating mass used an expert in the field of micromachine resonating masses to develop a virtual user which would pre-rate the individuals for the user. When the individuals were displayed for evaluation they were already rated, the idea being that the user would only need to adjust the ratings assigned to a minority of the population. Pallez et al. [92] used an eye tracker as the user’s only method of evaluating the individuals. The theory is that a user will pay more attention to those images that they find aesthetically pleasing and thus the algorithm uses user attention to evaluate the individuals in the population.

User evaluations of some of the individuals in the population can be used to infer fitness values of the rest of the population. This approach enables the use of population sizes comparable to those used in EAs. Gong et al. [46] used clustering to group the population into clusters in the genotype space and have the users evaluate the phenotypes corresponding to the centres of the clusters. An individual was assigned a fitness interval (as opposed to a single fitness value) according to their genotype’s proximity to the centre of its cluster and the fitness value of the genotype at the centre of the cluster. It is also possible to use interpolation methods to assign fitness values to individuals not rated by a user. Quiroz et al. [102] used a human evaluation method whereby the user would choose the best and worst individuals from a subset of the total population. The remainder of the population were assigned human ratings based on interpolation between the best and worst individuals in the genotype space.

Hybrid EAs that use both human evaluation and an objective fitness function have also been explored. Quiroz et al. [102] used a hybrid EA to develop user interfaces. The user would select a best and worst as stated above but an objective fitness function was also used which evaluated phenotypes based on their adherence to good design practices. Gong et al. [44] developed a means of building a model of the current user by using not only evaluations made by the user but evaluations made by previous users who generally agreed with the current user. The model was then used to perform the fitness evaluations until the algorithm dictated that the model needed updating.

2.2.3 For what problems are IEAs suitable?

The five questions of Section 2.1.2 need to be revisited in the light of the limitations imposed by using human evaluation as the fitness function. The questions are restated here for convenience:

- Is the measure of quality used to evaluate solutions, the fitness function (which in an IEA is human evaluation), noisy?
- Is the fitness function (human evaluation) unimodal in the search space?
- Is there an approach tailored to the problem available?
- Is the search space large?
- Is a suboptimal solution satisfactory?

In IEAs human evaluation, which can be thought of as a *subjective fitness function* is certainly noisy. If the user does not have a specific target in mind then their preferences can change during the course of the search thus causing individuals previously considered desirable to no longer be so. Even if the user does have a specific target in mind fatigue or exposure to new images can cause them to change their minds about the fitness of previously evaluated images.

Whether a subjective fitness function is unimodal or not depends on two factors. The first is user intent. For example, consider a system that generates facial composites such as EvoFIT or EFIT-V. If the user was trying to recreate a face of a particular person they have pictured in their mind then the subjective fitness function would be unimodal — faces that bear a closer resemblance to the one they have in their mind, and are therefore fitter, would have genotypes closer to the genotype of the face they are trying to recreate. If they were trying to create a face that is 'attractive' then there are likely be regions of the search space consisting of less attractive faces separating attractive faces. In this case the task is not unimodal despite the search space being identical. A system that generates art may have complicated mappings between the values in the genotype and the phenotypes that lead to similar phenotypes having dissimilar genotypes and thus even a search with a particular goal may have a non-unimodal subjective fitness function. Assuming that the subjective fitness function is unimodal, then why not use a hill climbing method? Well, in general a user cannot look at an image and specify the direction in which the variables in the genotype need to be altered in order to improve the phenotype, for if they could then the use of an IEA is a poor way of searching for a solution and sliders or number boxes should be used instead.

In the context of EAs, alternative approaches typically refer to other problem solving or optimisation methods such as neural networks. In this thesis the term 'alternative approaches' is taken to mean other means for users to supply input values to image enhancement and creation processes. Examples of alternative approaches would include colour swatches for the colour matching task of Chapters 4 and 5 and the slider based interface introduced for comparison purposes in Chapter 6.Typically, the alternative approaches follow the currently favoured paradigm of direct input of values using sliders and numeric text boxes. Direct input should be favoured if the problem is easily separable, that is, if the desired result can be obtained by setting the values for each input one at a time without the need to adjust an input value once it has been set.

Given the limitations concerning the number of images that can be presented to the user at once and the number of times a user is likely to be willing to evaluate a screen full of images it can be deduced that the number of possible solutions to a problem required for it to be considered as having a large search space is likely to consist of relatively few individuals in comparison to problems which can be solved using objective fitness functions. The number of distinct individuals required before a search space can be considered large depends upon how difficult the images are to evaluate, how many can be presented at once, and the patience of the user. If ten images are presented at once and a user is capable of reliably picking the best one of any ten images and they can do this about 100 times then the search space can consist of a maximum of a little under 1000 individuals. This is a rather optimistic estimate but it serves as an approximate upper limit to the number of individuals over which a complete search can be conducted.

The noisy nature of human evaluation and the limitations of human perception mean that a search using an IEA is a search for a solution in an optimal region as opposed to a search for a single optimum point. As with EAs, a different search method can be used to find the best solution near to the current best solution found by the IEA. There is, however, no guarantee that this will be a global optimum.

In the light of this discussion, it can be assumed that for all image processing and creation problems that the subjective fitness function will be noisy to some degree. It can also be assumed that due to limits on the human ability to distinguish between images that a suboptimal solution has to be acceptable for any approach to the problem. Due to the noise in the subjective fitness function, unimodality may not render the use of an IEA inappropriate but it may affect some of the choices made when implementing an IEA as some design options may not be suitable for non-unimodal subjective fitness functions. Combining Mitchell's suitability criteria [81] with the unimodality question provides a list of three suitability questions which should be considered before employing an IEA:

- Is the subjective fitness function, the fitness function that is assumed to exist in the user's mind),unimodal in the search space?
- What other approaches to the problem are available?
- Is the search space large?

It should be noted that the first question is less about the suitability of using an IEA and more about the design options that are likely to be suitable.

2.2.4 Which EAs are most suitable for adaptation to IEAs?

The limitations imposed on the IEAs by the use of human evaluation have some effect on the suitability of the various EAs and other nature inspired metaheuristic algorithms for conversion to use with human evaluation. A list of a few of the more well known metaheuristic algorithms is presented along with any aspects of each algorithm that may render it less suitable than other algorithms for use with human evaluation. For the purposes of this discussion suitability is not determined by consideration of the properties of the algorithm itself, for that will depend on the problem to be solved, only the demands placed on the user during the fitness evaluation process. This list briefly discusses the demands that would be made of the user to provide the information needed for parent and survivor selection for each of the algorithms.

Genetic algorithms

Genetic algorithms (GAs) were introduced by Holland in 1973 [54] (as cited in [29]) and are by far the most commonly used EA in image processing. This is likely to be because they are the most well known form of EA but also they are the most versatile due to representation options a GA affords; the genotype representation in a GA is a vector of bit-strings, integers, or real numbers. A GA's representation makes it suitable for use in image processing problems where the form of the solution is fixed but the input values need to be optimised. The limitations detailed in Section 2.2.1 do not lead to any particular problems beyond those already discussed with regards to population size and evaluation effort.

Evolution strategies

Evolution strategies (ESs) were introduced by Rechenberg in 1965 [104] (as cited in [7]). The genotype of an ES is a vector of real numbers which also includes mutation parameters. As GAs have more representation options they can be applied to a wider variety of problems than ESs, but most image processing input optimisation problems lend themselves to real valued genotypes as the inputs to image processes are generally real numbers. As with GAs, ESs have no extra limitations imposed by human evaluation.

Genetic programming

Genetic programming (GP) was introduced by Koza in 1989 [67]. GP uses a tree of nodes for its genotypes. GP is not suitable for input value optimisation but it is suitable for problems where the goal is to find an optimal way to combine a number of operators to accomplish a particular task. For example, Poli and Cagnoli [100] used GP to combine two images using operators such as '+', '−', and 'min' to highlight differences between the images. GP has no extra limitations other than those already discussed when using human evaluation.

Evolutionary programming

Evolutionary programming (EP) was first conceived by Fogel in the mid 1960's [32] (republished in [31]). EP has changed since it was first conceived; early EP used state machines to represent genotypes. Modern EP uses a vector of real numbers which include the mutation step size parameter in the same way as the genotypes of an

ES. The difference, however, is the way in which EP procreates. In EP each member of the population is parent to one offspring, which is a mutant of the parent. At the survivor selection stage there are μ parents plus μ offspring. Half of the individuals are chosen to survive into the next generation. This leads to the problem of the parents needing to be compared to the offspring. In this case the images need to be compared over multiple screens, or the population size needs to be half of the number of images that can fit on the screen, or the task needs to be such that the user's opinion of the images does not change over the generations so that the fitness values previously assigned to the parents can be used to compare the parents to the offspring. Therefore an ES or a GA would be a preferable option to EP.

Differential evolution

Differential evolution (DE) was introduced by Storn and Price in 1997 [119]. DE is not an EA but like an EA it is a population based metaheuristic problem solving algorithm and as such its appropriateness for use as an alternative to other methods of setting input values can be assessed with the IEA suitability questions above. DE uses a vector of real numbers for its genotype and so can be considered as an alternative to an ES or GA in terms of having a representation appropriate to the problem. At the variation stage, DE takes each individual in the population and applies recombination with a 'temporary' individual constructed from three other individuals in the population which are randomly chosen except for the constraint that they must be distinct. At the survivor selection stage, each parent is compared with its offspring and the fittest survives. If human evaluation is to be used in DE the survivor selection must occur at the same point as it does on a EA as parents need to be compared to offspring. This comparison can be in the form of a pairwise comparison between every parent/offspring pair or, if the task is such that the user's opinion of the images does not change over the generations, by using rating system to apply fitness values to the phenotypes. If the pairwise approach is used then μ pairwise comparisons must be performed each generation, which is a lengthier process than, for example, a three point rating scale. From this point of view, a GA or an ES is a preferable approach to DE.

Particle swarm optimisation

Particle swarm optimisation (PSO) was introduced by Eberhart and Kennedy in 1995 [28]. PSO is another non-EA population-based metaheuristic problem-solving algorithm. The solutions in PSO are represented as vectors of real numbers and so can be considered as an alternative to using an ES or a real valued GA. In PSO the representation of the solutions (what would be called genotypes in an EA) in the search space are referred to as *particles*. PSO works by having a fixed number of particles in the search space which are then moved around on a generational basis, that is, all of the particles are moved at once. Every generation, each particle's motion in the search space is defined by three aspects: a random motion, movement toward the position of that particle's best fitness up until that point, and motion toward the position of the best fitness found amongst all of the particles up until that point. PSO requires each particle's previous fittest position and the globally fittest position to be recorded. This poses a problem when using human evaluation the subjective fitness value of a point in the search space is liable to change during the search because of the changing text in which it is evaluated. Mádar et al. [79] addressed this problem by having the user compare each particle's most recent solution to the particle's fittest solution until to that point to determine whether the position of the fittest solution needed to be updated. This was performed by the user in a series of pairwise comparisons. The user would then select a global fittest output from the fittest output of each particle. If the population size is n then it can be seen that this requires the user to perform $n$pairwise comparisons each generation followed by selecting their preferred output from μ outputs, a lengthier process than, for example, than evaluating an entire population on a three point rating scale. From the point of view of user interaction a GA or an ES is a preferable approach over PSO. A variation of PSO called accelerated PSO does not use the particles' fittest positions in the particle motion part of the algorithm and thus only requires that the current fittest particle be chosen from the population. It can be seen that with regards to human evaluation accelerated PSO may be a viable alternative to an ES or DE.

Firefly algorithm

Firefly algorithms (FAs) were introduced by Yang in 2009 [144]. An FA has similarities to accelerated PSO in that the particles, now called *fireflies* move around the search space searching for the fittest position. The motion of a firefly is governed by two aspects: random motion and motion toward any fireflies that appear to be brighter. The perceived brightness of a firefly depends on the fitness value of the firefly and its distance from the viewing firefly. If there are two fireflies of equal fitness the one farther away will appear to be dimmer than the nearer one. In order to be effective, an FA requires that fireflies are at least ranked for fitness. From a human evaluation point of view, the entire population needs to be ranked every generation. This makes an FA a less appealing choice than PSO, an ES, or an EA.

2.3 Statistical methods used

A major weakness in much of the work with regards to IEAs, and to EAs too, is the lack of statistical analysis of the data collected. To ensure that the conclusions drawn from the experiments presented in this thesis are robust the data were subjected to statistical analysis.

The most appropriate statistical method to employ to analyse data depends on the form of the data collected. A variable is categorised as belonging to one of four types, in ascending order of the stringency the of requirements to belong to each category these categories are: categorical, ordinal, interval, and ratio. A categorical variable requires no form of ordering, it is only necessary that it is possible to state that for two values x and y that $x = y$ or $x \neq y$. An example of a categorical variable is blood group. An ordinal variable has an ordering such that it is possible to state $x > y$, $x = y$, or $x < y$. An example of an ordinal variable is the place of a runner at the end of a race. Interval variables have the property that the difference between values x and y is equal to the difference between values $x + z$ and $y + z$. An example of an interval variable is temperature of an object measured in degrees Celsius. A ratio variable is measured on a scale such that a value of $2x$ on the scale has twice the magnitude of a value of x. An example of a ratio variable is temperature of an object measured in Kelvin.

It is possible to recast data in one form to another whose qualifying criteria are less stringent. Data in ratio form can be transformed to interval form, which can be transformed to ordinal form, which can be transformed to categorical form. Statistical tests for data in ratio or interval form are more likely to find statistical significance than tests for the same data transformed to ordinal or categorical form. It can be seen that it is seldom desirable to make such transformations but it is however sometimes necessary. If a comparison needs to be made between ordinal data and interval data the interval data need to be transformed to, or treated as, ordinal data. It may also be the case that interval or ratio data do not meet the requirements for analysis using particular statistical tests for interval or ratio data and so need to be transformed to, or treated as, ordinal data so that statistical analysis can be performed.

A summary of the tests used, what they are used for, and (where appropriate) why they were chosen over other tests is presented in this section.

2.3.1 The binomial test

The binomial test is a test of significance for categorical data which can hold one of two values. In Chapter 6, a binomial test is used to compare participant preferences when they have two user interfaces to compare on a number of aspects. If there is no or little difference between the preference counts for each of the interfaces then it is concluded that there is no difference between the interfaces on that aspect. If one interface is preferred far more often than the other then it is concluded that that interface is generally preferable. More details about binomial tests can be found in [57].

2.3.2 The Friedman test

The Friedman test is a test to establish if the difference between treatments is significant when they are compared using a variable on an ordinal scale. The Friedman test is best used on experiments with a one-way design, that is, there is only a single independent variable; if used to analyse multivariate data it can fail to detect interaction effects. Of particular note is that the Friedman test ranks all of the data from each participant such that each participant will have a treatment ranked '1', a treatment ranked '2' and so on (unless there are ties). This is in contrast to the Wilcoxon matched-pairs test in which the data is ranked over all of the participants. The Friedman test can also account for multiple evaluations of each treatment from each participant. The Friedman test is therefore suitable for those sets of data in which participants were asked to rank images or other stimuli in order of preference.

The preferred method for comparing treatments according to interval or ratio data gathered in an experiment with a one-way design is to use analysis of variance (ANOVA [116]). There are some criteria that data needs to satisfy beyond being interval or ratio for ANOVA to be appropriate, notably that the data for each treatment have a Gaussian distribution and that the variances of the data for each treatment should be the same. If either of these criteria is not met then a statistical test for ordinal data should be used instead. The ratio data gathered in the experiments in this thesis (such as time taken to complete a run of the experiment) fails to satisfy at least one of these criteria and so to analyse the data from experiments with a one-way design the data are transformed to sets of rankings and the Friedman test is used instead of one-way ANOVA.

If a statistically significant difference between treatments is found using Friedman's test, a suitable post hoc test is required to identify which treatments are significantly different to which others. The post hoc test used for the Friedman test in this thesis is Fisher's least significant difference (LSD) test for ranks. More details of the Friedman test and Fisher's LSD post hoc test for ranks can be found in [19].

2.3.3 Aligned rank transform with multivariate analysis of variance

An aligned rank transform (ART) [143] is a method which allows multivariate ordinal data, or interval data which fails to meet the ANOVA criteria, to be analysed using multivariate ANOVA (MANOVA). In essence, an ART removes all variation in a measured variable due to all effects except one; the data are said to be aligned to this effect. The transformed variable is then converted to ranks and MANOVA is performed on the ranked data. Only the statistical significance of the effect the data was aligned to is recorded. The process is repeated for all main and interaction effects. No post hoc tests were necessary for the ART MANOVA as all multivariate experimental designs were 2×2 or $2 \times 2 \times 2$ and so the direction of any effects could be discerned from the mean ranks of the data.

2.3.4 The chi-square test for independence

The chi-square, or χ^2, test for independence, abbreviated here to chi-square test, is a test of association between two or more variables in which the data are categorical. A theoretical distribution of frequencies for each combination of variables is established for the case in which there is no interaction between the variables. The further the actual frequencies differ from this theoretical distribution the more likely that it is that there is a relationship between the variables. More details about the chi-square test can be found in [8].

2.3.5 Spearman's correlation coefficient

The Spearman coefficient of correlation, or Spearman's ρ, is a measure of correlation between two ranked variables. It is used to investigate the relationship between the measured variables in the experiments. Pearson's coefficient of correlation, or Pearson's r is the equivalent method to Spearman's ρ for ratio and interval data. Pearson's r requires that both variables to be compared are normally distributed, which the data gathered during the experiments in this thesis tends not to be. Consequently, all correlations to investigate the relationships between the measured variables are calculated using Spearman's ρ with interval and ratio data being transformed to ranked data if necessary.

2.3.6 Kendall's coefficient of concordance

Kendall's coefficient of concordance, or Kendall's W, is a measure of correlation used to measure the level of agreement between a number of judges who each assign ranks to a number of treatments such that each judge ranks the n treatments from 1 to n. A value of $W = 0$ signifies that there is total discord amongst the judges, a value of $W = 1$ indicates that the judges are in total agreement. In this thesis, Kendall's W is applied to the ranks awarded by the participants to provide extra information about the generality of the results.

2.4 Summary

There are a number of considerations that need to be borne in mind when implementing an IEA to solve a problem, the most important being whether the use of an IEA is appropriate for the problem — something which is not always apparent. Mitchell's [81] considerations to test the appropriateness of using an EA to solve a problem were adapted for testing the appropriateness of using an IEA. As with EAs, the problem to be solved usually suggests an appropriate representation which in turn aids in the selection of an appropriate algorithm. The fitness evaluation needs of the underlying algorithm need to be balanced against the capabilities of the user to ensure an efficient and effective search. An IEA is less likely to become trapped in a local optimum and can explore more of the search space if the population size is large. However, a small population is easier for a user to evaluate. An evaluation method that uses fine rating scales can provide more information to the EA which gives the algorithm developer more options on aspects like parent selection. However, a simple evaluation system is less burdensome on the user. From a user evaluation effort point of view, GAs, ESs, and accelerated PSO were identified as the most suitable algorithms for use in solving problems for which the use of an IEA is appropriate.

Statistical methods appropriate for the data gathered in the experiments of this thesis were summarised. The binomial, Friedman (with Fisher's LSD test), and ART with MANOVA tests were selected for evaluating the statistical difference between treatments in categorical, one-way ordinal, and multi-way ordinal data respectively. The chi-square test for independence, Spearman's correlation coefficient, and Kendall's coefficient of concordance were selected for determining associations between categorical, two-way ordinal, and two-way ordinal with repeated measures data respectively.

Chapter 3

Optimisation of weighted vector directional filters using an interactive evolutionary algorithm

3.1 Introduction

A common image processing task is to remove noise from, or suppress noise in, an image. One type of noise is salt and pepper noise. Salt and pepper noise gets its name from its appearance in grey-scale images — as grains of salt and pepper scattered across the image. Salt and pepper noise can be introduced to an image by data transmission errors, by flaws in analogue to digital converters when the image is captured, or by dirt between the scanner and the source if the image was digitised. Salt and pepper noise is normally treated using order statistic filters such as the median filter. The application of (scalar) order statistic filters in grey-scale image processing has been studied extensively [94]. More recent work includes the alpha trimmed mean filter [76], the fast and efficient median filter [58], and filtering based on the stationary wavelet transform [68].

The treatment of salt and pepper noise in colour images has received less attention than the grey-scale case. This is not surprising as the most popular approach to treating salt and pepper noise in colour images is to separate the red, green, and blue channels in the image, use an order statistic filter to treat each channel as a grey-scale image, and then recombine the channels.

Vector filtering is an approach to colour image filtering that uses data from all three channels at the same time. The pixel values of an image are treated as points in the sRGB colour space (more detail about colour spaces is given in Section 5.2.1). In a vector filter, the colours of the pixels around the pixel of interest are plotted as points in the sRGB colour space. The median pixel is the pixel in this set that is closest, in colour, to all of the other colours of the other pixels (see Section 3.2.1). Vector filtering requires the use of some distance metric in order to measure the distances between the pixels and thus define the pixel that is closest to the others. The original paper by Astola et al. [5] used the Euclidean distance. Trahanias and Venetsanopoulos [132] used the angle between the points as measured from the origin. Plataniotis et al. [96] used a combination of distance and angle. Plataniotis et al. [97, 99] added weighting to the distances based on local image statistics. Cree [22] provides a single framework for the various approaches and some evaluation of their effectiveness. Lukac at al. [75] provide a more detailed overview of vector filtering but with no comparison of the efficacy of the various flavours.

The particular vector filter this experiment is focused on is the *weighted vector directional filter* (WVDF). The WVDF is not an adaptive filter — the weights are fixed so that they are the same for every filtering window. The problem then is to determine what these weights should be in order to achieve the optimal result. To address this problem Lukac et al. [74] applied a genetic algorithm (GA) to optimise the weights. In [74], the GA develops a filter on a training image which is then used to denoise previously unseen images from the same imaging pipeline. A training image is created from an image which is considered to be noise-free. Noise of a known type and quantity is added to the noise-free image to create a training image. The efficacy of a filter can then be measured by filtering the training image and comparing the filtered image to the original uncontaminated image using an *image quality measure* (IQM).

Numerical IQMs are used to assess the effectiveness of various image compression and restoration techniques [141]. The most commonly used and most developed IQMs are full reference measures. Full reference IQMs compare a clean original image to one that has undergone compression or has been corrupted and then restored. The more similar the processed image is to the original, the more effective the process is judged to be.

Two assumptions are made when using the training image approach. The first assumption is that the effectiveness of the filter is to some extent independent of the image object scene. However, it is accepted that the training image chosen should be similar to the contaminated image, although what is meant by similar is open to interpretation. The second assumption is that the noise profile (both type and quantity) can be estimated reliably and can therefore be applied to the training image.

An alternative approach, the one explored in this experiment, is to use an IEA to optimise the filter weights. To assess whether the IEA approach may be appropriate to the problem, the IEA suitability questions of Section 2.2.3 are addressed:

- *Is the subjective fitness function unimodal in the search space?* No, the way in which the WVDF uses the values from the genotypes ensures that virtually every image output could originate from any one of an infinite number of sets of weights.

- *What other approaches to the problem are available?* It is possible to use sliders or number boxes to input the filter weights, but it is not possible to optimise the inputs one weight at a time as the relationship between the weights is difficult to predict. It is also possible, for images with a low amount of noise, to manually edit any individual pixels using even the most basic of graphics software packages. As has already been seen, an EA can be used to develop a filter on a atraining image [74].

- *Is the search space large?* It is very likely but it is difficult to be sure. There are nine weights but the smallest change in a filter weight that produces a noticeable difference in the filtered image depends on the values of all of the filter weights.

Consideration of the suitability questions indicates that there is sufficient justification for the use of an IEA to tackle the noise removal problem

EAs are generally quicker and more convenient to use than IEAs but require a fitness function which, in the context of perceptual image enhancement, is difficult to define mathematically. Therefore an appealing idea is to emulate perceptual image quality using an IQM. Whilst the development of IQMs and the assessment of their relative performances has been studied [6], little work has been done to compare them with human evaluations of image quality, though one notable example of such a comparison is provided by Sheikh et al. [109] who developed an extensive database of human evaluations. The database was used to test the suitability of many IQMs when considering the impact of compression artefacts on images [108]. In this experiment the opportunity was taken to compare five IQMs with human evaluation of image quality when considering the specific image processing task of denoising. If a suitable IQM can be found, and the assumptions of the training image approach hold, then the use of an EA is warranted and is indeed preferable to the use of an IEA.

3.2 Theory

3.2.1 Vector medians

The common definition of the median of a set of N scalars is that if the scalars are sorted numerically then the median is the scalar in the $(N + 1/2)$-th position (if N is odd) or the mean of the scalars in the $N/2$-th and $N/2 + 1$-th positions (if N is even). This definition cannot easily be generalised for multivariate data. The optimality property of a median [120] provides a basis for a solution. Under the optimality property, x_j is the median of a set S only if x_j minimises the mean absolute error with respect to S. Written mathematically:

$$x_{\text{median}} = \underset{x_j \in S}{\operatorname{argmin}} \sum_{I=1}^{n} |x_j - x_i| \tag{3.1}$$

where $x_1, x_2, \ldots, x_n$ are members of S. This definition can be extended to the multivariate, or vector, case [5]:

$$\mathbf{x}_{\text{median}} = \underset{\mathbf{x}_j \in S}{\operatorname{argmin}} \sum_{I=1}^{n} \|\mathbf{x}_j - \mathbf{x}_i\|_2 \tag{3.2}$$

In this particular case the median of the set S of vectors is taken to be the vector with the smallest aggregate Euclidean distance between itself and the other vectors in the set. The Euclidean distance is not the only dissimilarity measure that can be used; the definition of the vector median can be generalised [22]:

$$\mathbf{x}_{\text{median}} = \underset{\mathbf{x}_j \in S}{\operatorname{argmin}} \sum_{I=1}^{n} |d(\mathbf{x}_j, \mathbf{x}_i)| \tag{3.3}$$

where $d(\mathbf{x}_j, \mathbf{x}_i)$ is the dissimilarity between $\mathbf{x}_i$ and $\mathbf{x}_j$ according to some dissimilarity measure. Weighting can be included to place more importance on some elements in the set than others [97]:

$$\mathbf{x}_{\text{median}} = \underset{\mathbf{x}_j \in S}{\operatorname{argmin}} \sum_{I=1}^{n} w_i \left| d\left(\mathbf{x}_j, \mathbf{x}_i\right) \right| \tag{3.4}$$

where w_i are scalar weights that can be fixed or dependent on the members of S.

3.2.2 Vector filtering

In the vector filtering approach values of the pixel of interest and the surrounding pixels are represented as points in a three-dimensional space. Vectors are constructed from the origin to these points. The axes of the three-dimensional space correspond to the red, green, and blue colour channels. Each colour channel value lies in the range $[0, 255]$. A value of 0 in one of the colour channels at a particular pixel indicates that the colour is not present at that pixel. A value of 255 indicates that the maximum amount of that colour possible is present. It can be seen, for example, that, $(0, 0, 0)$ corresponds to black and $(255, 255, 255)$ corresponds to white. In the context of vector directional filtering the distance between two colours is defined as the angle between their corresponding vectors (Figure 3.1). The angle is not defined for black pixels. This problem is easily overcome by recognising that all shades of grey are represented by vectors with identical direction from the origin and so black pixels are mapped to the triplet $(255, 255, 255)$. The output of the filtering window is the pixel that has the smallest weighted sum of the angles between it and the other pixels in the window. The filtering window W is n pixels in size (in our case W is a 3×3 window so $n = 9$). The weight of the i-th position in the window is w_i. Using a vector representation, the i-th and j-th pixels in the window are $\mathbf{x}_i$ and $\mathbf{x}_j$ respectively. The output $\mathbf{x}_{\text{WVDF}}$ is the vector that satisfies

$$\mathbf{x}_{\text{WVDF}} = \underset{\mathbf{x}_j \in W}{\operatorname{argmin}} \sum_{i=1}^{n} w_i \arccos\left(\frac{\mathbf{x}_i \cdot \mathbf{x}_j}{\|\mathbf{x}_i\| \, \|\mathbf{x}_j\|}\right). \tag{3.5}$$

The basic vector directional filter (BVDF) is a special case of the WVDF in which all of the weights (w_i) in Equation 3.5 are equal:

$$\mathbf{x}_{\text{WVDF}} = \underset{\mathbf{x}_j \in W}{\operatorname{argmin}} \sum_{i=1}^{n} \arccos\left(\frac{\mathbf{x}_i \cdot \mathbf{x}_j}{\|\mathbf{x}_i\| \, \|\mathbf{x}_j\|}\right). \tag{3.6}$$

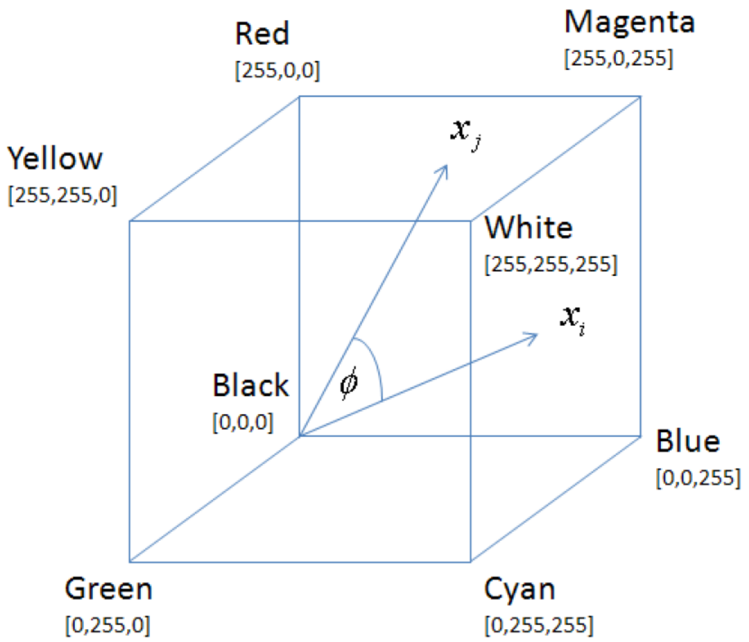


Figure 3.1: The distance between two colour vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ in the vector directional filter is the angle ϕ between them

3.2.3 Image quality measures

The IQMs that were compared to human evaluations of image quality were the mean absolute error (MAE), mean square error (MSE), mean quartic error (MQE), normalized colour index (NCD), and structural similarity index (SSIM). The MAE, MSE, and MQE have very similar mathematical forms:

$$\text{Error} = \frac{1}{c\ n} \sum_{k=1}^{c} \sum_{i=1}^{n} |x_{i,k} - o_{i,k}|^m$$

where c is the number of colour channels (three — red, green, and blue), n is the number of pixels in the image, $x_{i,k}$ and $o_{i,k}$ are the values of the of the i-th pixel on the k-th colour channel of the processed and original (pristine) images respectively. It is the value of m (the norm) that differentiates these measures. For the MAE $m = 1$, for the MSE $m = 2$, and for the MQE $m = 4$.

The NCD is defined by Plataniotis et al. [98] as

$$\text{NCD} = \frac{\sum_{i=1}^{n} \|\Delta_i\|}{\sum_{i=1}^{n} \sqrt{(o_{L^*,i})^2 + (o_{u^*,i})^2 + (o_{v^*,i})^2}} \qquad (3.7)$$

where

$$\|\Delta_i\| = \sqrt{\left(x_{L^*,i} - o_{L^*,i}\right)^2 + \left(x_{u^*,i} - o_{u^*,i}\right)^2 + \left(x_{v^*,i} - o_{v^*,i}\right)^2} \tag{3.8}$$

and n is the number of pixels in the image; $x_{L^*,i}$, $x_{u^*,i}$, and $x_{v^*,i}$ are the values of the i-th pixel of the processed image in the CIELUV colour space; and $o_{L^*,i}$, $o_{u^*,i}$, and $o_{v^*,i}$ are the values of the i-th pixel of the original image in the CIELUV colour space.

A general framework for the SSIM is detailed in Wang and Bovik [141]. The particular details of an implementation of the SSIM are selected according to the user's requirements and expertise. There is no standard adaptation of the SSIM for colour images. The approach that is used in this work is to convert the images to the Y′ channel in the Y′UV colour space in order to use the SSIM. This conversion is given by $\mathrm{Y}'_i = 0.299x_{i,\mathrm{red}} + 0.587x_{i,\mathrm{green}} + 0.114x_{i,\mathrm{blue}}$ where $x_{i,\mathrm{red}}$, $x_{i,\mathrm{green}}$, and $x_{i,\mathrm{blue}}$ are the red, green, and blue values respectively of the i-th pixel of the image. A local weighted SSIM with an 11×11 pixel sliding window is used. The weights are obtained from a discrete 2-D Gaussian distribution with a standard deviation of 1.5 and peak (mean) located at the centre of the window. The SSIM for a local window is given by

$$\mathrm{SSIM}\,(x, y) = \frac{\left(2\mu_x\mu_y + C_1\right)\left(2\sigma_{xy} + C_2\right)}{\left(\mu_x^2 + \mu_y^2 + C_1\right)\left(\sigma_x^2 + \sigma_y^2 + C_2\right)} \tag{3.9}$$

where μ_x and μ_y are the weighted mean pixel values of the original and processed images, σ_x^2 and σ_y^2 are the weighted variances of the original and processed images, and σ_{xy} is the weighted covariance between the two images. Constants C_1 and C_2 are included to prevent instability when $\mu_x^2 + \mu_y^2$ or $\sigma_x^2 + \sigma_y^2$ is close to zero. $C_1 = (0.01 \times 255)^2$ and $C_2 = (0.03 \times 255)^2$.

3.2.4 The select, multiply, mutate method

The select, multiply, and mutate (SMM) method is the name given by Gibson et al. [41] to a combination of particular parent selection, variation, and survivor selection operators designed to minimise user fatigue. The SMM method is not an IEA in its own right; it still requires the choice of an underlying IEA when it is implemented. When interacting with an IEA that implements the SMM method the user selects a single preferred member of the population, which is multiplied enough times so as to fill the population, with all but one individuals being mutated. It can be seen that an IEA that implements the SMM method, an SMM-IEA, is elitist as the parent is carried forward into the following generation. It can also been seen that the crossover operator is not used and that there is no separate parent and survivor selection stages. The encoding of the problem and the details of the mutation operator are defined by the problem itself. With regards to the EAs summarised in Section 2.2.3 it can be seen that PSO, DE, EP, and FAs are not suitable for use with the SMM method because of the need to at least rank the members of the population or because of the need for separate parent and survivor selection stages.

In this experiment, the SMM method is implemented using an IES to form what shall be referred to as the *SMM-IES*. In the SMM-IES each filter is represented as a chromosome consisting of ten genes. Nine of the genes are the filter weights $w_1, \ldots, w_9$, are real coded, and have values in the range $0 \leq w_i \leq 1$. Values of $w_i < 0$ could lead to some of the pixel positions in the filtering window providing a negative contribution to the aggregated distances. Not putting a suitable upper limit on the values of the w_i could lead to one or more pixel positions dominating the filter and the IEA being unable to reduce the weights of those positions so as to enable the search to find more desirable filters. The tenth gene is the mutation step size, σ, and is subject to the condition $\sigma \geq 0.075$. The minimum value of σ was set to ensure that the IEA could not stagnate, that some mutation would always occur when a new generation of filter weights was created.

An uncorrelated mutation with one step size was used for the mutation component of the algorithm. During the mutation stage, the step size gene was mutated by

$$\sigma' = \sigma \cdot e^{\tau \cdot N(0,1)} \tag{3.10}$$

where σ' and σ are the new and old step sizes respectively, $N(0,1)$ is a number taken at random from the standard normal distribution, and τ is a constant equal to 0.5. As the SMM method is working above an ES, the step size parameter is self adapting; it is evolved alongside the filter weights. The reasoning is that appropriate step sizes are more likely to generate desirable filter weights. After the step size parameter has been mutated, the filter weights are then mutated by what shall be be referred to in this thesis as *Gaussian addition*:

$$w_i' = w_i + \sigma' \cdot N(0,1) \tag{3.11}$$

where w_i' and w_i are the new and old weights respectively at position i in the window.

The mutation component of the algorithm is described in more detail in Eiben and Smith [29].

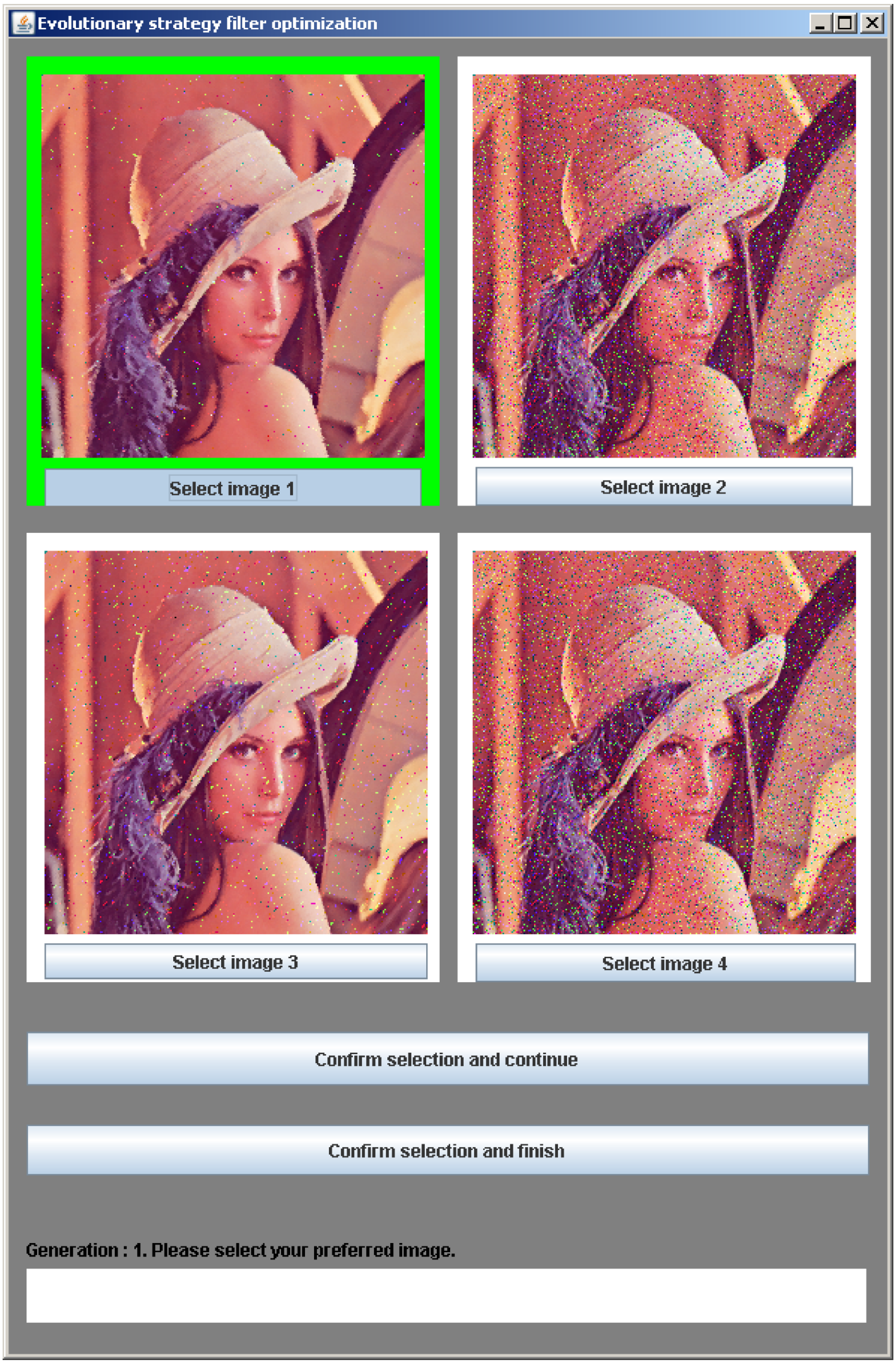
3.3 Method

3.3.1 The user interface

The user interface for the algorithm consisted of four image panels, each with a selection button underneath and two other buttons: 'Confirm selection and continue' and 'Confirm selection and finish'. The WVDF filters are computationally expensive; in the Java implementation used it takes a little less than 2 seconds to filter a 256×256 pixel image. For a population size of n, $n-1$ images need to be filtered each generation (one image is carried forward from the previous generation). It is burdensome to have to wait more than a few seconds for each new generation of images to be created. A wait of around five seconds was considered acceptable, hence a population size of four was chosen. At each iteration the participant was required to make a visual inspection of four filtered images displayed in the panel and select the one they judged to be fittest in the sense of image quality. After choosing a preferred image, the participant would select the image they thought was best by clicking the 'Select image' button underneath it. If the participant wished to continue developing their filter they would press the 'Confirm selection and continue' button. If the participant was satisfied or decided that no further improvement was forthcoming they would press the 'Confirm selection and finish' button. A screenshot of the interface is provided in Figure 3.2.

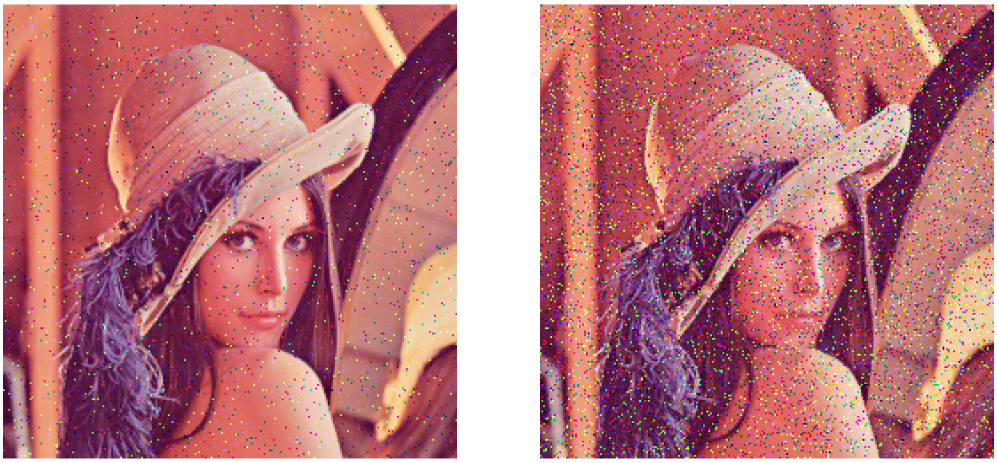
3.3.2 Test set-up

Thirty participants were used in this experiment. A large minority of the participants were postgraduate students in the School of Physical Sciences at the University of Kent. The others were a combination of undergraduate students and staff from the School of Physical Sciences and people not affiliated with the university in any way. The simplicity of the user interface meant that no particular skills were required of the participants other that they were familiar with basic computer use. At the very start of the experiment each participant was read a script. The script told the participants their task was to "improve the appearance of an image that has been contaminated by noise." The script also explained how to use the interface


Evolutionary strategy filter optimization
Select image 1
Select image 2
Select image 3
Select image 4
Confirm selection and continue
Confirm selection and finish

Generation : 1. Please select your preferred image.

Figure 3.2: Screenshot of the user interface for the development of the filters in the noise removal experiment



(a) Image I1: 2% noise

(b) Image I2: 8% noise

Figure 3.3: The noisy images used for developing WVDFs

and that the experimenter wanted to know the reasons for their decisions during the development and evaluation of their filters. Using the script ensured that each participant was given exactly the same information. The popular Lena test photograph scaled to a size of 256 × 256 pixels was the image used in this work. Two contaminated versions of the Lena photograph, which are referred to as I1 and I2, were created. Salt and pepper noise was applied to each of the red, green, and blue channels of each image. The probability of a particular pixel being contaminated on any channel was 2% for I1 and 8% for I2. Each participant was given the task of developing two filters to remove noise from images using the SMM-IES. Half of the participants developed a filter on I1 first and the other half developed a filter on I2 first. This was done to eliminate the possibility of systematic bias due to user fatigue or practice effects.

The Lena photograph and 2% salt and pepper noise were chosen because they were used in the development of the W2 filter [74] and therefore allowed a direct comparison between the SMM-IES optimised filters and the previously studied GA optimised filter. Lukac et al. [74] claimed that "The optimal GA-WVDF [W2] filters are consistent in performance even when the image corrupting noise differs quantitatively from the assumed during training noise model." To test the veracity of this assertion, both for the W2 filter and for filters developed using an IEA, the Lena image was contaminated by salt and pepper noise of a different level. In [74] the other noise levels used were 5% and 10%. It was decided that 8% was an appropriate level of noise for the second image.The noisy images are shown in Figure 3.3

The initial population in each run consisted of four mutants of the identity filter

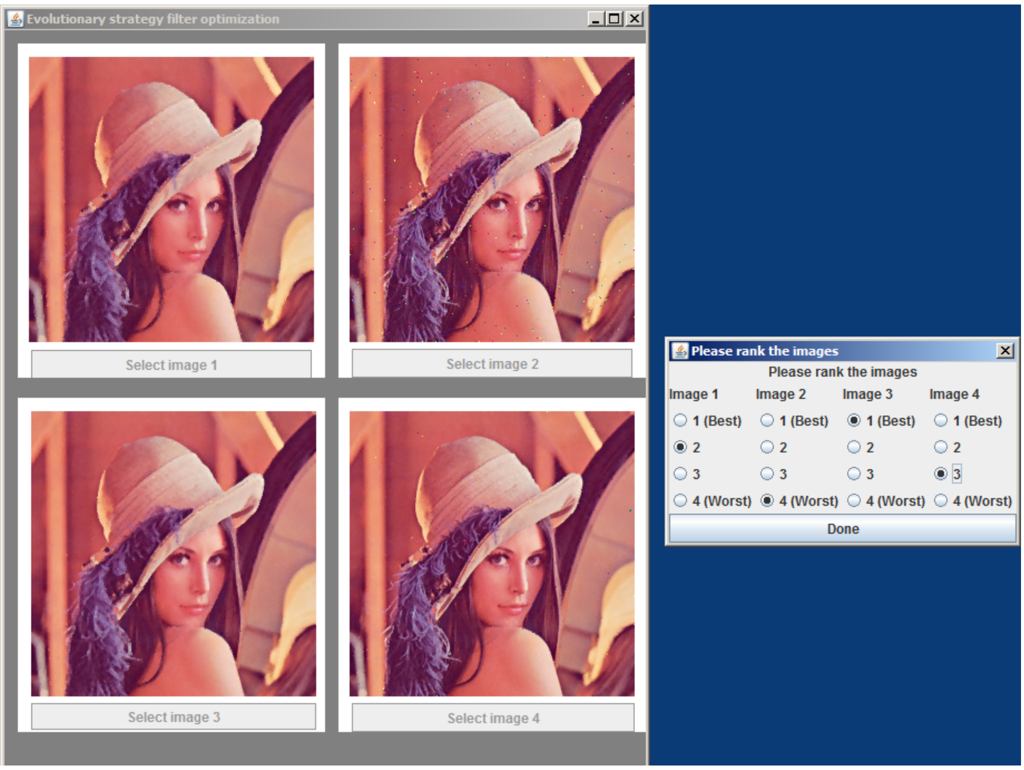
($w_5 = 1, w_i = 0$ for other values of i). The identity filter has no effect when applied to an image. The identity filter was chosen as the starting point because of its mathematical simplicity and to encourage the development of filters that were effective at removing noise but did not introduce many visible filter artefacts. It is likely that if stronger filters were used in the first generation, participants would be satisfied with the noise removal properties of the filters and neglect the effect of the artefacts introduced by the filters. This would lead to participants not developing optimal filters for the images, particularly for I1. The initial step sizes for the first generation were drawn uniformly randomly from the range $U[0.075, 1]$. Every generation thereafter consisted of the fittest filter of the previous generation and three mutant offspring spawned from it. The position of the image filtered by the selected member of the previous generation, the parent of the current generation, was determined randomly and placed amongst its offspring. This was done to ensure that any eye gaze positional bias would not affect the development of the filters.

3.3.3 Data gathered

The participants were instructed to give reasons for their choices of images whilst they were developing their filters; for example “Images 1 and 3 have less noise than 2 and 4, image 1 has an annoying pixel on Lena’s nose hence I choose image 3.” There were two reasons for having the participants do this; the first is that it encouraged participants to give more thought to their selections, the second is to aid in explaining the performances of the IQMs.

Asking the participants for their thoughts undoubtedly made them think more about their selections and elicited useful information regarding the criteria they used for adapting their IEA filters. Questioning also increased user fatigue and possibly reduced the number of generations a participant was willing to assess. However, it is important to realize that users of real world applications based on the IEA method would not be required to verbalize their thoughts.

After they had developed their filters, the participants were asked to compare the performance of four filters: their own filter developed on I1 (which is referred to as F1), their own filter developed on I2 (which is referred to as F2), the BVDF, and the GA optimised W2 filter (which has weights 0.1526, 0.2610, 0.2007, 0.2059, 1, 0.1992, 0.2115, 0.2581, 0.1435) developed by Lukac et al. [74]. The four filters were applied to one of the images (I1 or I2) and the results displayed in a 2 × 2 image array similar to the one used for developing the filters. A screenshot of the interface



Please rank the images

Image 1	Image 2	Image 3	Image 4
○ 1 (Best)	○ 1 (Best)	● 1 (Best)	○ 1 (Best)
● 2	○ 2	○ 2	○ 2
○ 3	○ 3	○ 3	● 3
○ 4 (Worst)	● 4 (Worst)	○ 4 (Worst)	○ 4 (Worst)

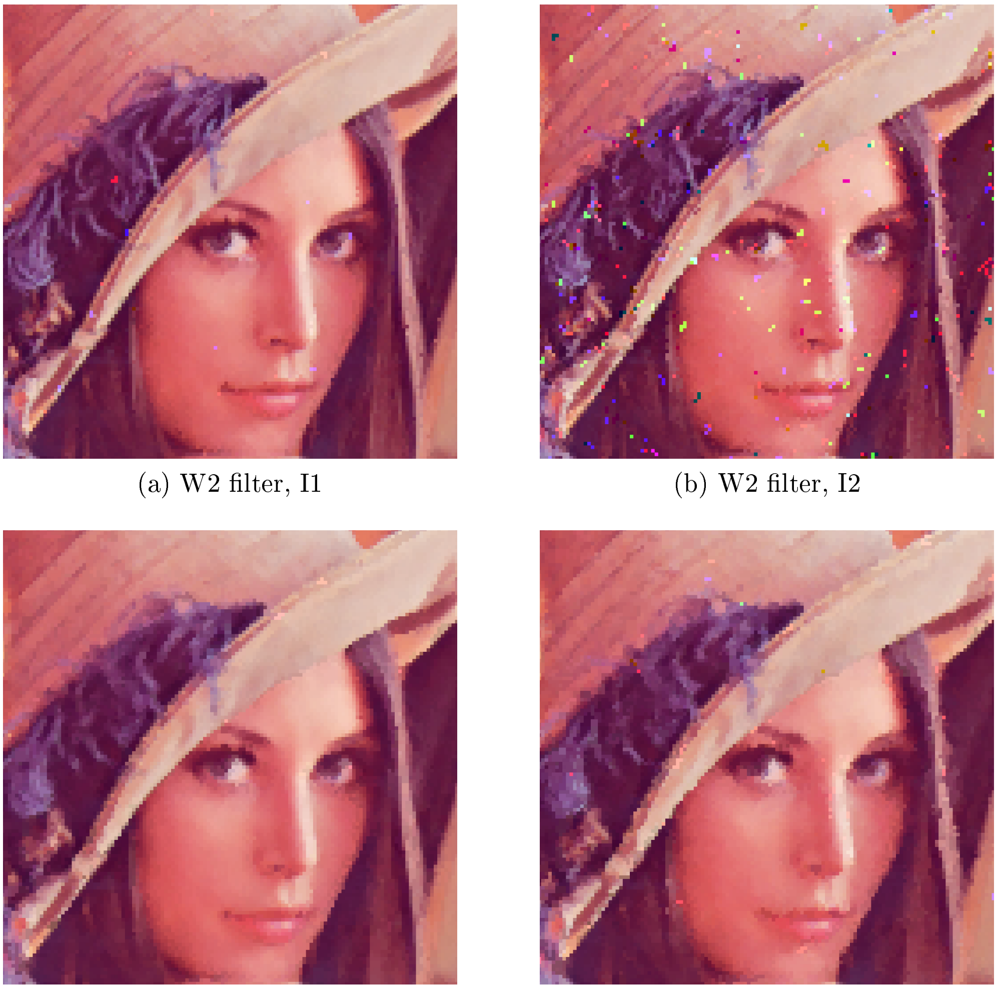
Figure 3.4: Screenshot of the user interface for the ranking of the filters in the noise removal experiment

is provided in Figure 3.4.The positional order in which the images were displayed to the participant was determined randomly. The images were not labelled so that the participant did not know which image corresponded to which filter. The participant was asked to rank the images in order of image quality, giving reasons for their preferences as they did when developing their filters. The ranking process was then repeated on the other image.

3.4 Results

3.4.1 Examples of filtered images

Figure 3.5 shows the result of applying the BVDF and W2 filters to the noisy images. Figure 3.6 shows the result of applying some of the participant developed filters which were ranked as 1 (best) at the comparison stage. It can be seen from Figure 3.6 that participants varied in their opinions of what constitutes a favourable filter. For example, Figure 3.6 (a) exhibits more noisy pixels but fewer visible image artefacts than Figure 3.6 (b).Table 3.1 gives the weights of the filters used on the



(a) W2 filter, I1

(b) W2 filter, I2

(c) BVDF, I1

(d) BVDF, I2

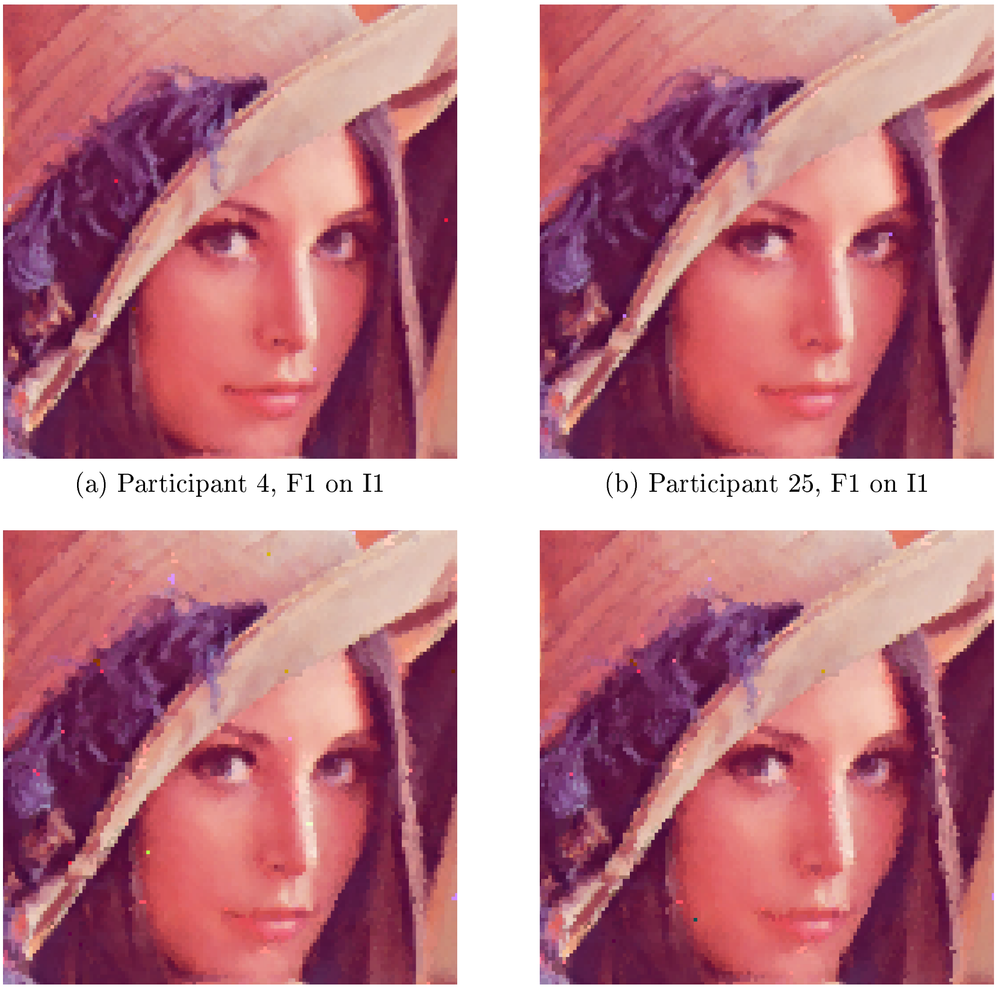
Figure 3.5: Sections of the results of applying the W2 and BVDF filters to the noisy images

images in Figure 3.6.

3.4.2 Analysis of the participant rankings

Kendall's W (see Section 2.2.4) was calculated for the participants' rankings over each of the two sets of ranked images. For I1, $\chi^2(3, N = 30) = 25.08, p < 0.001$, Kendall's W is 0.28 indicating weak agreement among the participants. For I2, $\chi^2(3, N = 30) = 68.64, p < 0.001$, Kendall's W is 0.76 indicating strong agreement among the participants.

The means and standard deviations of the participant awarded ranks are summarised in Table 3.2. Performing the Friedman test on the participant rankings on I1 showed that the difference between the performances of the filters was sig-



(a) Participant 4, F1 on I1

(b) Participant 25, F1 on I1

(c) Participant 15, F2 on I2

(d) Participant 29, F2 on I2

Figure 3.6: Sections of images filtered using participant developed filters. In all cases the participants had rated the images as the best of the four presented at the ranking stage. Note how (b) and (d) have more filter artefacts than (a) and (c).

Table 3.1: The weights of the WVDFs used to filter the images in Figure 3.6

		Weights								
Participant	Image	1	2	3	4	5	6	7	8	9
4	I1	0.4331	0.7122	0	1	1	0.1304	0.1338	0.3185	0.4507
25	I1	1	0.8171	1	1	1	1	1	0	0
15	I2	0.6779	0.8310	0.2161	1	1	1	0.0318	1	0.3992
29	I2	1	0.5047	0.7742	0.7747	0.5194	0.9620	0.8870	0.7555	0.9620

Table 3.2: Means (standard deviations) of the participants' rankings of the filtered images

	Filter			
Image	F1	F2	BVDF	W2
I1	1.97 (0.85)	2.27 (1.01)	2.27 (0.83)	3.50 (1.14)
I2	2.70 (0.54)	2.20 (0.61)	1.20 (0.55)	3.90 (0.55)

Table 3.3: Contingency table of participant preferences for filters F1 and F2 applied to images I1 and I2

	Image		
Preferred filter	I1	I2	Total
F1	19	8	27
F2	11	22	33
Total	30	30	60

nificant $\chi^2(3) = 25.08$, $p < 0.001$. Post hoc analysis using Fisher's LSD post hoc test indicated that the W2 filter performed significantly worse than the BVDF, the F1 filters, and the F2 filters (for the smallest significant difference, $t(87) = 4.283$, $p < 0.001$). For I2 the difference was also significant $\chi^2(3) = 68.04$, $p < 0.001$. The BVDF performed significantly better than the F2 filters $t(87) = 5.971$, $p = 0.001$ which in turn outperformed the F1 filters $t(87) = 2.986$, $p = 0.004$ which in turn outperformed the W2 filter $t(87) = 7.166$, $p < 0.001$.

These results do not give a clear indication on whether or not the filters developed for specific images perform better on those images. Table 3.3 provides preference counts of the participant developed filters F1 and F2 applied to I1 and I2. A chi-square test (see Section 2.2.4) performed on the data in Table 3.3 indicates that the filters developed for a particular image performed better on that image than they did on the other image $\chi^2(1) = 8.148$, $p = 0.004$.

3.4.3 Objective measures of image quality

To assess the efficacy of the IQMs, Spearman's ρ (see Section 2.2.4) between the ranks each participant assigned to the filtered images and the IQMs for the same filtered images were calculated. The means of the ρ values over all of the participants were calculated (see Table 3.4). Performing the Friedman test on the Spearman

Table 3.4: Mean (standard deviations) of the Spearman's correlation coefficients between the participant rankings and the IQMs

Measure	I1: 2% Noise	I2: 8% Noise
MAE	-0.253 (0.568)	-0.420 (0.346)
MSE	-0.287 (0.548)	0.867 (0.248)
MQE	0.400 (0.520)	0.900 (0.253)
NCD	-0.240 (0.537)	0.713 (0.291)
SSIM	-0.227 (0.527)	0.800 (0.257)

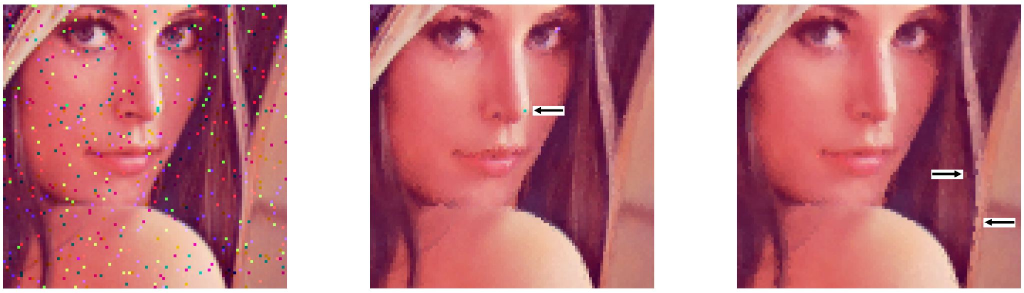
Table 3.5: Mentions of image selection considerations when developing and ranking the filters

	I1: 2% noise		I2: 8% noise	
Consideration	Developing	Ranking	Developing	Ranking
General noise	30	29	30	30
Single pixel	11	9	9	3
Filter artefacts	16	20	17	10

correlation coefficients for the IQMs on I1 showed that the difference between the performances of the IOMs was significant, $\chi^2(4) = 48.734, p < 0.001$. Post hoc analysis using Fisher's LSD post hoc test indicated that the MQE provided a significantly better model of human opinion than the other IQMs, $t(116) = 6.454, p < 0.001$ in comparison to the next best IQM; the SSIM. For I2 the difference was also significant, $\chi^2(4) = 84.854, p < 0.001$. The difference between the MQE and MSE was not significant $t(116) = 0.793, p = 0.430$. Both the MQE and the MSE performed significantly better than the SSIM, $t(116) = 2.730, p = 0.007$, which in turn performed significantly better than the NCD $t(116) = 2.730, p = 0.007$, which in turn outperformed the MAE, $t(116) = 8.719, p < 0.001$.

3.4.4 Selection considerations

When the participants were explaining the reasons for their image selections whilst developing and ranking their filters it was quickly found that the majority of their reasons could be summarised and placed into one of three categories: general noise, single pixels, and filter artefacts. The remaining comments were about how some images looked brighter than others, something which could not be attributed to the filters. It was realised that this effect was due to the variation with viewing angle of the perceived brightness of LCD displays. The three category observation allowed par-



(a) Section of I1

(b) Example of single pixel. The participant rejected this image because of the noise pixel on Lena's nose

(c) Example of filter artefact. The participant rejected this image because of the rough edge on Lena's hair

Figure 3.7: A section of I1 (a) with examples of single pixel (b) and filter artefacts (c)

ticipant comments to be recorded quickly using a simple code. For example, "Images 1 and 3 have less noise than 2 and 4, Image 1 has an annoying pixel on Lena's nose hence Image 3 is best. Lena's hair is more jagged here [participant points to Lena's hair near the eye on the viewer's right] on number 2 than on number 4 so I prefer 4 to 2" is written "$(1+3)/(2+4)$ GN, $3/1$ SP (Nose), $4/2$ IA (hair right near eye)". General noise refers to many noisy pixels, either over the entire image or a particular part (e.g. Lena's face). Single pixel refers to a particular noisy pixel that the participant has noticed, generally a pixel in an otherwise noise free part of the image and often in a prominent place such as on Lena's nose. Filter artefacts refer to parts of the image that have been worsened because uncontaminated pixels had been altered by the filter. Examples of single pixel and filter artefacts are given in Figure 3.7. The noticeable filter artefacts were normally introduced at boundaries between different parts of the image such as between Lena's upper arm and the mirror. Table 3.5 shows a count of the number of participants who mentioned each of the three considerations when developing their filters and when ranking the filters. A participant had to mention a consideration only once whist developing a filter for the consideration to be included in the counts of Table 3.5.

It can be seen from Table 3.5 that general noise was the most important consideration for the participants. This is to be expected as the goal is to improve the appearance of noise contaminated images. Single pixels were a more important consideration to the participants when ranking the filters applied to I1 than when ranking the filters applied to I2. This was because noisy regions in I1 were more likely to contain only a single noisy pixel after filtering than was the case for I2 which was more likely to have many noisy pixels. The difference between the number of participants who cited filter artefacts as a consideration at the ranking stage can be explained by the fact that as I2 was a noisier image, more of the participants found the introduction of filter artefacts to be of less concern than the removal of noise. For this reason, the participants gave more weight to the noisy pixels than to the filter artefacts.

The noisy pixels tended to have a larger deviation from the original values than the filter artefacts, thus a single noisy pixel tends to provide a greater contribution to an IQM value than a single image artefact pixel. The MQE gives more weight to the noisy pixels than any of the other IQMs, the MAE gives the least. This is why the MQE was the IQM which most closely modelled human opinion. There were more noisy pixels on I2 than I1 and in general this remained the case after filtering. This meant that the noisy pixels in I2 had a greater effect than the filter artefacts on the IQMs than was the case in I1. This explains why the IQMs (except the MAE, which does not give as much weight to noisy pixels as the other IQMs) performed better on I2 than on I1.

It was also observed that participants had a tendency to concentrate on a primary region of interest, only paying attention to other parts of the image once they were satisfied with the part they were focusing on. For example, a participant may choose to focus on Lena's face until they are satisfied that it is free from noise. They may then choose to focus on a particular noisy pixel in the background until that has been removed.

3.5 Conclusion

It has been demonstrated that the weights of a vector directional filter can be obtained using a simple IEA in which assessments of image quality are made by a human user. The method was more effective for improving perceptual image quality than a filter previously developed using a EA [74]. In the presence of 2% salt and pepper noise, the IEA filter developed on the 2% noise image was also more successful at improving image quality than the well known BVDF.

The assertion made in [74] that the WVDF filters were robust to changes in the noise level of the image is cast into doubt by the observation that filters tend to perform better on the images they developed on.

The poor performance of the EA based W2 filter in the experiment can be attributed to the use of the MAE as the fitness function used for its development. Five objective IQMs were evaluated and it was found that whilst all IQMs except the MAE provided a good model of human opinion on the noisier image, none of the image quality measures were satisfactory for both 2% and 8% noise. Of the five IQMs, the MAE was least similar to human perception of image quality. The nearly adequate performance of the MQE and the poor performances of the remaining IQMs provides evidence to support the use of human evaluation and the IEA approach.

The descriptive feedback provided indicates that the meaning participants assign to the image composition results in behaviour whereby they focus their attention on salient image regions. It would be difficult to design a numerical IQM that was capable of adapting its behaviour to reflect human interpretation of the image scene.

Chapter 4

Making use of rejection information using a hyperplane algorithm

4.1 Introduction

Minimising user fatigue is one part of ensuring that an interactive optimisation process performs as well as possible. An SMM-IEA, in which the user selects a single member of the population to seed the following generation, is relatively effortless compared to other assessment methods. Comparisons have been made between rating scales and evaluation methods to try to find evaluation methods that enable IEAs to present satisfactory results with minimal user effort. Yoon and Kim [145] compared a two point, a five point, and a continuous scale for rating individuals in an IEA. It was found that the participants preferred the two point scale. Takenouchi et al. [126] compared three methods of soliciting fitness ratings in an IGA: simple pairwise 'choose the best', pairwise with levels of disparity, and full scale rating of the entire population. It was found that the simple pairwise method was preferred by the participants. The approach explored in this experiment is to give users the option of explicitly rejecting individuals, essentially introducing a two point rating scale. The algorithm developed extends the SMM-IEA approach to use hyperplanes to segment the search space. This extension to the SMM-IES used in Chapter 3 will be referred to as the *hyperplane-IES*. In order for the use of the hyperplane-IES to be appropriate the subjective fitness function, the fitness function that is assumed to exist in the user's head, needs to be unimodal; the subjective fitness function should have no local optima. If the subjective fitness function is not unimodal, there is a good chance that the search will be directed toward a local optimum. There should be a single small region of the search space which produces optimum individuals, the farther the genotype of an individual is from this region, the less satisfactory its corresponding image should be.

When a new EA, or more commonly, a modification of some aspect of an existing EA, is developed its performance is compared to those of other algorithms using a suite of benchmark fitness functions such as those presented by Li et al. [73]. The benchmark functions are representative of the sorts of problem the EA would be used to solve. For example, He et al. [53] proposed a new fitness evaluation mechanism for multiobjective problems which was compared to the standard mechanism with the use of nine benchmark functions. There are no equivalent benchmark functions for IEAs. If IEAs are to be compared then some contrived task, for which an IEA may not even be a suitable approach, is devised and used.

A simple colour matching task was used by Breukelaar et al. [10] to compare the effectiveness of the self adaptive step size aspect of IESs to fixed step size parameters. A similar task was used by Cheng and Kosorakoff [16] for comparing the performances of IGAs to those of a variant called human-based GAs in which the user takes an active part in the recombination and mutation operations. The colour matching task is chosen for three reasons:

- The task can be explained, or demonstrated, quickly and with little chance of misunderstanding. This is important as participant misunderstanding of the task can add noise to the data gathered which is used to compare the performances of the algorithms.

- The task is not computationally intensive. The colour panels can be generated very quickly so that the participants to not have to wait between generations.

- The search space can be made perceptually uniform. This perceptual uniformity is achieved using the CIELAB [2] colour space. If a colour is chosen in the CIELAB colour space then all colours that lie on a sphere centred on that colour will, in theory, be perceived to be equally different from the chosen colour. The search space used in this experiment consisted of those colours in the CIELAB colour space which could be mapped to the sRGB colour space and thereby displayed on a monitor.

To discuss the suitability and the limitations of using the colour matching task as a trial task for evaluating IEAs, the IEA suitability questions of Section 2.2.3 are addressed:

- *Is the subjective fitness function unimodal in the search space?* Yes, a colour that is near the target colour in the search space will be perceived to be similar to the target colour. A colour that is farther from the target colour in the search space will be perceived to be less similar. It is for this reason that the colour matching task was chosen to test the hyperplane-IES.
- *What other approaches to the problem are available?* This question is not really relevant as the colour matching task is contrived for the purpose of testing IEAs. In reality a direct interface in which values are adjusted using three sliders would be a better approach for obtaining a colour match.
- *Is the search space large?* Large enough, though not very large. The search space has only three dimensions, but the difference between colours that can be distinguished by human perception is such that the search space consists of tens of thousands of colours.

Consideration of the IEA suitability questions indicates that the colour matching task is a reasonable approximation for a realistic task, weakened perhaps by the low dimensionality of the problem.

The purpose of this experiment is to see if there is any advantage in using the hyperplane-IES over the SMM-IES. A third IES was also used, what shall be referred to as the *dummy-IES*. The dummy-IES uses the same interface as the hyperplane-IES but in fact makes no use of the rejection information and is in fact the SMM-IES under the interface. The purpose of the dummy-IES is to see if any differences observed between the algorithms are due to the algorithms themselves or whether they are due to the difference between the interfaces.

4.2 Theory

In the SMM-IES, a user must choose the closest match to their target image to seed the next generation. In the hyperplane-IES extension to the SMM-IES the user may also, if they wish, choose to explicitly reject images, and thereby their corresponding genotypes, that are a particularly poor match for the target. Each of these rejected individuals is used to create a hyperplane which is used to make it less likely that genotypes will be generated in certain parts of the search space. The genotypes are represented by real-valued, N-dimensional vectors $\mathbf{c} = (c_1, c_2, \cdots, c_N)$ and there exists a target vector corresponding to the 'ideal' solution denoted by

$\mathbf{c}^t = (c_1^t, c_2^t, \cdots, c_N^t)$. In each generation, the user selects a preferred image with corresponding genotype $\mathbf{c}^s = (c_1^s, c_2^s, \cdots, c_N^s)$. The user may optionally reject one or more images with the i-th rejected image having corresponding genotype $\mathbf{c}^{r_i} = (c_1^{r_i}, c_2^{r_i}, \cdots, c_N^{r_i})$.

Consider a point P lying on the line $\mathbf{w} = \mathbf{c}^s - \mathbf{c}^{r_i}$ with distance $\alpha\,|\mathbf{w}|$ from point $\mathbf{c}^{r_i}$. We construct an $(N-1)$-dimensional hyperplane which passes through point P and which is orthogonal to the line $\mathbf{w}$. The hyperplane defines a discriminant function $f\,(\mathbf{c})$ which has the form

$$f\,(\mathbf{c}) = \mathbf{w} \cdot \mathbf{c} + \omega_0 \tag{4.1}$$

where

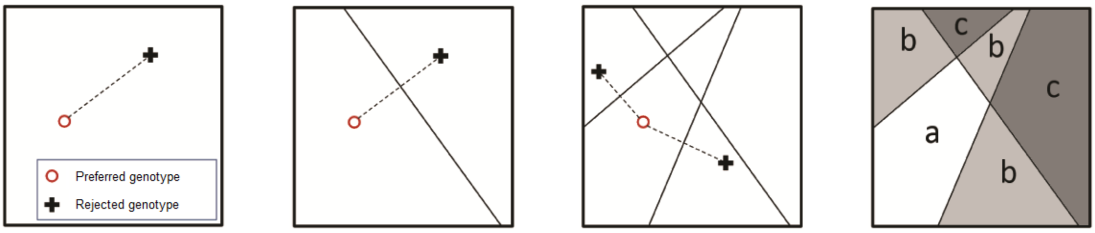
$$\omega_0 = -\left(\mathbf{w} \cdot \mathbf{c}^{r_i} + \alpha\,|\mathbf{w}|^2\right). \tag{4.2}$$

The discriminant function divides the space into two mutually exclusive regions: $\mathbb{R}_s^N$, the region in which $\mathbf{c}^s$ is located and $\mathbb{R}_{r_i}^N$, the region in which $\mathbf{c}^{r_i}$ is located. In general, for an arbitrary genotype $\mathbf{c}$, we have

$$\begin{array}{lcl} f\,(\mathbf{c}) > 0 & \Leftrightarrow & \mathbf{c} \in \mathbb{R}_s^N \\ f\,(\mathbf{c}) \leq 0 & \Leftrightarrow & \mathbf{c} \in \mathbb{R}_r^N. \end{array} \tag{4.3}$$

For every hyperplane a generated genotype $\mathbf{c}$ is behind, that is for every discriminant function for which $f\,(\mathbf{c}) \leq 0$, the genotype has a probability p_r where $0 \leq p_r \leq 1$ of being rejected. If $\mathbf{c}$ is behind n hyperplanes the its probability of being rejected is $1 - (1 - p_r)^n$. In this way, the probability landscape is successively modified to favour the generation of genotypes from within particular regions of the search space lying closer to $\mathbf{c}^s$. The essentials of a hyperplane-IEA are given in Figure 4.1.

The fundamental assumption in the hyperplane approach is that the user selected preferred image in the generation, which has genotype $\mathbf{c}^s$, will lie closer to the target vector $\mathbf{c}^t$ than the rejected vector $\mathbf{c}^{r_i}$. If this assumption was valid we could set $\alpha = 0.5$, that is, place the hyperplane corresponding to rejected point $\mathbf{c}^{r_i}$ exactly halfway between $\mathbf{c}^s$ and $\mathbf{c}^{r_i}$ with no possibility of placing the hyperplane between $\mathbf{c}^s$ and $\mathbf{c}^t$. It follows that as there would be no chance of any hyperplanes being placed that caused $\mathbf{c}^t$ to be in $\mathbb{R}_r^N$. Any genotypes generated in $\mathbb{R}_r^N$ could be rejected safely by the algorithm without impeding the search, hence we could set $p_r = 1$. This assumption does not always hold, however. There are two reasons for this.



(a) The current preferred individual and a genotype explicitly rejected by the user.

(b) A hyperplane is placed perpendicular to the vector between the preferred individual and the genotype.

(c) Two more genotypes (for example) are rejected by the user, hence two more hyperplanes are added to the search space.

(d) A genotype generated in region 'a' has no chance of being rejected by the algorithm. A genotype generated in a region 'b' is rejected with a probability p_r. A genotype generated in a region 'c' is rejected with probability $1-(1-p_r)^2$.

Figure 4.1: The essentials of a hyperplane-IEA in a simple 2D space.

The first is that in general the search space is not likely to be perceptually uniform — though this is less of a problem with colour matching in the CIELAB colour space than it would be for most problems — so that two images may seem to be equally similar to the target image when their genotypes are not equidistant from the target genotype in the search space. The second reason is that due to the human threshold of perception there is uncertainty in the perceptual distance between images. As well as accounting for these factors by setting p_r to be less than unity, the hyperplane approach can be made more 'forgiving' by setting a lifetime, l, for each hyperplane so that after l generations the hyperplane is removed. For example, if $l = 4$, a hyperplane added after the evaluation of generation 3 would remain in place when the populations of generations 4, 5, 6, and 7 are created. The hyperplane would be removed before the 8-th generation is created. It is also sensible to remove hyperplanes when an individual whose genotype lies behind one or more hyperplanes is selected as the preferred individual. If such hyperplanes are not removed then new individuals that are similar to the current preferred individual could be rejected by the algorithm when in fact they should be accepted. Therefore, if the preferred genotype lies behind any hyperplanes, the hyperplanes are removed before cloning and mutation.

The hyperplane-IEA bears a passing resemblance to support vector machines (SVMs). SVMs are supervised learning models that use hyperplanes to, at the most basic level, classify data into one of two classes [118]. An SVM uses data of a known classification to place a hyperplane between the data of the two classes so as to maximise the separation between the classes and the hyperplane, or, minimise the error if it is impossible to place a hyperplane without some data points being misclassified by the hyperplane. The hyperplane-IEA places a hyperplane between a single point belonging to the class *acceptable* and single point of the class *rejected*. The position of the hyperplane is determined by a fixed parameter α. In an SVM multiple hyperplanes are used to classify the data into more than two groups. In the hyperplane-IEA multiple hyperplanes are used to divide the search space into a region of *acceptable* which is at least partially surrounded by a region (or regions) *rejected*. In an SVM any new unclassified data points are classified according to which side of the hyperplane(s) they are. In the hyperplane-IEA the genotypes of potential members of the next generation are classified as being *acceptable* if they lie in the *acceptable* region, however, if they lie in the *rejected* region they may still be classed as *acceptable*. The goal of an SVM is to classify the unclassified data as accurately as possible. The goal of the hyperplane-IEA classification process in to ensure that the search for a satisfactory solution can be completed in as few generations as possible, that is, the accuracy of individual classifications is unimportant as long as the hyperplanes as a whole help the IEA to attain a satisfactory solution.

4.2.1 Finding the optimal values of l, p_r, and α

The values of l, p_r, and α should be set to appropriate values in order to make best use of the hyperplane-IES. As it was not feasible to test even a few combinations of the parameters properly using human evaluation a virtual user was employed. A virtual user is a model of a human user used to aid in the testing of IEAs. A virtual user may be an ideal user which always chooses the individual closest to the global optimum, never tires, is consistent in its behaviour, and will not stop until it has attained a solution very close to the global optimum. The virtual user may instead be designed to provide a more realistic model of human behaviour by, for example, being inconsistent in its opinions of the individuals, stopping long before an optimal solution is attained, or being inconsistent in how the individuals are rated. Virtual users have been used to provide proof of concept when pioneering new algorithms. In these cases, the IEA is effectively an EA in which the limitations imposed by the use of human users have been imposed, most notably the population size and the number of generations the algorithm will run for. Takagi and Pallez [125] used an ideal user to investigate the feasibility of interactive differential evolution (IDE) using Gaussian mixture models as a fitness function. Hornby and Bongard [55] developed a hybrid algorithm which would model the user and be used to evaluate individuals on their behalf in order to reduce fatigue; an ideal user was used to evaluate the effectiveness of the algorithm. Virtual users have also been used as a preliminary step in experiments involving human participants. Breukelaar et al. [10] used an ideal user to measure the difference between performances of the participants and the best possible performance when using an IEA to perform a colour matching task. Tanaka et al. [128] developed an IEA which was designed to converge upon multiple optima in a multimodal search space; to confirm that the algorithm could achieve this an ideal user was tasked with finding multiple optima in 2, 4, and 6 dimensions.

Virtual users have also been used to compare the effects of different parameter settings in IEAs. Kelly et al. [63] used an ideal user to compare the effect of changing the relative weighting of the wheels in a dual roulette wheel parent selection algorithm. Frowd [34] used a virtual user to confirm that the choices made for mutation probability, population size, and the way in which elitism was used in EvoFIT were appropriate. The decisions made by the virtual user were found to correlate well with those made by participants in previous experiments. Pallares-Bejarano [90] used data from previous experiments to help set the behaviour of the virtual user which was then used to compare the impact of various mutation probabilities, population sizes, and search space dimensions on various IEAs.

Developing the virtual user

The virtual user developed in this experiment was designed to account for three particular aspects of user behaviour:

- How a user perceives the distances between the target colour and the colours in the population.
- The distance from the target colour at which a user would be satisfied with a colour match.
- How a user decides how many colours they are going to reject.

The development of the virtual user was considered a minor aspect of the work and as such did not warrant the time-consuming process of gathering data from a number of participants. the data used to set the parameters of the virtual user were gathered over 21 runs of the algorithm during which a total of 457 generations of colours were evaluated by the author. The hyperplane-IES was not used for these runs, the rejected colours were recorded, but no use was made of the information. During the recording of the user behaviour the following data were recorded:

- The distances from the target colour to all of the colours for each generation.
- The distance between the target colour and the user selected colour for each generation.
- The number of colours, n, rejected for each generation.

From the gathered data the following values were calculated for developing the virtual user:

- The mean ($\mu_{hf} = 0.875$) and standard deviation ($\sigma_{hf} = 0.545$) of the final distances over the 21 runs.
- The means (μ_d) and standard deviations (σ_d) of the distances between the colours in the population and the target colour for each generation.
- The mean ($\mu_{he} = 0.812$) and standard deviation ($\sigma_{he} = 1.807$) of the error over all of the generations, the error for any one generation being the difference between the distance of the user selected closest colour match and the distance of the actual closest colour match in the CIELAB colour space.
- The mean ($\mu_{hr} = 5.611$) and the standard deviation ($\sigma_{hr} = 2.436$) of the number of colours rejected over all of the generations.

The virtual user was constructed based on four assumptions:

- A user does not become fatigued during the course of a run as the colour matching task is undemanding and quick to accomplish; the virtual user's behaviour does not account for fatigue.
- It was assumed that the number of colours rejected, n, depended on the mean μ_d and the standard deviation σ_d of the distances between the colours in the population and the target colour in some way. The assumed form was

$$n = \text{round}\left(m_n \cdot f\left(\mu_d, \sigma_d\right) + q_n\right) \tag{4.4}$$

where $f(\mu_d, \sigma_d)$ is some simple, as yet undefined, function of μ_d and σ_d, and m_n and q_n are constants to be found through a series of calculations.

- Users are not capable of identifying which of two colours is closest to the target if the difference in the distance between the colours is small in comparison to the distances of the colours from the target. The virtual user perceives all of the colours in the population (but not the target colour) to be slightly different to their actual colours. A colour $\mathbf{c} = (c_{L^*}, c_{a^*}, c_{b^*})$ is perceived to be at position $\mathbf{c}' = (c'_{L^*}, c'_{a^*}, c'_{b^*})$ having been translated according to

$$c'_{\{L^*,a^*,b^*\}} = c_{\{L^*,a^*,b^*\}} + (\mu_d \cdot m_s + k_s) \cdot N(0,1) \tag{4.5}$$

  where $(\mu_d \cdot m_s + k_s)$ is the translation factor and $N(0,1)$ is a random number from the Gaussian distribution. The k_s term is to account for human threshold of perception as below a certain threshold two colours are impossible for a user to distinguish. μ_d is the mean distance of the colours from the target and m_s is a constant of proportionality. m_s and k_s are constants to be found though experimentation.

- The virtual user is satisfied with a colour match when the closest perceived colour to the target is at a distance less than some threshold t.

A number of simple functions $f(\mu_d, \sigma_d)$ were tested. The values of each $f(\mu_d, \sigma_d)$ were calculated for each generation and the Pearson's correlation coefficients between n and the $f(\mu_d, \sigma_d)$s were calculated. It was found that that the correlation coefficient of greatest magnitude was -0.538 and was for $f(\mu_d, \sigma_d) = \mu_d/\sigma_d$. A plot of number of colours rejected versus μ_d/σ_d was used to find an approximate linear function for the number of colours rejected depending on μ_d/σ_d. This function is

$$\text{number of hyperplanes} = \text{round}\left(14.484 - 3.209 \cdot \frac{\mu_d}{\sigma_d}\right) \tag{4.6}$$

The values of m_s, k_s, and t were found heuristically by adjusting the values of m_s, k_s, and t and observing their effects on μ_{vf}, σ_{vf}, μ_{ve}, σ_{ve}, μ_{vr} and σ_{vr}. For each set of estimated values of m_s, k_s, and t the virtual user was run 500 times. The virtual user's output values of the mean (μ_{vf}) and standard deviation (σ_{vf}) of the final distances over the 500 runs, the mean (μ_{he}) and standard deviation (σ_{he}) of the error over all of the generations, and the mean (μ_{vr}) and standard deviation (σ_{vr}) of the number of colours rejected were recorded and compared to the equivalent user

Table 4.1: Developing the virtual user: the final measured scores

Virtual user	user
$\mu_{vf} = 0.853$	$\mu_{hf} = 0.875$
$\sigma_{vf} = 0.472$	$\sigma_{hf} = 0.545$
$\mu_{ve} = 0.831$	$\mu_{he} = 0.812$
$\sigma_{ve} = 1.712$	$\sigma_{he} = 1.807$
$\mu_{vr} = 5.611$	$\mu_{hr} = 5.543$
$\sigma_{vr} = 2.330$	$\sigma_{hr} = 2.436$

outputs μ_{hf}, σ_{hf}, μ_{he}, σ_{he}, μ_{hr}, and σ_{hr} respectively. The values of m_s, k_s, and t that were finally chosen were $m_s = 0.205$, $k_s = 0.043$, and $t = 0.66$. These values gave the output values shown in Table 4.1, which were deemed to be satisfactory.

Using the virtual user to find l, p_r, and α

The virtual user was used to perform the colour matching task 2000 times for various combinations of l, p_r, and α. The mean of the number of generations required to achieve a colour match over each set of 2000 runs was recorded.Table 4.2 shows the mean number of generations required for various l and α when $p_r = 1$. Table 4.3 shows the mean number of generations required for various p_r and α when $l = \infty$. With reference to Tables 4.2 and 4.3 it can be seen that the likely optimal number for l is 3 or 4, with $0.45 \leq p_r \leq 0.6$, and $0.25 \leq \alpha \leq 0.75$. With reference to Table 4.4 it can be seen that setting $l = 4$, $p_r = 0.6$, and $\alpha = 0.55$ enabled the virtual user to achieve a colour match, on average, in the fewest number of generations.

4.2.2 The IEAs used

Three IEAs are used in this experiment: the hyperplane-IES, the SMM-IES upon which it is based, and the dummy-IES which uses the same interface as the hyperplane-IES but works identically to the SMM-IES, that is, it ignores the rejection information provide by the user. In all three algorithms each colour is represented by a genotype consisting of four genes. Three of the genes are the L^*, a^*, and b^* colour components, are real coded, and have values limited to those which can be mapped to the sRGB colour space. The fourth gene is the mutation step size, σ, and is subject to the condition $\sigma \geq 1$

Table 4.2: Virtual user: mean number of generations required to achieve a colour match for various values of plane lifetime l and plane distance α from the rejected pointwith $p_r = 1$

	α			
l	0	0.25	0.5	0.75
1	25.41	24.08	22.24	21.65
2	22.54	19.91	18.75	21.39
3	20.17	18.36	18.24	23.60
4	19.92	18.20	19.58	26.75
5	19.99	18.73	21.59	31.35
6	20.39	19.39	23.40	35.88
7	20.81	20.55	25.44	40.57
8	21.24	21.42	28.25	45.11
9	22.25	22.94	30.13	53.13
10	23.05	23.85	32.58	58.48
11	23.46	25.14	36.98	66.11
12	24.58	27.37	40.96	72.12
13	26.08	29.87	43.89	80.83
14	26.21	30.95	49.57	93.22
15	27.56	32.43	54.15	101.54

Table 4.3: Virtual user: mean number of generations required to achieve a colour match for various values of rejection probability p_r and plane distance α from the rejected pointwith $l = \infty$

	α			
p_r	0	0.25	0.5	0.75
0.05	26.26	26.33	25.86	26.01
0.10	25.44	25.22	24.32	24.98
0.15	25.21	23.58	22.89	22.84
0.20	24.06	22.98	21.46	22.58
0.25	23.65	21.95	20.15	21.44
0.30	23.15	21.28	19.63	20.77
0.35	22.41	20.18	18.55	20.59
0.40	21.71	19.49	18.21	21.53
0.45	21.34	19.07	17.89	21.71
0.50	21.06	18.80	18.19	22.53
0.55	20.70	18.36	18.13	24.77
0.60	20.21	17.80	19.48	27.19
0.65	20.22	18.18	19.97	30.66
0.70	20.54	18.68	21.74	36.66
0.75	20.81	18.93	24.49	47.19
0.80	20.61	21.99	26.00	55.68

Table 4.4: Virtual user: mean number of generations required to achieve a colour match

	p_r							
	$l = 3$				$l = 4$			
α	0.45	0.50	0.55	0.60	0.45	0.50	0.55	0.60
0.25	22.55	21.89	21.10	20.46	21.09	20.56	19.66	19.09
0.30	22.06	21.30	20.73	20.12	20.82	19.92	19.14	18.56
0.35	21.76	20.88	20.48	19.69	20.07	19.25	18.64	18.24
0.40	21.15	20.50	19.76	19.19	19.91	19.08	18.54	18.00
0.45	20.99	20.24	19.33	19.04	19.20	18.94	17.96	17.85
0.50	20.38	19.70	19.14	18.71	19.35	18.75	18.24	17.86
0.55	20.32	19.68	19.18	18.54	19.02	18.60	17.98	17.76
0.60	20.50	19.52	18.94	18.70	19.21	18.83	18.23	18.22
0.65	20.22	19.55	19.00	18.90	19.53	18.99	18.83	18.73
0.70	20.15	19.59	19.21	19.02	19.25	19.06	19.06	19.17
0.75	20.42	19.97	19.66	19.45	19.80	19.71	19.72	19.53

4.3 Method

4.3.1 The user interfaces

Two slightly different interfaces were used for the colour matching experiment. Interface A (Figure 4.2) was the simpler of the interfaces and was used for the SMM-IES. Interface B was used for the hyperplane-IES and the dummy-IES. Each interface consisted of a main panel and a smaller panel to the right. The main panel itself consisted of nine colour panels. Each colour panel had an inner and an outer colour. The outer colour was the target colour. The outer colour was the same for all panels and did not change during the course of a run. The inner colours were changed by the algorithm according to the participant's input. The participant's goal was to make the inner colour of any one of the panels match the outer colour.

The SMM-IES required the participant to select the colour panel with the inner colour which, in the participant's opinion, had the closest match to the outer colour. When a colour was selected, its panel's border would turn green. The participant could change their mind and select another panel, but they could only select one. After a panel was selected the participant could either end the run by clicking the

'Finish' button if they were satisfied with the colour match or proceed to the next generation of colours by clicking the 'Next' button if they were not.

Interface B (Figure 4.3) was the interface used for the hyperplane-IES and the dummy-IES. Interface B had all of the functionality of Interface A but with added controls to enable the rejection of particularly dissimilar colours. A participant could right click a panel to explicitly reject a colour. The border of the panel of a rejected colour would turn red. Right clicking the panel a second time would 'un-reject' a colour. Two extra buttons were added to the right panel: 'Reject all' and 'Reject none'. Reject all would reject all of the colours, that is, turn all of the borders red with the exception of the selected colour. Reject none would clear all of the rejections, that is turn all of the borders black with the exception of the selected colour. The participant could reject anything from zero to eight colours, though they still had to select one colour.

Nine colour panels were presented to the participant at each generation. There were two reasons for this. The main reason is that in a genuine image processing task it would be unlikely that there would be space on the screen for any more than nine images of reasonable size. The other reason is to limit the effects of fatigue. The burden of selecting a single best individual from a population of n scales linearly with n, however, the burden of choosing to select or reject the remaining members of the population scales with $2^{(n-1)}$.

4.3.2 Test set-up

Twenty-four participants were used in this experiment. The participants were predominantly postgraduate students in the School of Physical Sciences at the University of Kent, the remainder were undergraduate students or staff. As the task was cognitively simple, no knowledge was required except a very basic level of computer literacy. However, because of the nature of the colour matching task, it was important that participants were not colour blind.

At the start of the experiment the participants were read a script detailing what the experiment consisted of. There then followed a practice run using the SMM-IES and a second practice run using the dummy-IES. The colour used on these runs was a shade of cyan (CIELAB values $(53.02, 5.12, -45.05)$, sRGB values $(69, 128, 204)$).The point of these runs was to ensure that the participants knew how to use the interfaces and could ask questions about the experiment. No data were recorded for these runs. After any questions were answered the recorded experiment began. Each participant

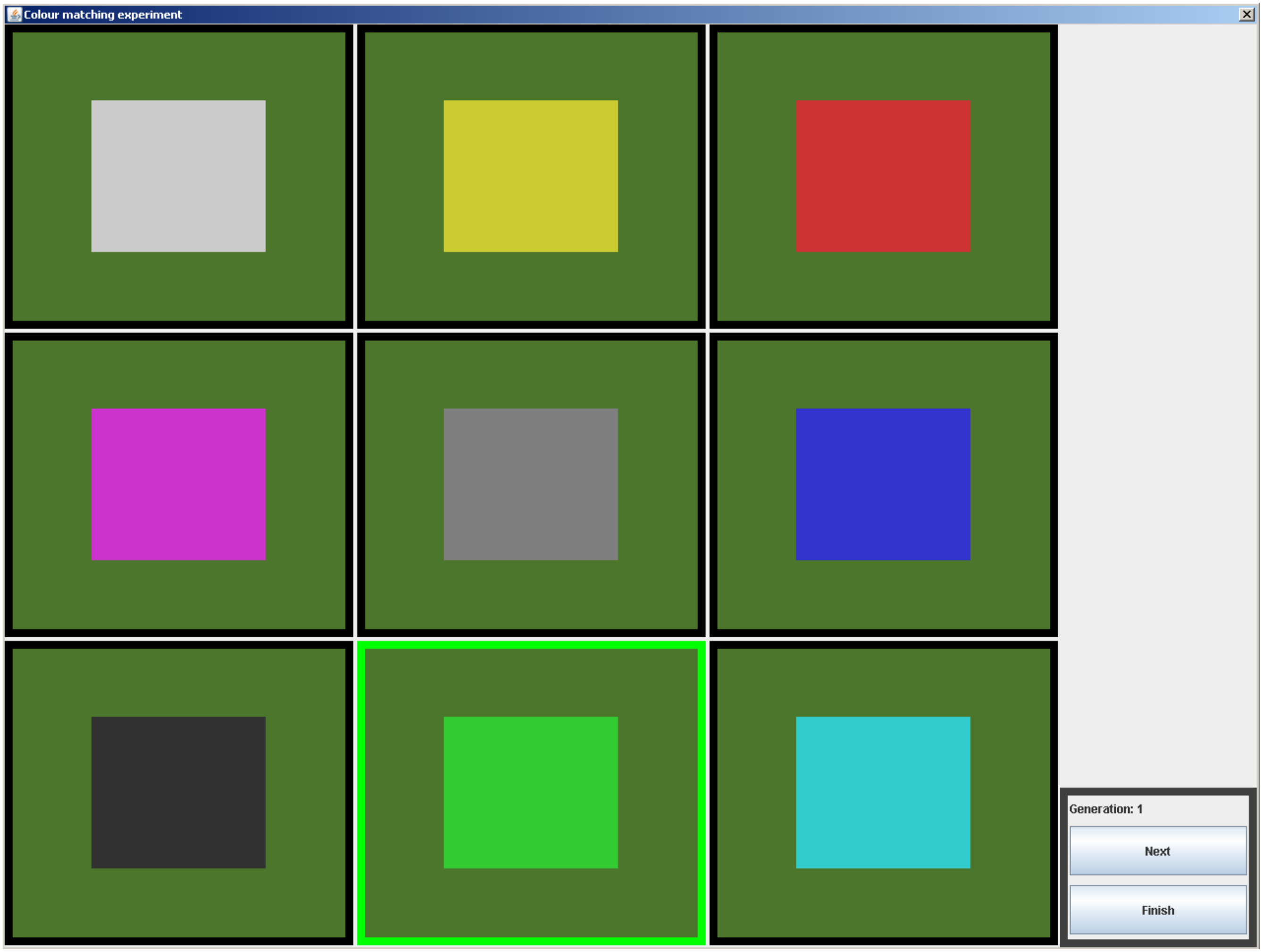


Figure 4.2: Screenshot of Interface A for the colour matching task showing the initial population of colours

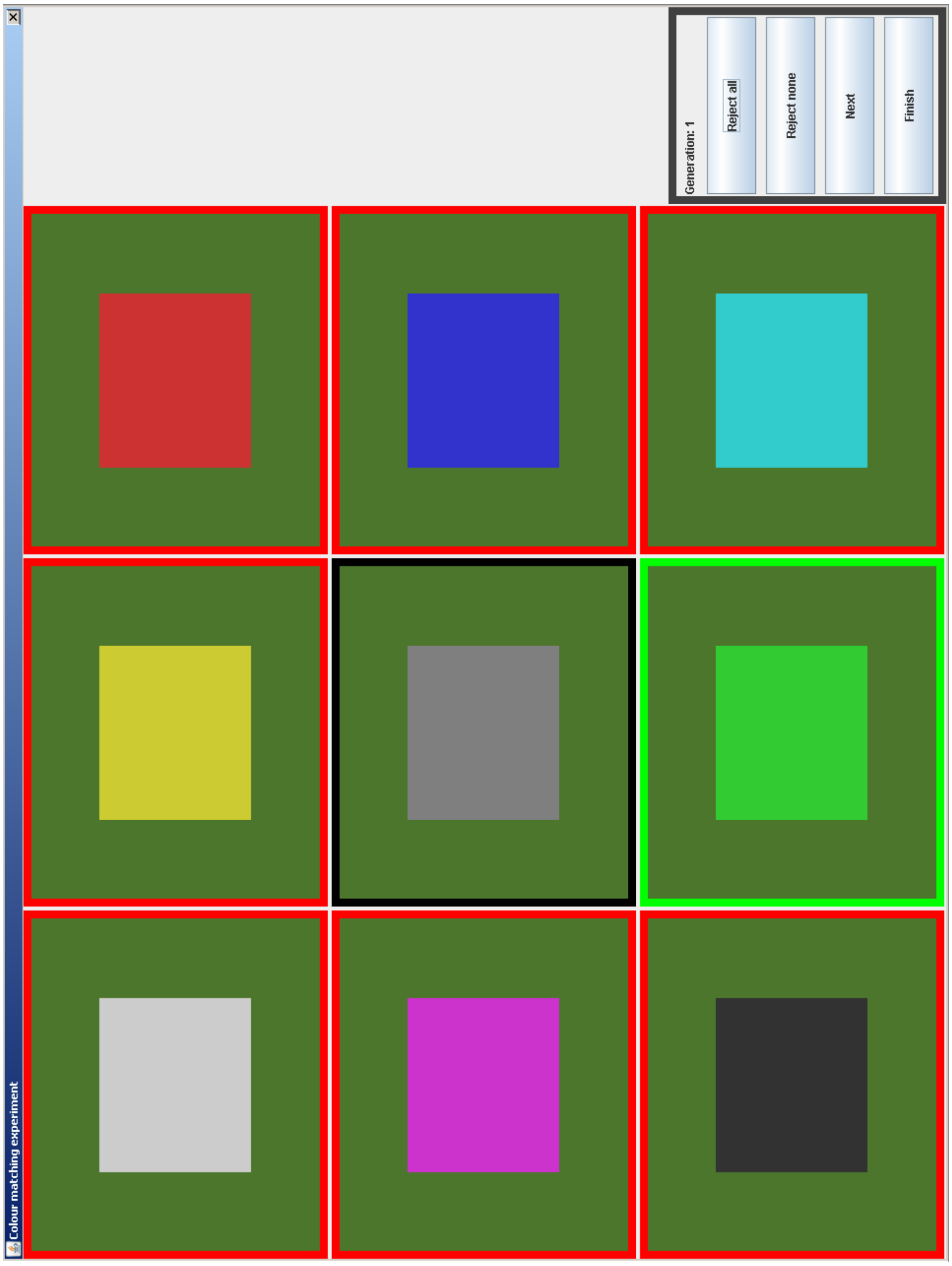


Figure 4.3: Screenshot of Interface B for the colour matching task

completed six runs in the recorded part of the experiment, one run for each of the algorithms trying to match a shade of orange (CIELAB values $(65.12, 23.36, 57.74)$, sRGB values $(220, 140, 50)$), and one run for each algorithm to match a dark shade of green (CIELAB values $(45.43, -27.94, 34.81)$, sRGB values $(75, 118, 45)$). Half of the participants performed the task using the orange target colour first, the other half the green target colour first.

The usual approach to generating the initial population in an EA is to generate the individuals at random. With the colour matching task this would have consisted of drawing nine colours at random from the search space. The drawback of using a random initial population is that the results would have been severely affected by whether one of these random colours happened to be close to the target colour, as the search space was not particularly large. Therefore, the initial population of colours were chosen and coded into the algorithm as opposed to being generated randomly. It is desirable, though not important, that the initial population of colours has the property that each colour in the population has an equal chance of being the closest colour to any other colour selected at random from the search space. The search space used for this experiment had a shape not conducive to finding initial colours that had this property. The sRGB colour space, however, does. If the initial population of colours are chosen such that one is at the centre of the sRGB cube and the remainder are on the diagonals from the centre to each of the eight vertices then finding a suitable initial population of colours becomes relatively easy. Of course, the initial population will only approximately satisfy the equal probability property but the initial population as defined is more evenly spread that a random initial population would be. The initial distribution is of less concern than trying to ensure that the target colours are the same distance from their respective closest colours in the initial population. To clarify, it should not be the case that one target colour is close to one of the initial colours whilst the other target colour is not close to any of the initial colours. If one target colour was closer to an initial colour than the other target colour it would introduce a bias. The distance between two colours x and y in the CIELAB colour space is

$$d = \sqrt{(x_{L^*} - y_{L^*})^2 + (x_{a^*} - y_{a^*})^2 + (x_{b^*} - y_{b^*})^2}. \tag{4.7}$$

The distance between the orange target colour (CIELAB values $(65.12, 23.36, 57.74)$) and the yellow initial colour (CIELAB values $(79.76, -17.16, 70.85)$ is 45.03. The distance between the dark green target colour (CIELAB values $(45.43, -27.94, 34.81)$)

and the middle grey initial colour (CIELAB values $(53.19, 0, 0)$) is 45.80.

4.3.3 Data gathered

Three kinds of objective data were collected: the time taken for the participant to complete a run, the number of generations they took, and the accuracy of the final colour match as measured in the CIELAB colour space.

Although objective measures can be used to measure the effectiveness of the algorithm, the participants' perceptions are also important. If a participant perceived one algorithm to take less time than another, even if it in fact took longer, this is of interest. After the first three runs (one for each of the IESs) the participants were asked the following questions:

1. Which run did you feel took the least amount of time?
2. Which run did you feel took the most amount of time?
3. Which run did you feel was easiest?
4. Which run did you feel was hardest?
5. In which run did you feel you had the most control?
6. In which run did you feel you had the least control?

For example, a participant may say that the first run took the least amount of time and the second one the most, that the first run was easiest and the third one the hardest, and that the second run offered most control and the third run the least.

After the second three runs, the participants were asked the same questions again but only in reference to the fourth, fifth, and sixth runs.

4.4 Results

4.4.1 Comparisons between the IESs

Each participant performed two runs using each of the SMM-IES, hyperplane-IES, and the dummy-IES. Following the advise given by Byron Morgan of the statistics help-desk at the University of Kent to not treat repeated evaluations of a single treatment by a single participant as having been performed by different participants, the average of each of the measured variables (final distances, number of generations, time taken, ease of use, perceived speed, and perception of control) over the two runs performed by a participant on a single treatment was used. These averages were treated as a single run for the purposes of finding the means and standard deviations of the measured variables, so that each participant was treated as having performed only one run using each of the algorithms. The means and standard deviations of the measured variables over all of the runs for each of the algorithms are presented in Table 4.5.

Performing the Friedman test (see Section 2.2.4) on the distances between the participants' final colours and the target colours for the three IESs showed that the differences between the final distances were not significant, $\chi^2(2) = 2.583, p = 0.2748$. The Friedman test performed on the number of generations taken to achieve a colour match showed that the differences between the SMM-IES, the hyperplane-IES, and the dummy-IES were also not significant, $\chi^2(2) = 3.935, p = 0.140$. Performing the Friedman test on the time taken showed that the difference in between the algorithms was significant, $\chi^2(2) = 39.857, p < 0.001$. Post hoc analysis using the Fisher LSD post hoc test for ranks indicated no significant difference between the hyperplane-IES and the dummy-IES, $t(118) = 1.196, p = 0.236$ but the SMM-IES was significantly faster than the hyperplane-IES, $t(118) = 5.958, p < 0.001$, and the dummy-IES, $t(118) = 7.149, p < 0.001$.

Performing the Friedman test on the ranks awarded by participants on the ease of use between the SMM-IES, the hyperplane-IES, and the dummy-IES showed that the differences between the algorithms were not significant, $\chi^2(2) = 4.740$, $p = 0.0935$. The Friedman test performed on perceived speed of the algorithms showed that the differences between algorithms were also not significant, $\chi^2(2) = 2.5552$, $p = 0.279$. Performing the Friedman test on the participants' perception of control showed that the differences between the algorithms were significant, $\chi^2(2) = 19.844$, $p < 0.001$. Post hoc analysis indicated a statistically significant greater level of perceived control for the hyperplane-IES over the dummy-IES, $t(118) = 2.229$, $p = 0.0277$, and a statistically significant greater level of perceived control for the dummy-IES over the SMM-IES, $t(118) = 4.830$, $p < 0.001$. The above results show that the participants felt they had more control over the process when they were using the hyperplane-IES than when they were using the dummy-IES and the SMM-IES and more control when they were using the dummy-IES than they did when they were using the SMM-IES. The SMM-IES provided a quicker colour match than the hyperplane-IES and the dummy-IES, even though the participants did not perceive this to be the case.

Table 4.5: Means (standard deviations) of the dependent variables in the evaluation of the effectiveness of the hyperplane-IES

Algorithm	Generations	Time taken	Final distance	Perceived speed	Ease of use	Control
SMM	20.81 (8.93)	96.5s (59.7s)	2.96 (2.00)	1.90 (0.72)	1.88 (0.68)	2.42 (0.65)
Hyperplane	17.88 (7.84)	168s (105s)	3.29 (3.20)	1.94 (0.66)	1.90 (0.75)	1.60 (0.61)
Dummy	22.75 (12.19)	181s (116s)	3.38 (1.96)	2.17 (0.60)	2.23 (0.59)	1.98 (0.60)

4.4.2 Correlations between the measures

The Spearman's correlation coefficients between the dependent variables were calculated for each of the IESs. Table 4.6 shows the correlation coefficients and their p-values. To avoid treating a single participant as two participants, the averages of the measured variables over the two runs performed by each participant for each IES were used.

Table 4.6 shows that the final distance was not significantly correlated with any of the other measured variables. The time taken and the number of generations have a strong correlation. The perceived speed and ease of use were strongly correlated. The strong correlation between the perceived speed and the ease of use does suggest that participants felt that the algorithm that was easiest to use was the one that participants felt enabled a colour match in the quickest time.

4.4.3 Discussion

The large variances in the data, particularly in the objective measurements, are worth remarking upon. The large variances could be explained by differences in the participants' ability to distinguish between colours, or perhaps because some participants became fatigued before the end of a run. If either, or a combination, of these cases accounted for the entirety of the variation then one would expect to see a negative correlation between the number of generations or the time taken and the final distance. This would be because the algorithms would be moving toward an optimal colour match and a participant who was satisfied with a relatively poor colour match would finish sooner and after fewer generations. An observation was made during the experiment that sometimes a colour would be selected that was a close match to the target colour but the step size would still be large. This colour would be selected repeatedly over subsequent generations as no closer match would be generated because the large mutation step size meant that offspring colours were generated that were farther from the target colour than the preferred colour was. Eventually the algorithm would generate a closer match or the participant would give up. This tendency of the IEA to get stagnate in this way suggests that the use of a self adaptive mutation step size is not an appropriate way of adjusting mutation in an IEA. Self adaptive step sizes are the defining characteristic of an ES, it is therefore reasonable to question the suitability of using an ES in an IEA. A comparison of the standard deviations of the number of generations that participants

Table 4.6: Correlations between the dependent variables in the evaluation of the effectiveness of the hyperplane-IES

			Spearman's correlation coefficients					
		IES	Generations	Time taken	Final distance	Perceived speed	Ease of use	Control
p-values	Generations	SMM	—	0.787	-0.014	-0.069	-0.012	0.137
		Hyperplane	—	0.774	-0.437	0.580	0.698	0.445
		Dummy	—	0.746	-0.172	0.413	0.228	-0.041
	Time taken	SMM	< 0.001	—	-0.167	0.061	-0.020	-0.142
		Hyperplane	< 0.001	—	-0.123	0.520	0.585	0.411
		Dummy	< 0.001	—	0.015	0.230	0.146	-0.033
	Final distance	SMM	0.950	0.618	—	0.276	0.245	0.020
		Hyperplane	0.033	0.567	—	0.027	-0.088	-0.132
		Dummy	0.423	0.947	—	0.237	0.456	0.265
	Perceived speed	SMM	0.968	0.777	0.192	—	0.704	-0.012
		Hyperplane	0.003	0.009	0.900	—	0.884	0.615
		Dummy	0.045	0.279	0.264	—	0.800	0.585
	Ease of use	SMM	0.957	0.926	0.249	< 0.001	—	0.066
		Hyperplane	< 0.001	0.003	0.683	< 0.001	—	0.532
		Dummy	0.284	0.495	0.025	< 0.001	—	0.671
	Control	SMM	0.524	0.507	0.926	0.955	0.761	—
		Hyperplane	0.029	0.046	0.539	0.001	0.008	—
		Dummy	0.850	0.877	0.211	0.003	< 0.001	—

used to achieve a colour match as presented in Table 4.5 with the average number of generations required by the virtual user to achieve a colour match for various values of p and α as presented in Table 4.4 suggests that the effort involved in fine tuning the parameters of an IEA using a virtual user is not rewarded.

4.5 Conclusion

A simple colour matching task was used to see whether the SMM-IES could be improved by making use of a user's ability to identify images which are particularly unlike their target image. Hyperplanes were introduced to the search space to reduce the number of images presented to the user which were likely to be less desirable than those the user had already rejected.

Providing users with the ability to reject individuals gives the users a feeling of greater control over the algorithm. The hyperplane-IES did not improve the proximities of the final colours to the target colours and in fact the SMM-IES was significantly quicker than the hyperplane-IES and the dummy-IES. This adds support to the conclusions of Yoon and Kim [145] and Takenouchi et al. [126] that the simplest evaluation methods are the most desirable.

The differences between the ways the participants used the IEAs means that the use of a virtual model of user behaviour such as the virtual user used to set the parameters for the hyperplane-IES is unlikely to possess any advantages over the use of the designer's own evaluation of the suitability of the parameter values. It also calls into question the validity of any experiments in which comparisons between algorithms are based solely on data gathered from virtual users such as those of Hornby and Bongard [55] and Tagaki and Pallez. [125].

Chapter 5

Comparison of search spaces and search algorithms

5.1 Introduction

When implementing an IEA to perform some task the nature of the task determines how the phenotypes are represented as genotypes. The form of the genotypes, and the values they are permitted to take, determines the search space. The choice of genotype, and hence search space, is usually determined by convenience of implementation. Takagi [124] identifies a difference between the genotype space, which he refers to as the parameter space, and the psychological space, which exists in a user's mind. For some problems it may be possible to use a search space which better corresponds to the psychological space and hence the requirements of the users. Sugimoto and Honda [121] used multidimensional scaling to create a search space which better approximated human perceptual distances between cartoon faces. Five participants were used to perform a cartoon face matching task in the implementation convenient search space and in the psychologically based search space. The only comparison between the performances of the search spaces was a visual inspection by the authors who concluded that the psychologically based search space produced faces more like the target face. There appears to have been no other work in developing or evaluating perceptually uniform search spaces.

The colour matching task lends itself to a comparison between ease of implementation based and psychologically based search spaces. The CIELAB colour space [2] is a well established psychologically based colour space which is designed to be perceptually uniform. A more convenient colour space to use for the colour matching task is the sRGB colour space [1] developed by Microsoft and Hewlett Packard amongst other companies. It is relatively easy to convert between the CIELAB and sRGB colour spaces which is why the CIELAB colour space was used in the experiment reported in Chapter 4. In this chapter an experiment is reported in which the colour matching task is performed in both the CIELAB and sRGB colour spaces to determine whether using the CIELAB colour space confers any advantage over using the sRGB colour space.

The experiment of Chapter 4 introduced an IEA which made use of user evaluation beyond selecting the single best individual from the population. It was found that the time taken to gather the rejection information was not rewarded with a reduction in the overall time taken to achieve a colour match. It is possible that the approach of using information beyond the choice of the single best individual provides an objective pay-off; it might be that the hyperplane-IES as implemented was ineffective but another algorithm may perform better given the same information at the user interface. There has been some work done in the comparison of IEAs and other biologically-inspired metaheuristic algorithms. Akbal et al. [4] developed a facial composite task to compare the performances of five biologically-inspired metaheuristic algorithms but decided that their work was not rigorous enough to draw any conclusions. Lee and Cho [70] used an image enhancement task to compare an IDE algorithm to an IGA and to a direct input manipulation method and found that participants generally favoured the IDE algorithm for usability. Cheng and Kosorukoff [16] compared what they called human-based GAs to more conventional IGAs using a colour matching task and found that human-based GAs achieved the target colour in less than half the number of generations of the conventional IGAs. In these experiments the interfaces for each of the algorithms compared were different; any differences observed could have been due to the rating method of the interface as opposed to the choice of algorithm.

The colour matching task of Chapter 4 was used to make comparisons between IEAs because all of the participants could be given the same objectives starting from the same initial population. The target colour that the participants were trying to match was visible at all times; such a task shall be referred to here as being *with target*. Breukelaar et al. [10] used the same colour matching task to compare different mutation step size parameters in an IES. Sugimoto and Honda [121] had the target visible at all times in a task in which participants had to attempt to recreate a cartoon face. An experiment in which there is a well defined target but in which that target is not visible to the participant during the evolutionary process shall be referred to here as being *without target.* In the continuing development of both EFIT-V and EvoFIT, without target facial composite tasks were used to test the efficacy of the software [37, 114]. A without target task requires the participant to already know or to memorise a target. The ability of the participants to memorise and/or recall the target adds noise to the data gathered as the participants will vary in the ability to do this making it more difficult to distinguish between the performances of whatever algorithmic design options are being tested. Without target tasks are also more realistic than with target tasks as if a visual representation of the target exists then the use of an IEA is unlikely to be the best way of reproducing that target. If no difference between the performances of the algorithmic design options can be detected when performing a comparison using a without target task then it demonstrates that any differences between the design options are insignificant in comparison to the variation in human ability to complete the without target task.

In this chapter the performances of searches conducted in the sRGB and CIELAB colour spaces using a hyperplane-IEA and a simple IGA are compared using both a with target colour matching task and a without target colour matching task.

5.2 Theory

5.2.1 Colour spaces

Colour spaces are a means of representing colours with simple numerical values. In an RGB colour space each colour is described by the amounts of red, green, and blue light present. The sRGB colour space was designed as a standard means for computers to tell display devices which colours to display and is therefore the simplest colour space to use as the genotype space in the colour matching task. The sRGB colour space is not perceptually uniform to the human visual system; distances between colours in the sRGB colour space do not necessarily correspond to their perceptual difference. The CIELAB colour space was designed such that the Euclidean distance between two colours as represented by points in the CIELAB colour space corresponds to their perceptual difference. A colour in the CIELAB colour space is represented by three values: L^*, a^*, and b^*. L^* is the luminosity value and is a measure of how light a colour is, a^* is a red-green axis with positive values meaning the colour is more red and negative values meaning the colour is more green, and b^* is a yellow-blue axis with positive values meaning that the colour is more yellow and negative values meaning that the colour is more blue. The genotypes can be stored as CIELAB values such that the colour matching task is conducted in the CIELAB colour space as in the previous colour matching experiment. The CIELAB colour space extends beyond the sRGB colour space and therefore it is possible for colours to be generated by the algorithm which cannot be rendered in the sRGB colour space and therefore cannot be displayed on a typical monitor [14]. This is not necessarily due to the physical limitations of the monitor but because of the limitations imposed by the sRGB colour space. When performing the colour matching task in the CIELAB colour space, the colours need to be checked to see if their values are valid, that is, to ensure that the r, g, and b values are in the range $[0, 255]$. When performing the colour matching task, if a genotype is generated which cannot be mapped to the sRGB colour space the genotype is discarded, the parents are returned to the mating pool, and the offspring generation process goes back to selecting parents from the mating pool.

The mapping between the CIELAB colour space and sRGB colour space is relatively simple but involves mapping via the CIEXYZ colour space. The mapping processes for both sRGB to CIELAB and CIELAB to sRGB are given here.

sRGB to CIELAB

The mapping from the sRGB to CIEXYZ colour spaces takes the red, green, and blue values to be in the range $[0, 1]$ so it may be necessary to first linearly scale the values from $[0, 255]$ to $[0, 1]$ by dividing the r, g, and b values by 255. The sRGB to CIEXYZ process starts with a form of inverse gamma correction:

$$C = \begin{cases} \left(\frac{c+0.055}{1.055}\right)^{2.4} & \text{for } c > 0.003928 \\ \frac{c}{12.92} & \text{for } c \leq 0.003928 \end{cases} \tag{5.1}$$

where $c \in \{r, g, b\}$ and $C \in \{R, G, B\}$. Then, the X, Y, and Z values are calculated:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4124 & 0.3576 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9505 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \tag{5.2}$$

The L^*, a^*, and b^* values are calculated from the X, Y, and Z values:

$$\begin{aligned} L^* &= 116 \cdot f_1\left(\tfrac{Y}{100}\right) - 16 \\ a^* &= 500 \cdot \left[f_1\left(\tfrac{X}{95.047}\right) - f_1\left(\tfrac{Y}{100}\right)\right] \\ b^* &= 200 \cdot \left[f_1\left(\tfrac{Y}{100}\right) - f_1\left(\tfrac{Z}{108.883}\right)\right] \end{aligned} \tag{5.3}$$

where

$$f_1(c) = \begin{cases} c^{\frac{1}{3}} & \text{if } c > \left(\frac{6}{29}\right)^3 \\ \frac{1}{3}\left(\frac{29}{6}\right)^2 c + \frac{4}{29} & \text{if } c \leq \left(\frac{6}{29}\right)^3 \end{cases} \tag{5.4}$$

CIELAB to sRGB

The process of mapping from the CIELAB colour space to the SRGB colour space begins with mapping from the CIELAB colour space to the CIEXYZ colour space:

$$\begin{aligned} X &= 95.047 \cdot f_2\left(\tfrac{a^*}{500} + \tfrac{L^*+16}{116}\right) \\ Y &= 100 \cdot f_2\left(\tfrac{L^*+16}{116}\right) \\ Z &= 108.883 \cdot f_2\left(\tfrac{L^*+16}{116} - \tfrac{b^*}{200}\right) \end{aligned} \tag{5.5}$$

where

$$f_2(c) = \begin{cases} c^3 & \text{if } c > \frac{6}{29} \\ \frac{c-16/116}{7.787} & \text{if } c \leq \frac{6}{29} \end{cases} \tag{5.6}$$

Then convert from the CIEXYZ colour space to the sRGB colour space:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 3.2406 & -1.5372 & -0.4986 \\ -0.9689 & 1.8758 & 0.0415 \\ 0.0557 & -0.2040 & 1.0570 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}. \tag{5.7}$$

Then the r, g, and b values are calculated from these R, G, and B values using gamma correction:

$$c = \begin{cases} 12.92 \cdot C & \text{for } c \leq 0.0031308 \\ 1.055 \cdot C^{\frac{1}{2.4}} - 0.055 & \text{for } c > 0.0031308 \end{cases} \tag{5.8}$$

where $C \in \{R, G, B\}$ and $c \in \{r, g, b\}$. Finally, the r, g, and b values are linearly mapped from the range $[0, 1]$ to $[0, 255]$ and rounded to the nearest integer.

5.2.2 The IEAs used

It was observed in Chapter 4 that the self adaptive step size aspect of an IES caused the SMM-IES and the hyperplane-IES to stagnate for several generations. To avoid this problem the hyperplane-IEA used in the experiment reported in this chapter does not use a self adaptive mutation step size but has the mutation parameter set externally by the user. This change reduces the length of the genotypes of the colours from four to three as the genotypes no longer carries any information about mutation step size. The removal of the mutation step size from the genotype means that the IEA used is more accurately categorised as an IGA than as an IES. The hyperplane-IEA used in this chapter is therefore referred to as the *hyperplane-IGA*. The other IEA used is a simple real valued IGA referred to as the *simple IGA*. The representation of the simple IGA is the same as that used for the hyperplane-IGA. Rather than using a single preferred individual to be the sole parent of the following generation the simple IGA allows other individuals to be selected as parents as well.

In the simple IGA each individual in the following generation has two parents and each pair of parents produces only one child. Eight new individuals are needed to fill the next generation (as the preferred individual from the previous generation is carried through to the next generation). It follows that a mating pool of sixteen parents is required.

In the simple IGA stochastic universal sampling is used to select parents to go into the mating pool. The simple IGA follows Frowd's method [34] and allows only three levels of selection: preferred, selected, and not selected. When building the sampling wheel all of the selected individuals are assigned equal sized wedges except the preferred individual which is assigned a double sized wedge.

Once the pool has been filled, parents are pulled out and paired at random. Each pair creates one offspring using *uniform crossover* (a process in which each gene in the offspring has an equal chance of taking its value from either parent). After recombination the new individual is mutated.

Both the hyperplane-IGA and the simple IGA use the same mutation operator: Gaussian addition. The new value of the n-th gene of the new individual is given by

$$g'_n = g_n + m \cdot s \cdot N\left(0,1\right) \tag{5.9}$$

where g_n is the pre-mutated gene value, $m \in [0,1]$ is the mutation factor set by the user on the interface, s is the scaling factor, and $N\left(0,1\right)$ is a random number from the Gaussian distribution with a mean of 0 and a standard deviation of 1. The value of s was set to 45 when the sRGB colour space was used and 18 when the CIELAB colour space was used.

5.3 Method

5.3.1 The user interfaces

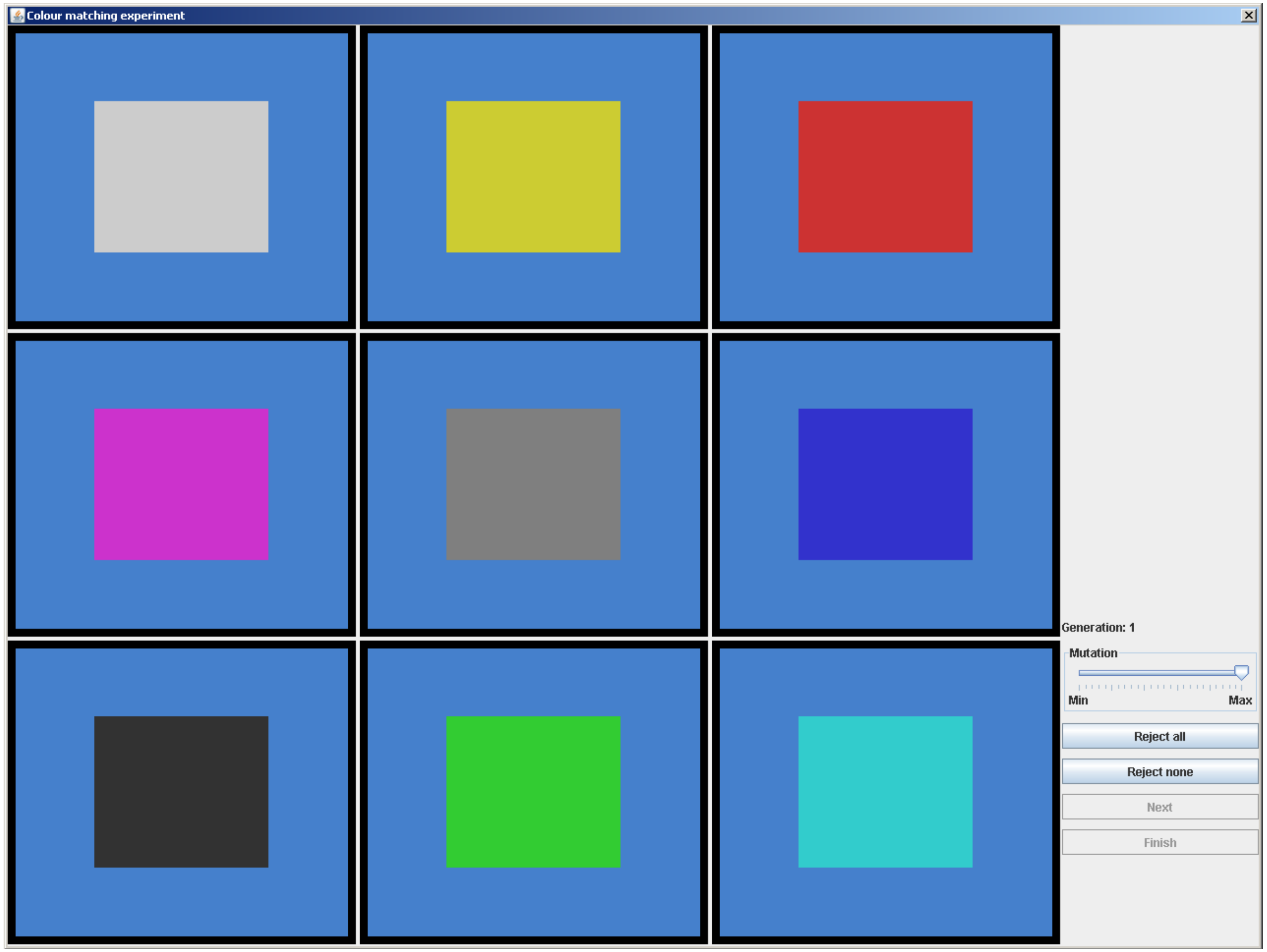
The interface used in the with target runs was nearly identical to the interface for the hyperplane-IES in the previous colour matching experiment, that is, the participants would choose a simple preferred colour from the nine colours on display and reject any colours that were particularly unlike the target colour. A slider was added to the interface to provide the means of adjusting the mutation parameters as it was necessary that the mutation parameter of both the hyperplane-IGA and the simple IGA could be adjusted by the participants during the experiment.The interface is shown in Figure 5.1. As this experiment focused on the underlying algorithms of the IEAs and not the interfaces the only difference between interfaces was due to whether the task was with target or without target.

In the runs performing the without target task the participants had to memorise the target colour and then try to achieve a colour match for the memorised colour. At the start of each run a colour panel filled with the target colour was displayed on the monitor for 10 seconds. The target colour was not shown to the participants during a run and so the interface used for the with target task had to be modified slightly. The without target interface is shown in Figure 5.2.

5.3.2 Test set-up

Twenty-four participants were used in this experiment. The participants were mainly postgraduate students in the School of Physical Sciences at the University of Kent, the remainder were staff or undergraduate students. As the task was cognitively simple, no knowledge was required except a very basic level of computer literacy. However, as with the previous colour matching experiment, it was important that participants were not colour blind.

This experiment was designed to compare two different colour spaces (CIELAB and sRGB) and two different algorithms (hyperplane-IGA and the simple IGA) for a colour matching task both with the target colour present and without.



Generation: 1

Figure 5.1: Screenshot of interface for the with target colour matching task in the algorithm/space experiment

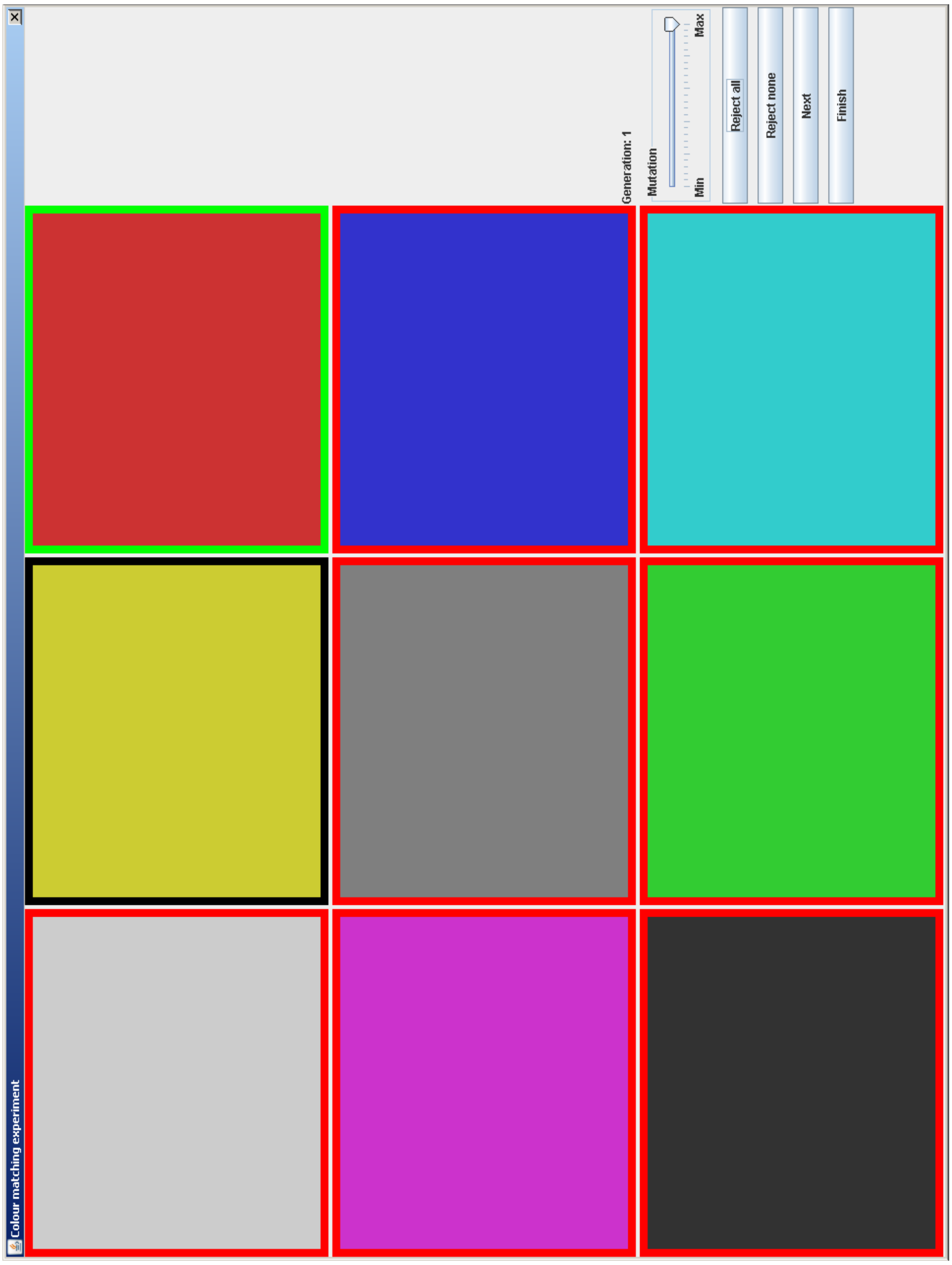


Figure 5.2: Screenshot of interface for the without target colour matching task in the algorithm/space experiment

At the start of the experiment the participants were read a script detailing what the experiment consisted of. There then followed a practice run performing the with target colour matching task followed by a practice run performing the without target colour matching task. Each participant completed eight runs in the recorded part of the experiment, one run for each combination of colour space and IEA performing the with target colour matching task followed by one run for each combination of colour space and IEA performing the without target colour matching task.

The same target colours were used as in the colour matching experiment of Chapter 4: a cyan for the practice run for both the with target and without target tasks, and an orange and a dark green for the recorded experiment. Half of the participants used the orange for the with target task and the dark green for the without target task, the other half of the participants used the dark green for the with target task and the orange for the without target task. The initial population for each task was the same as that used in Chapter 4.

5.3.3 Data gathered

Four types of objective data were collected: the time taken for the participant to complete a run, the number of generations they took, the accuracy of the final colour match (the distance of the participants' final colours to the target colour) as measured in the CIELAB colour space, and the accuracy of the final colour match as measured in the sRGB colour space. Testing the experiment revealed that it was difficult to compare the colour spaces and IEAs subjectively because they all had the same interface and thus it was hard to remember which run seemed fastest or provided the most control. Also, the results of the first colour matching experiment suggested that the subjective data depended more upon the appearance of the interface than the behaviour of the IEA. As a consequence of these observations no subjective data were gathered in this experiment.

5.4 Results

5.4.1 Comparisons between the colour spaces and the IGAs

The means and standard deviations of the measured variables (number of generations, time taken, final distance to the target in the CIELAB colour space, and final distance to the target in the sRGB colour space) are given in Table 5.1 for the with target task and Table 5.2 for the without target task. Each of the measured variables were transformed using ART (see Section 2.2.4) and subjected to two-way ANOVA having two colour spaces (CIELAB and sRGB) and two IEAs (hyperplane-IGA and the simple IGA). The ART two-way ANOVA was performed for the with target task (Table 5.3) and the without target task (Table 5.4). It can be seen that the main effect of colour space was significant for the number of generations on the with target task, with the CIELAB colour space requiring fewer generations to achieve a colour match. The main effects of colour space and IEA were not significant for any of the remaining measured variables for either the with target task or the without target task. The interaction between colour space and IEA was not significant for either the with target task or the without target task for any of the measured variables.

5.4.2 Correlations between the measures

The Spearman's correlation coefficients between the dependent variables were calculated for both the with target (Table 5.5) and without target (Table 5.6) tasks. In each case the correlations were calculated for each combination of colour space and IGA of each task. In both the with target task and the without target task there was a very strong correlation between the final distances in the CIELAB colour space and those the sRGB colour space. There was also a strong correlation between time taken and the number of generations. It was expected that the final distances would be strongly correlated as whilst there is a noticeable difference between the CIELAB and sRGB colour spaces on the large scale, on a local scale around any particular colour they are very similar. The strong correlations between number of generations and time taken were also expected. On the sRGB colour space hyperplane-IGA without target runs there was a weak but significant negative correlation between the number of generations and the final distance in the CIELAB space suggesting that those participants who had a better recollection of the target colour (and therefore had a shorter final distance) would require more generations to achieve a satisfactory colour match. This was the only significant correlation found between number of generations and final distance in the CIELAB colour space so no general correlation between final distance and number of generations should be inferred.

5.4.3 Observations of user behaviour

Table 5.1: Means (standard deviations) of the dependent variables in the comparison of the hyperplane-IGA and simple IGA algorithms and the sRGB and CIELAB colour spaces for the with target colour matching task

Algorithm	Colour space	Generations	Time taken	Final distance CIELAB	Final distance sRGB
Hyperplane	CIELAB	16.9 (17.8)	185s (206s)	5.31 (3.07)	12.0 (7.80)
Simple	CIELAB	13.5 (7.62)	148s (78.3s)	5.67 (3.61)	15.3 (11.9)
Hyperplane	RGB	23.5 (23.3)	221s (167s)	4.97 (3.18)	12.0 (9.29)
Simple	RGB	19.0 (11.9)	197s (113s)	5.24 (3.32)	13.2 (10.5)

Table 5.2: Means (standard deviations) of the dependent variables in the comparison of the hyperplane-IGA and simple IGA algorithms and the sRGB and CIELAB colour spaces for the without target colour matching task

Algorithm	Colour space	Generations	Time taken	Final distance CIELAB	Final distance sRGB
Hyperplane	CIELAB	8.25 (3.29)	84.9s (42.3s)	18.5 (10.1)	46.7 (26.8)
Simple	CIELAB	8.04 (3.48)	85.3s (50.5s)	17.0 (7.53)	42.6 (21.2)
Hyperplane	RGB	8.75 (4.39)	93.2s (62.6s)	19.5 (6.48)	49.7 (18.3)
Simple	RGB	9.29 (5.10)	102s (72.2s)	18.8 (9.81)	44.5 (20.9)

Table 5.3: ART with two-way ANOVA for the dependent variables in the comparison of the hyperplane-IGA and the simple IGA and the sRGB and CIELAB colour spaces for the with target colour matching task

	Colour space		Search algorithm		Interaction	
Variable	$F(1,92)$	p-value	$F(1,92)$	p-value	$F(1,92)$	p-value
Generations	6.953	0.010	0.445	0.112	0.048	0.828
Time taken	3.551	0.063	0.008	0.931	0.028	0.867
Final distance CIELAB	0.263	0.609	0.141	0.708	0.021	0.886
Final distance sRGB	0.266	0.607	0.208	0.650	0.004	0.948

Table 5.4: ART with two-way ANOVA for the dependent variables in the comparison of the hyperplane-IGA and the simple IGA and the sRGB and CIELAB colour spaces for the without target colour matching task

	Colour space		Search algorithm		Interaction	
Variable	$F(1,92)$	p-value	$F(1,92)$	p-value	$F(1,92)$	p-value
Generations	0.264	0.609	0.008	0.931	0.354	0.553
Time taken	0.101	0.751	0.017	0.897	0.084	0.772
Final distance CIELAB	0.892	0.348	0.879	0.351	0.318	0.574
Final distance sRGB	1.063	0.305	1.191	0.278	0.424	0.517

Table 5.5: Correlations between the dependent variables: with target

		Colour space and IGA	Spearman's correlation coefficients			
			Generations	Time taken	Final distance CIELAB	Final distance sRGB
p-values	Generations	CIELAB hyperplane	—	0.786	-0.091	-0.083
		CIELAB simple IGA	—	0.621	0.187	0.247
		sRGB hyperplane	—	0.695	0.175	0.164
		sRGB simple IGA	—	0.764	-0.57	0.022
	Time taken	CIELAB hyperplane	< 0.001	—	-0.135	-0.134
		CIELAB simple IGA	0.001	—	0.178	0.265
		sRGB hyperplane	0.002	—	0.047	0.036
		sRGB simple IGA	< 0.001	—	-0.226	-0.177
	Final distance CIELAB	CIELAB hyperplane	0.673	0.529	—	0.958
		CIELAB simple IGA	0.381	0.403	—	0.966
		sRGB hyperplane	0.412	0.828	—	0.957
		sRGB simple IGA	0.792	0.287	—	0.965
	Final distance sRGB	CIELAB hyperplane	0.700	0.531	< 0.001	—
		CIELAB simple IGA	0.245	0.210	< 0.001	—
		sRGB hyperplane	0.445	0.869	< 0.001	—
		sRGB simple IGA	0.918	0.408	< 0.001	—

Table 5.6: Correlations between the dependent variables: without target

			Spearman's correlation coefficients			
		IES	Generations	Time taken	Final distance CIELAB	Final distance sRGB
p-values	Generations	CIELAB hyperplane	—	0638	0.003	0.015
		CIELAB simple IGA	—	0.694	-0.027	-0.039
		sRGB hyperplane	—	0.732	-0.483	0.167
		sRGB simple IGA	—	0.737	-0.188	-0.131
	Time taken	CIELAB hyperplane	< 0.001	—	0.177	0.050
		CIELAB simple IGA	< 0.001	—	0.077	0.106
		sRGB hyperplane	< 0.001	—	-0.286	-0.074
		sRGB simple IGA	< 0.001	—	-0.077	-0.087
	Final distance CIELAB	CIELAB hyperplane	0.990	0.408	—	0.847
		CIELAB simple IGA	0.901	0.722	—	0.816
		sRGB hyperplane	0.017	0.175	—	0.731
		sRGB simple IGA	0.380	0.719	—	0.877
	Final distance sRGB	CIELAB hyperplane	0.945	0.818	< 0.001	—
		CIELAB simple IGA	0.856	0.621	< 0.001	—
		sRGB hyperplane	0.434	0.731	< 0.001	—
		sRGB simple IGA	0.544	0.685	< 0.001	—

Every participant completed all of the with target runs trying to match one single target colour and all of the without target runs matching a different single target colour. The same initial population of colours was used for every run. These experimental decisions were intended to eliminate noise in the data due to randomness in the initial population or any effects from variation in the target colours. The user interface was also identical for all runs of the experiment, with the exception of the difference between the with target and without target tasks. These factors combined meant that the participants perceived that they were performing exactly the same task, with the same starting population, target colour, and algorithm four times in a row. It was observed that sometimes participants would change their behaviour between runs in an attempt to achieve a quicker colour match. This change in behaviour was most apparent when participants chose to select or reject different colours when evaluating the initial population from one run to the next. From comments made by the participants it was apparent that a few of them developed models of what was happening in the algorithm and made decisions based on their expected behaviour of the algorithm as opposed to their instructions. In such cases a colour other than the perceived closest match may be preferred because the participant thought that the selection would lead to a quicker colour match. These behaviours undoubtedly contributed noise to the data collected.

Many of the participants did not make effective use of the mutation slider to adjust the degree of mutation in the algorithm — the participants would fail to reduce the mutation value when the IEA produced colours closer to the target colour and so there would be a problem where the same individual was the best in the population for several generations. This is probably the main cause of the large variances in the measured variables. This problem can be addressed by having the mutation slider decrement slightly each generation automatically. Ideally this would cause the algorithm to converge on a good colour match but it could also cause users to become more aware of the mutation slider because they can see the effect of altering its value.

Participants who did make frequent use of the slider were generally making difficult minute adjustments at the low value end of the slider. This problem can be addressed by using a power or exponential scaling on the values input using the slider. For example, if the values input by the slider are in the range $[0, 1]$, squaring these input values before setting the mutation value would change the effect of the slider on the IEA; a change in the slider position at low slider values would result in a smaller change in the mutation value than the same change at larger slider values.

5.5 Conclusion

A simple colour matching task was used to compare the performances of two IEAs which used identical interfaces: a simple IGA and a hyperplane-IGA. The same task was used to compare two search spaces, the convenient sRGB colour space and the perceptually uniform CIELAB colour space.

The lack of significant difference between the hyperplane-IGA and the simple IGA suggests that for a unimodal subjective fitness function neither of these algorithms has an advantage over the other. This lends weight to the idea that the differences between algorithms reported in [16] and [70] are due to differences between the interfaces and the rating method rather than the underlying algorithms.

The lack of difference between the colour spaces with regards to the final distances to the target colour, particularly for the with target, task does not support the assertion of Sugimoto and Honda [121] that using a more perceptually uniform search space leads to a better match to the target.

The significant difference of the number of generations (and 'marginally significant' difference of time taken) between the colour spaces for the with target task suggests that the use of a psychologically based search space can help the IEA to attain a solution more quickly. The lack of differences between the colour spaces for the without target task, however, suggests that for more realistic tasks the use of a psychologically based search space makes no difference. This finding is taken to indicate that the effort of constructing a psychologically based search space is only warranted if the search space is easy to construct (or already exists) or if the parameters in the implementation based search space are particularly non-linear with respect to the psychologically based search space. A sensible compromise may be to use non-linear scaling to render non-linear variables approximately linear.

Chapter 6

Comparison of interactive methods for contrast enhancement of images

6.1 Introduction

Over the past decade digital cameras have become ubiquitous. Their low cost has led to virtually all mobile telephones, themselves having become commonplace, incorporating a digital camera. This prevalence of digital cameras, combined with the convenience and negligible cost of capturing photographs using digital cameras in comparison to film cameras, has enabled casual photography to become a part of daily life [26].

The cameras used for casual photography, low end dedicated cameras and particularly cameras included in mobile telephones, are prone to noise. This noise is due to the drive to increase the resolution of digital cameras without increasing the size of sensor arrays [15]. Individual sensors have therefore become smaller. This has led to greater statistical variation in the number of photons detected by neighbouring sensors, particularly in low light conditions. This variation in the number of photons detected leads to Gaussian noise in the captured photographs.

Another aspect of mobile telephone cameras is the general lack of control offered to the users in terms of settings. In order to make the cameras easy and convenient to use all settings, such as exposure time and ISO rating, are controlled by software in the camera. Low end dedicated digital cameras offer more control over the settings, however, adjusting these settings can be a laborious task and users may be disinclined to take the time required to do so. For this reason dedicated cameras offer software control of settings in the same manner as mobile telephone cameras.

The settings, and the software that controls them, are designed by experts and generally achieve their goal of enabling users to take satisfactory photographs using the camera. Sometimes, however, a photograph can be improved with the application of relatively simple image processing methods, most notably contrast enhancement. The nature of digital images means that such improvements are possible using a personal computer or even a mobile telephone. Photographs captured by casual users tend to be for the purpose of recording events as opposed aesthetic reasons [26]. It is unlikely that such users would be inclined to spend time learning how to use software to enhance images or spend time actually enhancing them. This consideration suggests that the use of an IEA may be an appropriate approach for enabling casual users to enhance photographs.

When a user enhances an image with image processing they typically have some goal in mind. An example goal is "make the people in the foreground easier to see". The goal may require that more than one image process be applied to the image. Maybe it is necessary to find the best ordering of a number of processes or maybe the order of the processes can be pre-set and it is the input values for the processes that are adjusted. Within the context of EAs the order of the processes or the input values required can be referred to as phenotypes. If the order of processes or the input values are developed by some other method, say by direct input from the user, or the phenotypes are to be exported for use elsewhere, then the term phenotype is no longer appropriate. Here the term *recipe* is introduced and thus in this experiment the participants develop recipes with which to process the photographs.

EAs have been used previously to develop contrast enhancement recipes. Hoseini and Shayesteh [56] used a combination of ant colony optimisation, GAs, and simulated annealing to develop mapping functions for greyscale photographs. Munteanu and Rosa [84] used a GA to optimise a local contrast enhancement method. Subjective evaluation showed that the resulting recipes were shown to be an improvement over contrast stretching and histogram equalisation. Shyu and Leou [110] used a GA to optimise the weights for combining four mapping functions and their five input parameters to create a single transformation for use on colour images. Verma et al. [137] used ant colony optimisation to optimise the parameters of a mapping function based on sigmoid transformations. Gorai and Ghosh [48] used PSO to optimise a local contrast enhancement process.

The EAs above all use statistical image data of the processed images as fitness functions. Automatic approaches cannot know which parts of an image a user wishes to emphasise or what degradation they are willing to tolerate in other parts of the image to achieve this. There is, therefore, scope for the use of interactive approaches.

To assess whether the IEA approach may be appropriate to the contrast enhancement problem, the IEA suitability questions of Section 2.2.3 are addressed:

- *Is the subjective fitness function unimodal in the search space?* Unlikely, there are likely to be parts of the search space which produce photographs a user considers poor between parts which produce photographs they consider good.
- *What other approaches to the problem are available?* The normal approach is for a piece of software to provide a number of image enhancement processes which the user makes use of if they know the processes are available. It is up to the user to apply them in the appropriate order and they may find that input values they have used for a process are unsuitable only after they have applied another process. This can lead to a lot of doing, undoing, and redoing. An alternative to this is to provide a sequence of processes in which the users can adjust the input values for each stage of the process on one panel. Every time an input value is adjusted the whole process is applied to the original photograph, thus eliminating the need for repeated undoing and redoing.
- *Is the search space large?* As with the noise removal problem of Chapter 3 it is difficult to be sure. Both the compound (see Section 6.2.2) and the piecewise intensity transfer (see Section 6.2.3) contrast enhancement processes take nine input values to determine the output of the processes.The smallest change in an input value that produces a noticeable difference in the processed photograph depends on the values of some, or most, of the other input values.

As expected, consideration of the IEA suitability questions indicates that there is sufficient justification for the use of an IEA in finding optimal input values for a contrast enhancement process and indeed IEAs have been used in the contrast enhancement of images. Tokuda et al. [130] compared the use of an IGA to set the shape of gamma adjustment functions for enhancing grey-scale images to a manual approach. It was found that participants preferred the images that had been enhanced using mapping functions developed using the IGA. It was also found that the participants preferred using the IGA to the manual method. Ma and Takagi [77] used interactive genetic programming (IGP) and a manual method to develop recipes that were a composite of various known processes such as gamma adjustment, sigmoid transformation, and image sharpening. It was found that participants preferred images that were processed using recipes developed using IGP. Jung et al. [61] developed an IGA for adjusting brightness, contrast, and colour balance on photographs on mobile telephones. Lee and Cho [70] built upon this work and compared interactive differential evolution (IDE) to an IGA and a manual method. It was found that participants preferred using the IDE and IGA methods to the manual method. A general discussion on the use of IEAs in image processing can be found in Jakša et al. [60].

In this experiment two means of finding optimal input values for recipes are compared: the simple IGA introduced in Chapter 4 and a direct interface which enables users to manipulate input values directly using a number of sliders. Two different image processing functions are compared also: a compound process which uses a combination of common image processes and a piecewise intensity transfer function.

6.2 Theory

6.2.1 The noise treatment process

In some cases, particularly in underexposed photographs taken in low light conditions, the Gaussian noise in a photograph is quite visible before any form of contrast enhancement is applied, such as the case with the Books photograph (Figure 6.7 (c)) used in this experiment. In other cases though, it is not particularly visible unless contrast stretching of the noisy (usually dark) areas is performed on the photograph, as is the case with the Atlas photograph (Figure 6.7 (a)). Whilst noise treatment may not appear to be necessary before contrast enhancement processes are performed it may become apparent that some noise treatment is necessary after other enhancement processes have been applied to the photograph. Five filtering approaches were considered for treating Gaussian noise in the images before enhancement: bilateral filtering [131], Vijaykumar filtering [138], Non-local means filtering [13], foveated non-local means filtering [33], and colour block matching and 3-dimensional filtering [23].

Bilateral filtering was introduced by Tomasi and Manduchi in 1998 [131]. Like Gaussian filters, pixels that are geometrically close to the pixel of interest are given greater weight than those farther away. Unlike Gaussian filters, however, the similarities of the values of other pixels in the filtering window are also taken into account. Bilateral filtering was designed to have an improved performance over Gaussian filtering at boundaries between areas of different pixel intensities of an image. For example, consider a vertical boundary separating a white region (pixel value 255) and a black region (pixel value 0). Pixels on both sides of the boundary would be a medium grey after Gaussian filtering. As bilateral filtering also accounts for similarities of pixel values the white and black pixels would have very little effect on each other even if they are in close proximity, thus preserving the boundary between the regions.

The filter referred to here as the Vijaykumar filter was introduced by Vijaykumar et al. in 2010 [138]. Like bilateral filters, the Vijaykumar filter uses both the proximity and similarity of pixel values when assigning a new value to the pixel of interest. The Vijaykumar filter replaces the value of the pixel of interest with the mean of those pixel values in the filtering window whose values differ from that of the pixel of interest by less than a certain threshold; the threshold being defined by a smoothing factor and an estimate of the noise level of the image set by the user. If there are too few similar pixels in the window, the window is increased in size and the mean is calculated from similar pixel values taken from over a wider area.

Non-local means (NLM) filtering was introduced by Buades el al. in 2005 [13]. It can be thought of as a generalisation of bilateral filtering. As with bilateral filtering the proximity of other pixels in the filtering window partially determines their weight when calculating the new value of the pixel of interest. Similarity of pixel values is also used to help determine the new value of the pixel of interest. In bilateral filtering it is the similarity between the value of a pixel and that of the pixel of interest that determines its weight; however, with NLM filtering it is the similarity of the regions around pixels that determines the weight of the pixels in the filtering window. The idea is behind NLM filtering is that similarity between regions of an image is more likely to be due to repeated patterns in the underlying image data than being due to noise.

Foveated NLM filtering was introduced by Foi and Boracci in 2012 [33]. As the name suggests foveated NLM filtering is an extension to NLM filtering which draws upon the foveation aspect of the human visual system. In the human visual system, visual acuity is at its greatest in the direction the eye is looking. In foveated NLM filtering the regions being compared undergo a process which blurs the regions in a manner such that there is no blurring at the centre of the regions but blurring becomes more pronounced at the edges.

The colour block-matching and 3-dimensional (CBM3D) filter was introduced by

Dabov et al. in 2007 [23]. The CBM3D filter is the colour extension of the block-matching and 3-dimensional (BM3D) filter introduced in the same paper. Like the NLM filters the BM3D filter uses region matching, or in the terminology of [23], block matching. The BM3D filter, however, is far more involved than NLM filtering or any of the other filters considered. At a basic level the filtering process can be split into two parts. In the first part, similar blocks are stacked, undergo a 3D wavelet transform, hard thresholding (to remove low amplitude components, which are assumed to be noise), and an inverse transformation. The filtered blocks are added to a first estimate filtered image with overlapping blocks being aggregated. In the second part the process is similar but Wiener filtering, using the first estimate as the estimate of the uncorrupted image, is performed instead of hard thresholding. As the BM3D filter only works on monochrome images the CBM3D filter first needs to transform the input image into a colour space which separates luminosity from chromaticity. BM3D filtering is performed on the luminosity channel and then the final image is assembled.

Buades et al, [13] used objective image measures to demonstrate that NLM filtering could outperform bilateral filtering on Gaussian noise. Similarly, Foi and Boracci [33] demonstrated that foveated NLM filtering could marginally outperform NLM filtering. Shao et al. [107] found that BM3D filtering outperformed NLM filtering, from which it is inferred here that BM3D filtering also outperforms foveated NLM filtering. All of these comparisons were made using objective image measures, which as was demonstrated in Chapter 3 are not always a reliable measure of filter performance. Whilst no comparison between Vijaykumar filtering and BM3D filtering could be found the long execution time of Vijaykumar filtering (over 2 seconds for a 256×256 image) made it less appealing than CBM3D filtering (for which execution time for an image the same size was less than one second). A visual inspection of the performances all of the filters confirmed that CBM3D filtering demonstrates the best performance of the five filtering methods considered.

There are a number of parameters that can be adjusted in the CBM3D filter. Most of them have little effect on the result of applying the filter, and changing some of the inputs has only a detrimental effect. Only two of the parameters are suitable for optimisation: σ, the estimated standard deviation of the noise in the image, and the hard thresholding parameter λ_{3D} used in the first part of the filtering process. In visual terms, altering λ_{3D} has the same effect as altering σ. This means that the CBM3D filter is effectively controlled by one input value: σ.

Low values of σ mean that the filter has less effect on noise in the photograph but also less of the 'cartooning' effect which can result from applying the filter. Larger values of σ mean that the noise is less visible but the cartooning effect is more pronounced.

6.2.2 The compound contrast enhancement process

A common approach to applying a number of image enhancement processes to an image is to apply them sequentially. This is what his happening when one uses typical photo editing software such as Photoshop [3]; one image enhancement process is applied after another. Sequential processing can lead to obvious quantisation effects due to the rounding of transformed values performed at the end of every process. Also, it is sometimes desirable to apply a process but in a subtle manner such that the process does not have an excessive effect on an image. The compound process used in this experiment accounts for both of these factors by using a weighted combination of the outputs of the image processes used. The compound contrast enhancement process is a combination of four common image enhancement processes: histogram equalisation, local contrast enhancement, gamma adjustment, and sigmoid transformation.

It is considered desirable that all of the contrast enhancement processes are performed in a colour space which separates luminance from chrominance. This is desirable because performing the contrast enhancement procedures used in this work in the sRGB colour space can lead to chromatic distortions in the processed images. There are many colour spaces that separate luminosity from chromatic components of colour. The HSV colour space [47], the CIELAB colour space [2], or any one of the colour spaces designed for use in televisual broadcast: YUV, YIQ, or YCbCr [14] could have been chosen. Preliminary testing with histogram equalisation in the luminosity channel of each of the colour spaces revealed that the HSV colour space could introduce blocking artefacts and over-saturate colours. The outputs of the CIELAB, YUV, YIQ, and YCbCr colour spaces were very similar. As the sRGB to CIELAB colour space conversions took much longer than the sRGB to YUV, YIQ, and YCbCr colour spaces CIELAB was not used. The YIQ colour space was chosen at random from the remaining colour spaces. The YIQ colour space has very simple transformation functions between the sRGB colour space and the YIQ colour space:

$$\begin{aligned} Y &= 0.299R + 0.587G + 0.114B && (6.1) \\ I &= 0.595716R - 0.274453G - 0.321563B && (6.2) \\ Q &= 0.211456R - 0.522591G + 0.311135B && (6.3) \end{aligned}$$

$$\begin{aligned} R &= Y + 0.9563I + 0.6210Q && (6.4) \\ G &= Y - 0.2721I - 0.6474Q && (6.5) \\ B &= Y - 1.1070I + 1.7056Q && (6.6) \end{aligned}$$

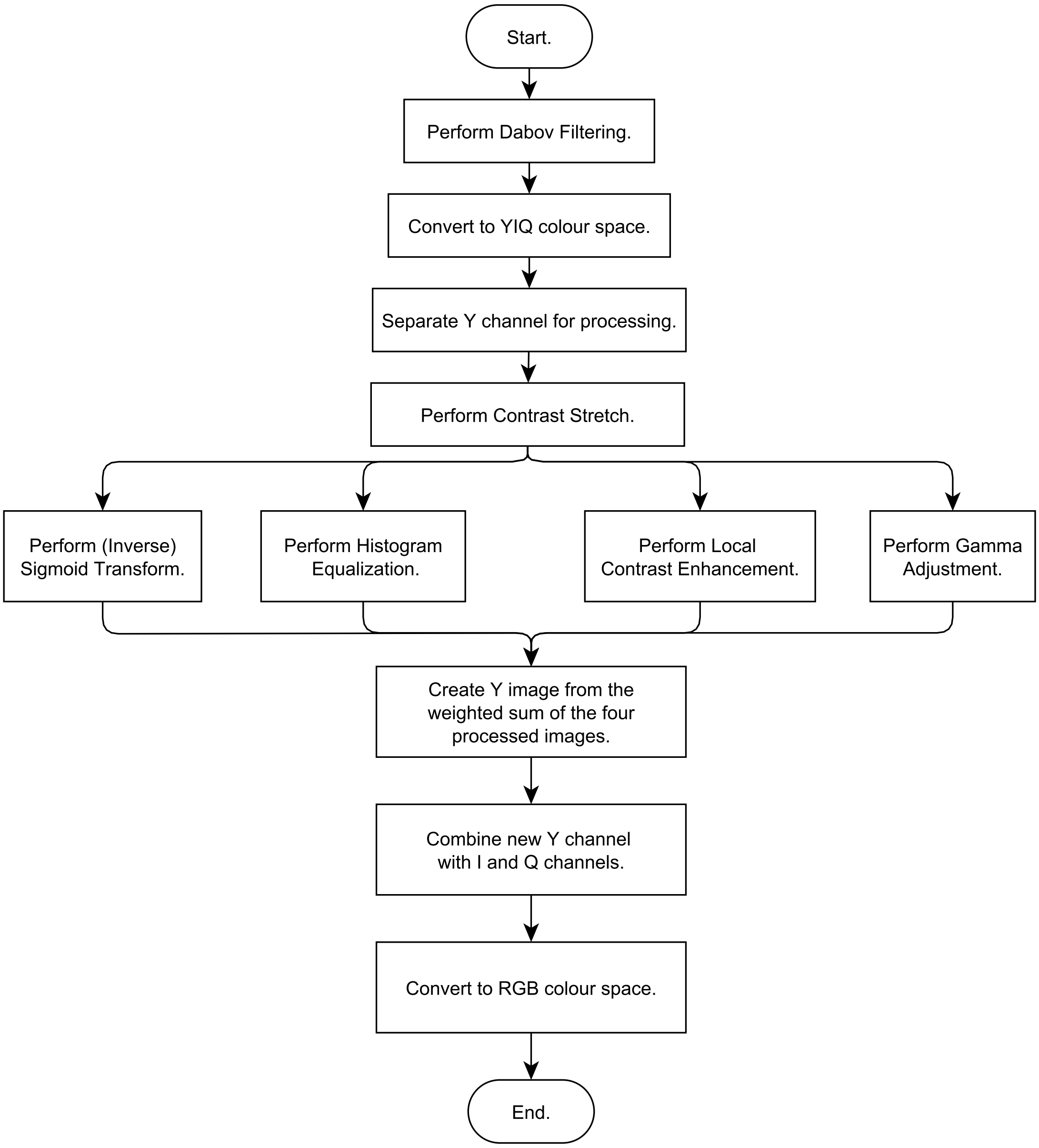
The r, g, and b values in the sRGB colour space lie in the range $[0, 255]$. The R, G, and B values in Equations 6.3 and 6.6 are expressed in the range $[0, 1]$. The pixel values need to be linearly scaled before being converted to the YIQ colour space and after conversion from the YIQ colour space. The Y values lie in the range $[0, 1]$ so there is no need to scale the Y values for the contrast enhancement processes.

The compound image enhancement process starts with the noise treatment process. The noise treatment process is performed first because noise can be accentuated as well as the contrast in contrast enhancement processes, particularly when using local contrast enhancement. The next step is the colour space conversion from the sRGB colour space to the YIQ colour space. The Y channel data are treated as a monochrome image in the contrast enhancement processes. The compound process begins with contrast stretching in order to make best use of the gamma adjustment and sigmoid transformation processes.

Each process takes the contrast stretched Y image as an input and outputs a processed Y image. These outputs are then recombined using a normalised weighted sum of the images to produce a new Y image. This new Y image is used to replace the Y channel in the original YIQ image. This YIQ image is transformed to the sRGB colour space and the result is the processed image. The full compound process is illustrated in Figure 6.1.

A global contrast enhancement process is one in which each new pixel value depends only upon its current value and not upon its position or the values of surrounding pixels. Contrast stretching, gamma adjustment, sigmoid transformation and histogram equalisation are all global contrast enhancements. A local contrast enhancement is one in which a new pixel value depends on both its current value

Figure 6.1: The compound process


Start.
Perform Dabov Filtering.
Convert to YIQ colour space.
Separate Y channel for processing.
Perform Contrast Stretch.
Perform (Inverse) Sigmoid Transform.
Perform Histogram Equalization.
Perform Local Contrast Enhancement.
Perform Gamma Adjustment.
Create Y image from the weighted sum of the four processed images.
Combine new Y channel with I and Q channels.
Convert to RGB colour space.
End.

and the values of the surrounding pixels. Local contrast enhancement is, as the name suggests, a local process.

Contrast stretching

Contrast stretching is useful when the minimum pixel value in an image is greater than the minimum possible value x_{min} and/or the maximum pixel value is less than the maximum possible value x_{max}. Performing contrast stretching on an image ensures that the image uses the largest possible contrast range. The most basic form of contrast stretching maps the pixel values according to

$$y = \frac{x - x_{\min}}{x_{\max} - x_{\min}}. \tag{6.7}$$

This particular form of contrast stretching is not robust; a single full intensity pixel in an otherwise dark image prevents this form of contrast stretching from having an effect [113]. A remedy to this problem includes setting a stretching window such that a certain proportion of the pixels are mapped to the maximum and minimum values. The simple form was chosen over the more effective method because it was applied only as a preliminary step to improve the effectiveness of the gamma adjustment, sigmoid transform, and the piecewise intensity transfer processes. Contrast stretching was not intended to be a significant part of the image enhancement processes.

Gamma adjustment

This contrast enhancement process was originally derived due to the need to correct for the non-linearity between input voltage and display amplitude in cathode ray tube (CRT) monitors [101]. Gamma adjustment is used to accentuate the contrast in darker regions at the expense of diminishing the contrast in lighter regions (for $\gamma < 1$) or accentuate the contrast in lighter regions at the expense of diminishing the contrast in darker regions (for $\gamma > 1$) [14]. Gamma adjustment works by mapping the pixel values according to

$$y = x^{\gamma} \tag{6.8}$$

where x and y are the input and output values respectively and are scaled to the range $[0, 1]$. Gamma adjustment has only one parameter; γ.

Sigmoid transform

A sigmoid function is a function that has an 's' shape and a pair of horizontal asymptotes as $x \to \pm\infty$. Normally, the term 'sigmoid function' refers to the logistic function [82]:

$$y = \frac{1}{1 + e^{-x}} \tag{6.9}$$

To be of use in contrast enhancement the sigmoid function should pass through the points $(0, 0)$ and $(1, 1)$. For the sake of mathematical convenience we shall state that it should pass through the point (1/2,1/2). The sigmoid function of Equation 6.9 needs to be generalised in order to be able to satisfy these conditions. this generalised sigmoid function is written here as

$$y = m\left(\frac{1}{1 + e^{-s(x-b)}} + c\right) \tag{6.10}$$

where s determines the degree to which the sigmoid function is linearly scaled along the x-axis, m is the linear scaling on the y-axis, b is translation along the x-axis, and c is the translation along the y-axis. There are three points through which the function must pass: $(0, 0)$, $(\frac{1}{2}, \frac{1}{2})$, and $(1, 1)$ and four parameters that can be adjusted to achieve this: s, m, b, and c. One of the parameters can therefore be used to define the slope of the sigmoid function. For the sake of convenience, s is chosen as the defining parameter. We need to write Equation 6.9 in terms of s, x, and y Substituting $y = 0$, $x = 0$ into Equation 6.10 yields

$$m\left(\frac{1}{1 + e^{sb}} + c\right) = 0. \tag{6.11}$$

Discarding the trivial solution $m = 0$ we obtain

$$c = -\frac{1}{1 + e^{sb}} \tag{6.12}$$

so we now have

$$y = m\left(\frac{1}{1 + e^{-s(x-b)}} - \frac{1}{1 + e^{sb}}\right). \tag{6.13}$$

Substituting $y = \frac{1}{2}$, $x = \frac{1}{2}$ into Equation 6.13 yields

$$\frac{1}{2} = m\left(\frac{1}{1 + e^{-s\left(\frac{1}{2}-b\right)}} - \frac{1}{1 + e^{sb}}\right) \tag{6.14}$$

and substituting $y = 1$, $x = 1$ into Equation 6.13 yields

$$y = m\left(\frac{1}{1+e^{-s(x-b)}} - \frac{1}{1+e^{sb}}\right). \tag{6.15}$$

Using Equations 6.14 and 6.15 to obtain an equation in b and s:

$$\frac{\left(1+e^{sb}\right)\left(1+e^{s\left(b-\frac{1}{2}\right)}\right)+\left(1+e^{s(b-1)}\right)\left(1+e^{s\left(b-\frac{1}{2}\right)}\right)-2\left(1+e^{s\left(b-\frac{1}{2}\right)}\right)\left(1+e^{sb}\right)}{\left(1+e^{sb}\right)\left(1+e^{s\left(b-\frac{1}{2}\right)}\right)\left(1+e^{s(b-1)}\right)} = 0. \tag{6.16}$$

The denominator of Equation 6.16 tends to ∞ at a greater rate than the numerator for large s and b — solutions that are of no interest. Multiplying Equation 6.16 through by the denominator and multiplying out the numerator we get

$$-e^{sb} + 2e^{-\frac{s}{2}+sb} - e^{-s+sb} + e^{2sb-\frac{s}{2}} + e^{-\frac{3s}{2}+sb} - 2e^{-s+2sb} = 0. \tag{6.17}$$

After dividing Equation 6.10 through by common factors and rearranging we obtain

$$1 - 2e^{\frac{-s}{2}} + e^{-s} = e^{sb-\frac{s}{2}}(1 - 2e^{-\frac{s}{2}} + e^{-s}) \tag{6.18}$$

hence, $b = \frac{1}{2}$. Substituting for b in Equation 6.15 yields

$$c = -\frac{1}{1+e^{\frac{s}{2}}}. \tag{6.19}$$

Substituting for c, b and $y = \frac{1}{2}$, $x = \frac{1}{2}$ in Equation 6.10 and solving for m yields

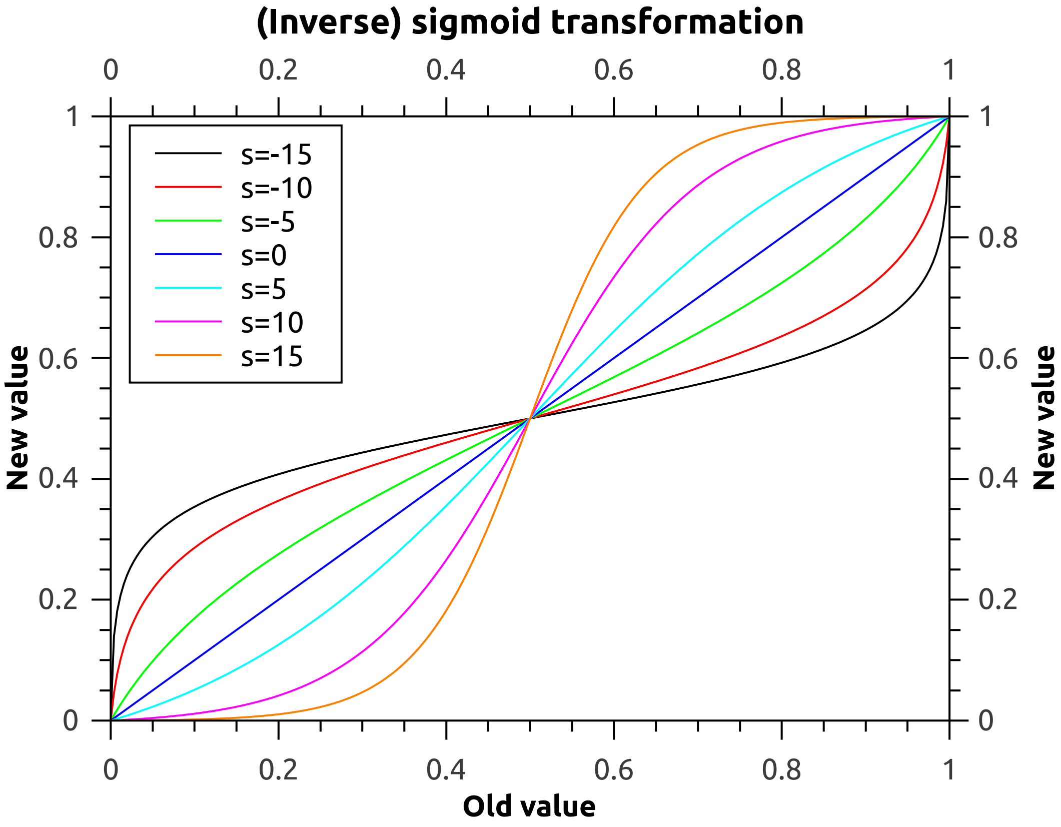
$$m = \frac{e^{\frac{s}{2}}+1}{e^{\frac{s}{2}}-1} \tag{6.20}$$

and hence

$$y = \frac{e^{\frac{s}{2}}+1}{e^{\frac{s}{2}}-1}\left(\frac{1}{1+e^{-s\left(x-\frac{1}{2}\right)}} - \frac{1}{1+e^{\frac{s}{2}}}\right). \tag{6.21}$$

The sigmoid transformation, the image process based on the adjusted sigmoid function, increases the contrast of the middle pixel intensities at the expense of decreasing the contrast of the lighter and darker pixel intensities. It is likely to be the case, however, that the reverse is wanted; increasing the contrast of the lighter and darker pixel intensities at the expense of reducing the contrast of the middle pixel intensities. The inverse adjusted sigmoid function can be obtained from Equation

Figure 6.2: The sigmoid function for various values of s. The negative values of s correspond to the inverse sigmoid transformation



6.21 by swapping the symbols x and y and rearranging to make y the subject:

$$y = \frac{1}{2} - \frac{1}{s}\ln\left(\frac{e^{\frac{s}{2}} - x(e^{\frac{s}{2}} - 1)}{x(e^{\frac{s}{2}} - 1) - 1}\right). \tag{6.22}$$

As $s \to 0$, $y \to x$ for both mappings. Figure 6.2 shows the mappings for various values of s for the sigmoid transformation and the inverse sigmoid transformation. The shapes of the curves suggest that the sigmoid transformation and the inverse sigmoid transformation can be combined into one function. The approach taken is to allow s to take negative values in the input of the combined function. If the value of s is positive then the sigmoid transformation is performed taking s as its parameter. If s is negative then the inverse sigmoid transformation is used with $-s$ as its parameter. For the sake of convenience the combined function of sigmoid transformation and inverse sigmoid transformation will simply be referred to as the sigmoid transformation.

Histogram equalisation

Histogram equalisation is a well known contrast enhancement technique for global contrast enhancement. Histogram equalisation, at its most basic level, takes no parameters as input, only the image to be processed. The goal of the histogram equalisation process is to ensure that the distribution of pixel values after the process is uniform, that is, there are an equal number of pixels of each of the possible pixel values in the image. The pixel values are mapped according to

$$y_k = \text{round}\left[\frac{L-1}{N}\sum_{x=0}^{k} n_x\right] \tag{6.23}$$

where N is the number of pixels in the image, L is the number of possible pixel intensities in the image, n_x is the number of pixels with intensity value x, and y_k is the number of pixels in the processed image with the intensity level k. A more detailed description of the histogram equalisation process, along with its derivation, can be found in [47].

The above description of digital histogram equalisation assumes that the input intensity levels are $0, 1, \ldots, L-1$. The histogram equalisation used in the contrast enhancement process in this experiment however is performed in the Y-axis of the YIQ colour space, and thus can take any value in the range $[0, 1]$. The Y values need to be placed into N equally sized bins before the process can be used. If too few bins are used, too few intensity levels remain after processing. If too many bins are used extra processing is perform for no discernible effect. The number of bins was chosen to be 256 for no reason other than that is the number of intensity levels in an 8-bit image.

Local contrast enhancement

One reason why histogram equalisation can fail to enhance an image is that there is already an even distribution of pixel values in the image. An image in which there are large areas dominated by pixels of similar values, but the overall distribution is even, would benefit more from local contrast enhancement than global contrast enhancement.

Adaptive histogram equalisation (AHE) was presented by Pizer et al. in 1987 [95] and is a popular means of performing contrast enhancement at a local level. The most direct method of performing local contrast enhancement would be to perform a histogram equalisation in the region around each pixel and set the new value of the pixel according to the mapping function defined. This approach, however, is very computationally intensive. Another approach would be to split the image into smaller sections, or tiles, and perform histogram equalisation on each tile separately. This approach would lead to blocking artefacts at the boundaries between the tiles. The implementation used in MATLAB's *adapthisteq* function, and thereby this work, uses interpolation to ensure smooth transitions between tiles. In this implementation the image is split into non-overlapping tiles and each tile undergoes histogram equalisation. The values of the pixels in the top left quarter of the top left tile, top right quarter of the top right tile, etc. are mapped according to the mapping function for those tiles. The values of the pixels in the left half of the leftmost tiles (but not the corners) etc. are mapped according to the tile they are in and the single neighbouring tile (above or below) to which they are closest. The degree to which each of the two tiles determines the new pixel value is determined using linear interpolation. The remaining pixel values, that is, of those pixels not in the corners or at the edges, are determined using bilinear interpolation using the tile the pixel is in and the three other closest tiles to the pixel. For example, a pixel in the bottom left quarter of a tile has its new value set mainly by the histogram mapping function of that tile but also by the histogram mapping functions of the tiles to the left, bottom, and bottom-left.

The number of tiles used has an effect on the result of applying AHE. The number of tiles that produce the best image is determined by the image itself and the intention of the person applying the AHE process. This makes the number of tiles an appropriate variable to be optimised in the image enhancement process. The number of tiles was represented as a parameter t which was the number of tiles in each dimension so a value of $t = 4$ would correspond to $4 \times 4 = 16$ tiles. The value of t was stored as a real number, it was rounded to the nearest integer when input into the local contrast enhancement process.

AHE is the name given to the basic local contrast enhancement method but target histograms other than uniform (the target distribution of pixel values in histogram equalisation) are possible. One such distribution is the Rayleigh distribution

$$y = \frac{x}{\alpha^2} e^{-\frac{x^2}{2\alpha^2}} \tag{6.24}$$

where α is a parameter of the distribution, x is the pixel value normalised to the range $[0, 1]$ and y is the probability density. The value of α changes the shape of the target histogram and therefore the image resulting from the process. It follows that the value of α is another variable that can be optimised to improve the performance of the algorithm.

The AHE algorithm can make regions of an image worse in some situations. If a relatively uniform region contains noise AHE will be likely to accentuate it. It is also likely that the level of contrast enhancement that AHE would provide to that region would be too great. This can be countered using contrast limited adaptive histogram equalisation (CLAHE). In contrast limiting, the counts of the pixel values at each intensity level are manipulated so as to make the original pixel value distribution appear to be closer to the uniform distribution than it actually is. The manipulation consists of setting a ceiling for all of the intensity pixel counts in the histogram. Any pixel counts exceeding this ceiling are redistributed amongst all of the intensity levels, even those that exceeded the ceiling. This manipulation causes the mapping function between the original intensity values and those of the enhanced image to have less impact than it otherwise would have. MATLAB's default value of 0.01 was used.

6.2.3 The piecewise intensity transfer contrast enhancement process

An intensity transfer process is a global process which takes the value of each pixel in a monochrome image and maps it to a new value. In theory, an intensity transfer process is not as restricted as the histogram equalisation, gamma adjustment, and sigmoid transformation processes as the mapping process can take any form desired. However, of all possible mapping transforms, of which there are 256^{256}, the vast majority would render image unrecognisable. In practice, it is necessary to restrict the possible forms the intensity transfer function can take. A simple approach is to define the transfer function mathematically, two examples being gamma adjustment and sigmoid transformation processes. Multiple functions can be combined to create a single transfer function. Pal et al. [89] and Shyn and Leou [110] used GAs to optimise the inputs and weights of enhancement processes which used a weighted sum of various transfer functions to create a single transfer function. Hashemi et al. [51] used a GA to develop a function in which 256 integers constrained to the range $[0, 255]$ were sorted into ascending order to form the intensity transfer function.

As with the compound process, the piecewise intensity transfer process starts with the application of the CBM3D filter followed by contrast stretching in the YIQ colour space. The transfer function chosen in this experiment is a monotonically increasing piecewise function. The piecewise intensity transfer function was chosen to be monotonically increasing in order to avoid any reordering of intensity levels, that is, all pixels of a particular intensity level that had higher intensity level than all pixels of another intensity level would not have a lower intensity level after processing.

There were eight sliders controlling the compound process on the direct interface (see Section 6.3.1 and so to mitigate the effects of having a difference in the number of sliders between the image processes it was decided that the piecewise intensity transfer process should also have eight sliders. As each section required one slider to determine its relative weight, eight sections were used. These sections were chosen to be equal in size as measured along the x-axis, so that the first section covered intensity levels 0 to $\frac{31}{255}$, the second section intensity levels $\frac{32}{255}$ to $\frac{63}{255}$ [1] and so on. The intensity transfer function is defined by eight weights, one for each piece of the function. The weight of a piece determines what proportion of the pixel values in the processed image the piece maps to. For example, if the first piece has a weight of 0.25 then the pixel values $\left[0, \frac{31}{255}\right]$ in the pre-mapped Y channel map to $\left[0, \frac{63}{255}\right]$ in the post-mapped Y channel. Likewise, if the first piece has a weight of 0.0625 then the pixel values $\left[0, \frac{31}{255}\right]$ in the pre-mapped Y channel map to $\left[0, \frac{15}{255}\right]$ in the post-mapped Y channel.

The most elementary requirements for the piecewise intensity transfer function is that it should be monotonically increasing and that it should be continuous at the boundaries between pieces. A piecewise linear function would fulfil these requirements and would likely have been adequate for the task. To ensure that any regions of continuous intensity change did not acquire abrupt changes in gradient after enhancement it was decided to ensure that the first derivative of the piecewise functions at the boundaries was also constant. Using splines would satisfy this condition but splines are not guaranteed to maintain monotonicity. The use of a piecewise cubic hermite interpolating polynomial (PCHIP) ensures that the intensity transfer function is continuous, has continuous first derivatives at the section boundaries, and maintains monotonicity. A PCHIP function is made from a series of cubic functions. A cubic function can be uniquely determined using the two points at each end of a section and the first derivatives at those points. The points are determined using the weighting method above. The first derivatives at the end points

[1] The intensity levels are in the range $[0, 1]$

are calculated using a weighted combination of the gradients of the piecewise linear function either side of each boundary point. There are instances in which the first derivative is not calculated from the gradients, notably when the piecewise linear gradients either side of a point are of different signs or when the gradient either side of a point is equal to zero. In such instances the first derivative at the point is set to zero in order to maintain the monotonicity of the PCHIP function.

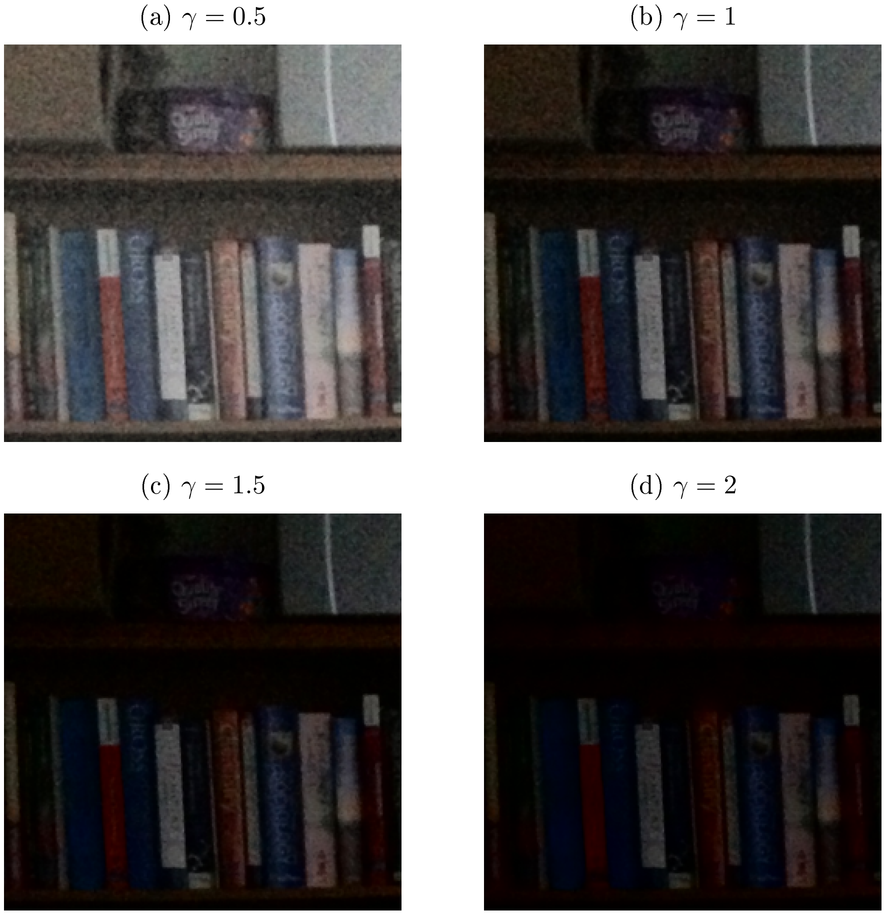
6.2.4 The IEAs used

In both the piecewise and composite image processes the genotypes were represented by nine genes with each gene being a real number. In the piecewise intensity transfer process one of the genes corresponded to the value of σ in the noise removal process and the other eight genes corresponding to the weights in the transfer function. In the compound process, one of the genes represented σ in the noise removal process, four of the genes were the weights of the contributions of the four processes and the remaining four genes represented the input values of the four processes — γ for gamma adjustment, s for sigmoid transformation, and α and t for the local contrast enhancement. Histogram equalisation took no input parameters.

The possible values the weights and inputs to the image processes could take were constrained to ensure that input values that were particularly unlikely to lead to an improvement for any photograph were not used. The weights for both the compound and the piecewise intensity transfer processes were not permitted to go below zero. For the evolutionary interface, the piecewise intensity transfer function weights and the weights of the contrast enhancement functions of the genotypes were normalised so that they summed to 1. The value of γ in the gamma adjustment process was constrained to the range $[0.1, 10]$. The sigmoid transformation input s was constrained to the range $[-30, 30]$. In the local contrast enhancement, α was constrained to the range $[0.3, 1]$, the value of t was constrained to the range $[2, 32]$.

A change in the inputs to the noise treatment, local contrast enhancement, and gamma adjustment processes do not scale linearly with their effect on the photographs. For example, Figure 6.3 demonstrates the effect of various values of γ in the gamma adjustment process. Note that a change in value from 0.5 to 1 is more significant than a change in value from 1 to 1.5 which is in turn more significant than a change in value from 1.5 to 2. This non-linearity means that changing, for example, γ by some amount when it is small has a larger effect than a change of the same size when γ is large.

Figure 6.3: Gamma adjusted photographs for various values of γ



(a) $\gamma = 0.5$

(b) $\gamma = 1$

(c) $\gamma = 1.5$

(d) $\gamma = 2$

This non-linearity has an effect on how the interfaces behave. For the direct interfaces, a small adjustment to the slider that controls the value of γ when the slider is toward the lower (left) end of its range would have a much greater effect than the same adjustment if the slider is toward the upper (right) end of its range. This inconsistency is something a user should not need to adapt to as it can be compensated for. The situation is worse for the evolutionary interface in which individuals with low values of γ would be perceptually altered to a much greater extent than individuals with a high value of γ for equal values of the mutation parameter. The problem in each case can be remedied by using a non-linear scaling between the input values set in the interfaces and the values input to the image processes. An exponential scaling was used for the values of γ, t, α, and σ such that

$$\gamma = e^{\gamma_{\text{search}}} \tag{6.25}$$

$$t = e^{t_{\text{search}}} \tag{6.26}$$

$$\alpha = e^{\alpha_{\text{search}}} \tag{6.27}$$

$$\sigma = e^{\sigma_{\text{search}}} \tag{6.28}$$

where γ, t, α, andσ are the input values of the image processes and γ_{search}, $\mathbf{t}_{\text{search}}$, α_{search}, and ω_{search} are the corresponding values in the search space. This scaling meant that the range of values that the parameters could take in the search space needed to be appropriately scaled. For example, as the value of γ was constrained to the range $[0.1, 10]$, the value of γ_{search} was constrained to the range $[\ln(0.1), \ln(10)] = [-2.30, 2.30]$.

The simple IGA described in Section 5.2.2 was used in this experiment; as it is likely to be better suited to problems that do not have unimodal subjective fitness functions than the hyperplane-IGA.

A single mutation factor for all of the input values to the image processes would not be suitable, a change of 0.5 for one input value may have a greater effect than a change of 0.5 for another input value. Adjustments made to the input values of the image processes by the mutation operation needed to reflect this. The slider had a minimum value of 0 and a maximum value of 1.The scaling factors were set originally to be equal to about one third of the possible range of values for that gene. However, early runs of the algorithm demonstrated the necessity for adjustment and so the scaling factors were set heuristically. The scaling factors were set to: $\text{scale}_\sigma = 1$, $\text{scale}_{\text{weights}} = 0.75$, $\text{scale}_\gamma = 3$, $\text{scale}_s = 5$, $\text{scale}_t = 1$, and $\text{scale}_\alpha = 0.5$.

Maximum and minimum values were set for each of the genes such that the input values to the contrast enhancement process did not fall outside the ranges given above, except the weights which had a minimum of 0. The weights of new genotypes were normalized after being generated so that they summed to 1. If any of the genes of a new genotype had a value outside the permitted range, the new genotype was deleted, the parents returned to the mating pool, and the process began again with pulling two parents at random from the pool.

6.3 Method

6.3.1 The user interfaces

Two types of interface between the user and the underlying image processes were compared in this experiment, direct interfaces and an evolutionary interface.

The direct interfaces

In the direct interfaces the original photograph and the result of applying the image processes were displayed to the participant. The input values of the image processes were manipulated directly using a panel of sliders. The processed photograph would update according to the new set of input values after every alteration of the sliders. It was possible to zoom in on the photographs. The image panels for the photographs were configured such that all panels would display the same area of photograph so that zooming in on a portion of one photograph would zoom in on the same portion of the other photograph. The participant would continue making adjustments to the input values until they were satisfied, or until they thought no further improvement was possible, by clicking on the ‘Finish’ button. The direct interfaces for the compound and the piecewise algorithms were slightly different as can be seen from the screenshots in Figures 6.4 and 6.5.

The input values from the sliders for the direct interface for the compound process were transformed in the same way as those from the search space of the IEA for the compound process. The value of σ in the direct interface for the piecewise intensity transfer process was also scaled accordingly.

Figure 6.4: The direct interface for the piecewise image enhancement process

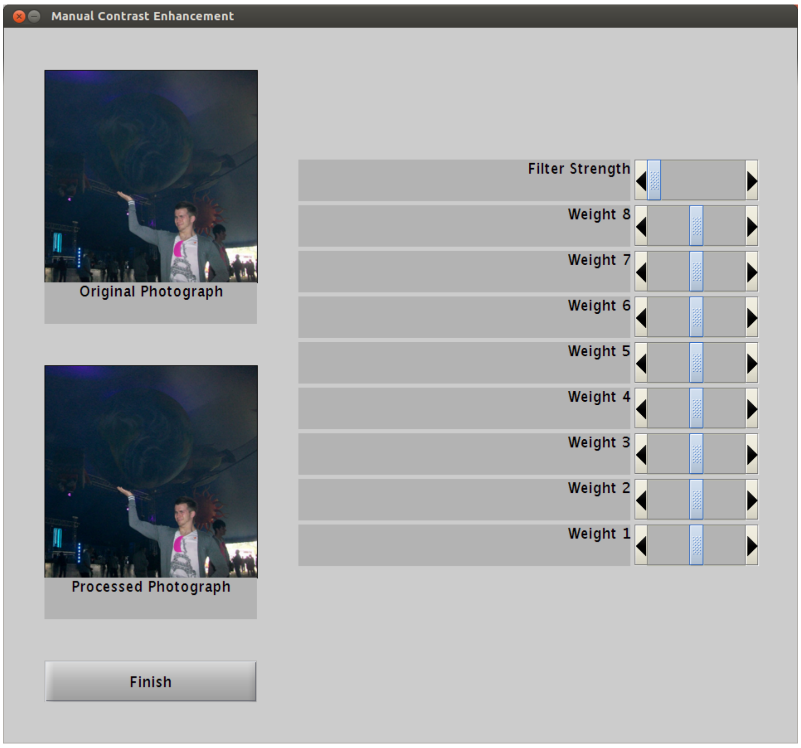
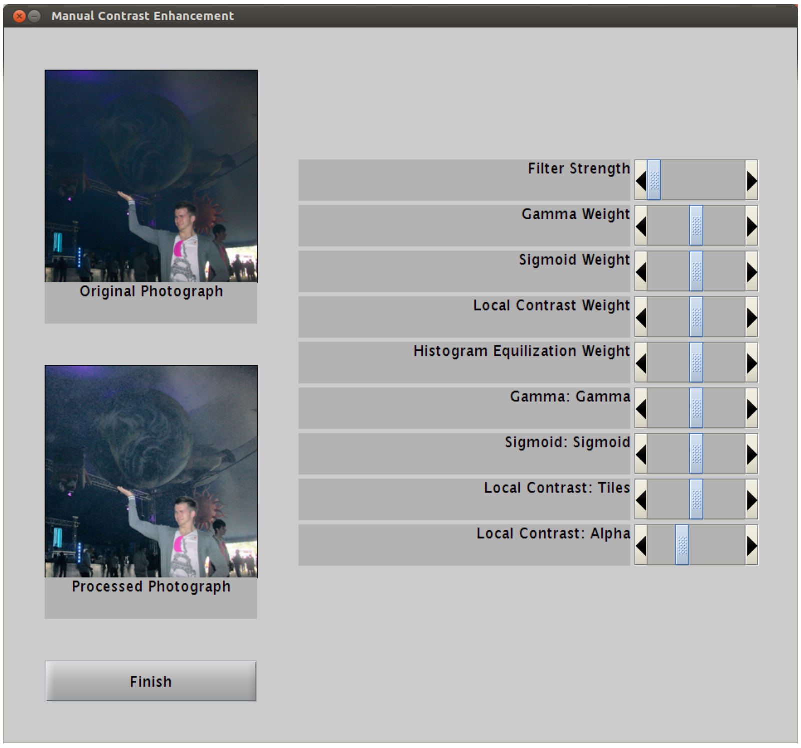


Figure 6.5: The direct interface for the compound image enhancement process



The evolutionary interface

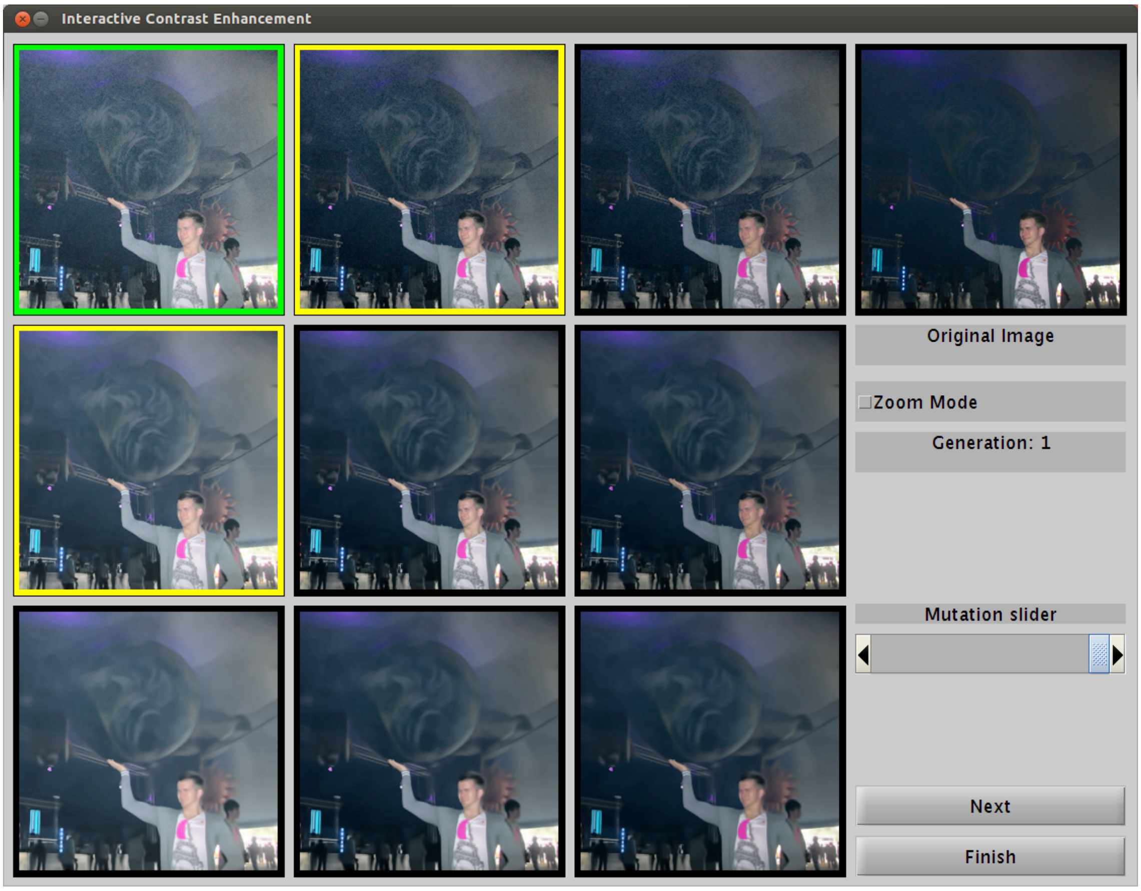
The interface for the IEAs was very similar to that of the algorithm/space colour matching experiment of Chapter 5. Nine photographs were displayed in a 3×3 grid. The original image was displayed in the top right corner for reference. A 'zoom mode' was implemented which could be activated by clicking a check box, which would enable Matlab's default zoom functions. All of the photographs were linked together so that zooming in on a portion of one photograph would zoom in on the same portion of all of the photographs. The mutation slider enabled the participant to set the mutation factor of the underlying GA. The mutation slider was also decremented by 0.05 per generation by the software (the slider's range was $[0, 1]$). The mutation slider was programmed so that the mutation factor in the underlying algorithm was no longer linearly related to the slider's value but to the square of its value. This meant that small adjustments could be made in the mutation factor for when the algorithm was nearing convergence. A screenshot of the interface is given in Figure 6.6.

Every generation the participants would choose a best photograph from the photographs on display and select it by clicking on it using the left mouse button. The participants also had the option of selecting other images that they thought were good, any number from zero to eight. This could be achieved by clicking on the photographs using the right mouse button. A green border was placed around the photograph the participant preferred, a yellow border for those photographs the participant also selected, and a black border for those photographs that were not selected. Once they were satisfied with their selections, the participant would go to the next generation by pressing the 'Next' button. The preferred photograph was carried forward into the next generation. The preferred photograph and the selected photographs were used in creating the next generation of photographs. The participants would continue the process until they were satisfied, or until they thought no further improvement was possible, by clicking on the 'Finish' button.

6.3.2 Test set-up

Thirty participants were used in this experiment. The participants were principally drawn from the postgraduate students in the School of Physical Sciences at the University of Kent, with the remainder being undergraduate students or staff. As the task was cognitively simple, no knowledge was required except a basic level of

Figure 6.6: The evolutionary interface for both contrast enhancement processes



computer literacy.

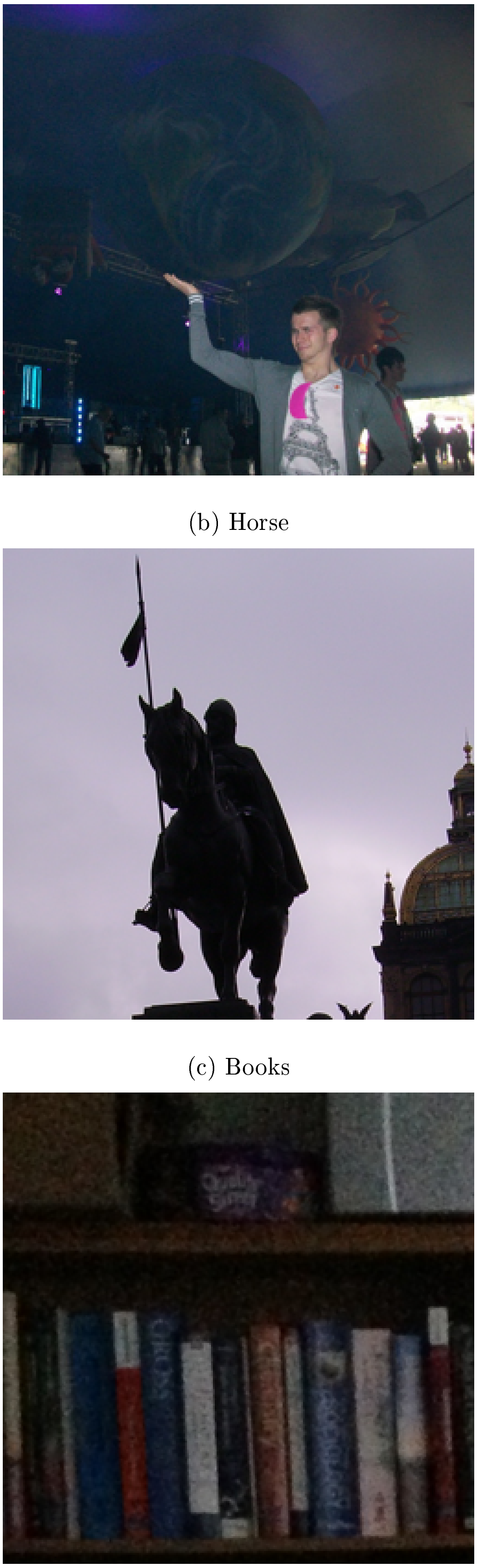
Two interfaces were used; direct and evolutionary, two image processes were used; compound and piecewise intensity transfer, and three photographs; Atlas, Horse, and Books. Each participant developed twelve recipes — one for each combination of interface, image process, and photograph. The first four recipes were developed on the Atlas photograph. Data from the development of these recipes were not recorded or used for evaluations as the goal of the first four runs was to give the participants a practice run on each of the combinations of interface and contrast enhancement process. The principal reason for the practice runs was to ensure that the time it took to learn how to use the interfaces and how they behaved did not affect the recorded time it took to develop each of the recipes. Data were recorded from the development of the other eight recipes, most importantly the time taken to develop the recipes. It was the recipes developed in these eight runs that were compared at the end of the experiment.

The practice photograph, Atlas depicts a young man ‘resting’ a globe on his hand, though this is not apparent until the photograph undergoes some contrast enhancement. The Atlas photograph was taken by the author’s brother using a Kodak Easyshare M1063, a low end dedicated digital camera, and posted to the Facebook social networking website. The Atlas photograph is slightly noisy but also underexposed. This photograph therefore encourages use of the filter as well as the contrast enhancement processes.

The Books photograph is typical of underexposed photographs taken with devices whose primary function is not taking photographs. It was taken by the author’s supervisor using an iPhone 5 in the author’s office. The photograph was then reduced and cropped. The reduction and cropping was performed so that there was enough content in a 256×256 photograph (a 256×256 section of the original photograph had very little content). The image is underexposed and when the contrast is enhanced the noise in the image becomes particularly visible. This image encourages heavy use of the filter as well as some use of the contrast enhancement processes.

The Horse photograph is a section of a photograph taken by the author using a Pentax Optio 50, a low end dedicated digital camera. The photograph has virtually no noise but does benefit from contrast enhancement. This image demonstrates a common problem with digital cameras, that the details of a dark object in the foreground can be hidden because of light from a bright background such as sky or light through a window. The photographs are shown in Figure 6.7

Figure 6.7: The photographs used in the contrast enhancement experiment



(a) Atlas (the training photograph)

(b) Horse

(c) Books

The piecewise intensity transfer process on the direct interface started with no noise removal (the 'Filter Strength' tab was set to 0) and all of the weight sliders were set to be equal to 0.5 so that each section had the same weight.

The compound process on the direct interface also started with the 'Filter Strength' set to 0. The four weights were set to 0.25. The value of γ was set to 1, s was set to 0, t was set to 8 and α was set to 0.4. These values were selected because they were neutral (in the cases of γ, the weights, and s) or because they were the default values used in the Matlab implementation of the function (in the cases of t and α). This led to most of the sliders taking their centre positions, giving the appearance of neutrality in the initial values.

The piecewise intensity transfer process on the evolutionary interface had an initial population of photographs with $\sigma = 0.1, 5, 15$ and the weights set with $weight_i = i^{power}$ where $power = 0.5, 1, 2$. Each combination of these weights and values of σ were in the initial population.

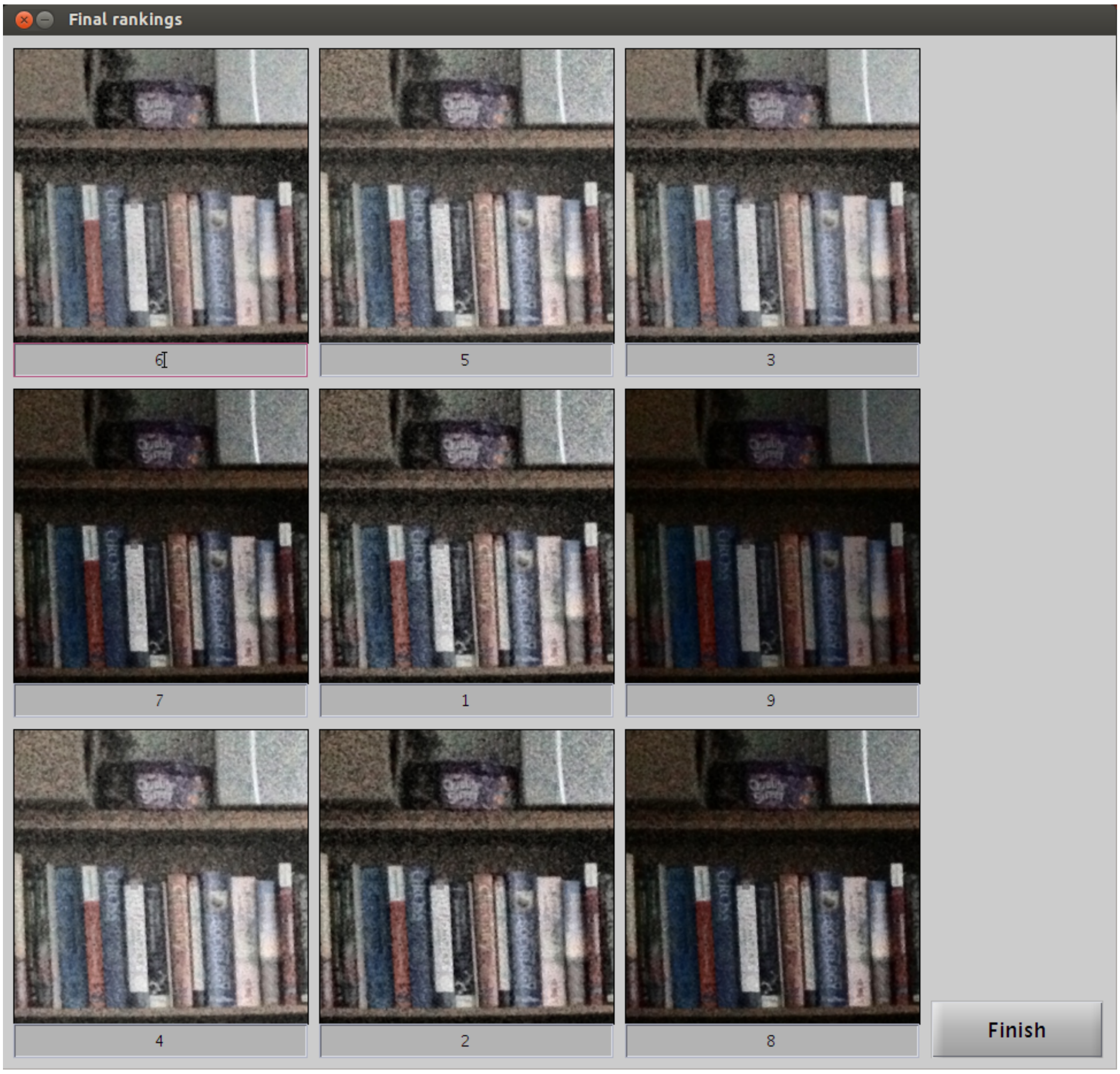
The compound process on the evolutionary interface had an initial population of photographs with $\sigma = 0.1, 5, 15$ and $\gamma = 0.25, 1, 4$. The remaining values were $t = 8$, $s = 0$, and $\alpha = 0.4$.

An observation made from the colour matching experiment was that participants would often develop (erroneous) mental models of what the evolutionary algorithms were doing beneath the interfaces. It was not feasible to give the participants a full explanation of how the image processes and interfaces worked. The participants were informed as to which interface, image, and image process they were to be using before each run. This was done to avoid any problems with participants becoming frustrated with inconsistent behaviour between the two image processes when using the evolutionary interface.

6.3.3 Data gathered

After the participants had developed their eight recipes the participants were asked to rank the results of processing the photographs using those recipes. The Horse or the Books photograph (whichever the participant used to develop recipes first) was processed with the eight recipes the participant developed. The eight resulting photographs plus the original were displayed in a random order on an interface very similar to the evolutionary interface. This interface also included a zoom function which worked like that of the evolutionary interface. The participant was asked to rank the photographs from 1 to 9 with 1 being the best photograph and 9 being the

Figure 6.8: The ranking window



worst. The ranking window is shown in Figure 6.8. After the participant had ranked the photographs of the processes applied to one image they ranked the photographs of the processes applied to the other.

For a subjective non-targeted task such as photograph enhancement, participant perception of the results of the processes is as important as the objective measurements made. If someone perceives approach A to be better than approach B then they will prefer to use approach A, thus the participants were asked to provide subjective feedback on the two interface types. It was decided not to ask for feedback on the different image processes partly because they are not the focus of the experiment but mainly because of the likelihood that asking the participants to compare the image processes would impose a cogitative burden that would affect the validity of the data gathered about the participants' preferences concerning the interfaces.

Five questions were asked about the interfaces:

- Which interface did you feel was fastest?
- Which interface did you feel gave you most control?
- Which interface did you feel was easiest to use?
- Which interface did you feel gave you the most satisfactory results?
- All things considered, which interface did you feel was the best?

The participants were shown the questions just before the recorded runs of the experiment so that they could consider them whilst they were performing the photograph enhancement task. The participants answered the questions at the end of the experiment.

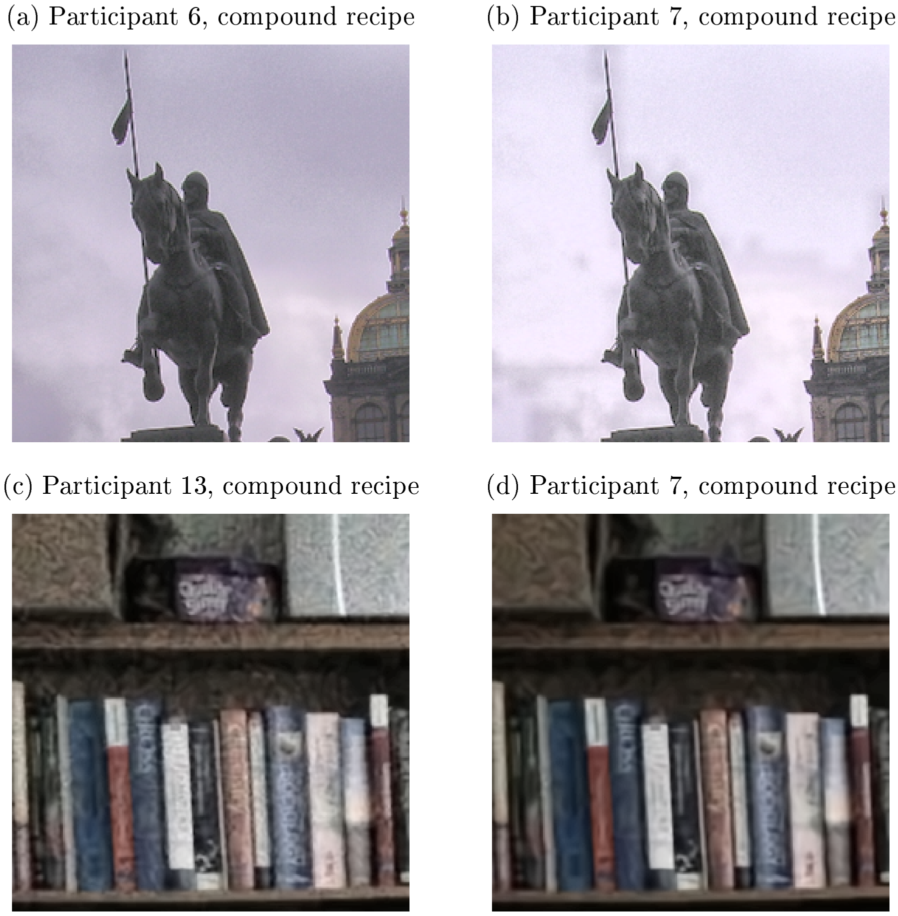
The only objective data gathered for comparing the interfaces and processes was the time taken to develop each recipe, that is, the time taken to complete each run of the experiment.

6.4 Results

6.4.1 Examples of enhanced photographs

Figures 6.9 and 6.10 show some of the processed photographs that were ranked as 1 (best) at the comparison stage. It can be seen that the participants had different opinions on what constituted a satisfactory result.Tables 6.1 and 6.2 show the recipes used to create the photographs in Figures 6.10 and 6.9. Figure 6.11 shows the piecewise intensity transfer functions used to create the photographs in Figure 6.10 The input values are the Y values of the pixels in the images, which lie in the range $[0, 1]$. The output values are the new Y values of the pixels after the images have been processed using the piecewise intensity transfer function. Figure 6.11 shows that for these four particular piecewise intensity transfer recipes that recipes that resulted in an overall brightening of the images were preferred. It may have been possible to obtain similar processed images using a logarithmic transform or gamma adjustment with $\gamma > 1$.

Figure 6.9: Examples of photographs enhanced using compound recipes ranked 1 (best)



(a) Participant 6, compound recipe

(b) Participant 7, compound recipe

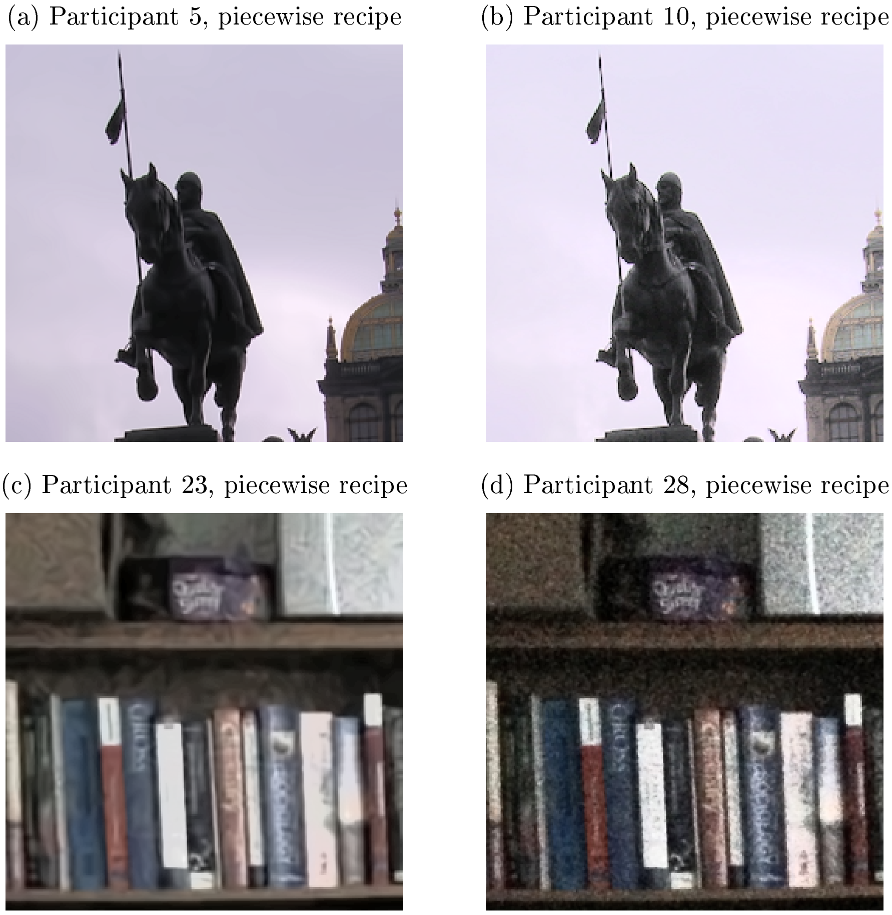
(c) Participant 13, compound recipe

(d) Participant 7, compound recipe

6.4.2 Perceptual feedback

Binomial tests (see Section 2.2.4) were performed to compare the participants' preferences on each of the five questions asked on the feedback questionnaire. The results

Figure 6.10: Examples of photographs enhanced using piecewise intensity transfer recipes ranked 1 (best)



(a) Participant 5, piecewise recipe

(b) Participant 10, piecewise recipe

(c) Participant 23, piecewise recipe

(d) Participant 28, piecewise recipe

Figure 6.11: Graphs of the piecewise recipes applied to the photographs in Figure 6.10

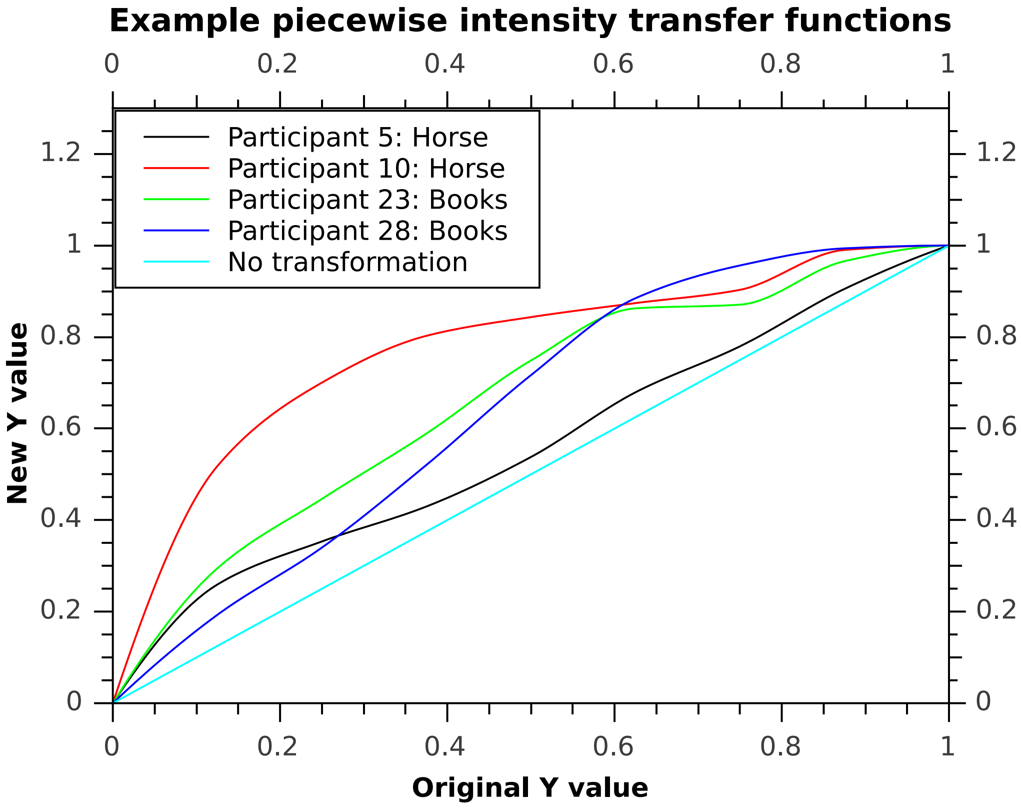


Table 6.1: Example recipes developed for the compound process

			Weights							
Participant	Photograph	σ	γ	sigmoid	CLAHE	Hist. Eq.	γ	t	α	s
7	Horse	0.2490	0.3745	0.1637	0.3161	0.1456	0.2695	12	0.3772	-13.8280
8	Horse	0.1597	0.2381	0.0777	0.5705	0.1137	0.1208	27	0.8025	8.7031
14	Books	6.7656	0.5961	0.0021	0.1425	0.2592	0.7693	6	0.3345	3.1366
8	Books	11.8935	0.1741	0.5551	0.0959	0.1749	0.8792	29	0.6453	-0.7719

Table 6.2: Example recipes developed for the piecewise intensity transfer process

			Weights							
Participant	Photograph	σ	1	2	3	4	5	6	7	8
5	Horse	3.8912	0.2815	0.0913	0.0741	0.1049	0.1371	0.0975	0.1217	0.0920
10	Horse	1.0920	0.5179	0.1824	0.1023	0.0409	0.0306	0.0290	0.0872	0.0096
23	Books	12.0342	0.2946	0.1517	0.1427	0.1596	0.1139	0.0081	0.0950	0.0342
28	Books	0.5871	0.1930	0.1469	0.1806	0.1961	0.1681	0.0716	0.0374	0.0062

Table 6.3: Interface preferences

	Preferred interface			
Factor	Direct	Evolutionary	p-value	Significant
Fastest	14	16	0.856	no
Most control	26	4	< 0.001	yes
Easiest to use	7	23	0.005	yes
Most satisfactory results	12	18	0.362	no
Overall best	12	18	0.362	no

are shown in Table 6.3. The table shows that there was a significant preference for the direct interface in terms of control offered and a significant preference for the evolutionary interface in terms of ease of use.

Chi-square tests of independence (see Section 2.2.4) were performed between each pair of questions the participants were asked in the feedback questionnaire. There was significant interaction between the interface which the participants felt gave the most satisfactory results and the interface the participants thought was best, $\chi^2(1, N = 30) = 26.00, p < 0.001$; in fact, whichever interface a participant thought gave the most satisfactory results was the one that they chose as best overall. There was also significant interaction between the interface which the participants felt was fastest and the interface the participants thought was easiest to use, $\chi^2(1, N = 30) = 5.593, p = 0.018$.

6.4.3 Rankings and timings

Kendall's coefficient of concordance for ranks was calculated for the participants' rankings over each of the two sets of ranked images. For the Horse photograph, $\chi^2(8, N = 30) = 171, p < 0.001$. Kendall's W is 0.713 indicating strong agreement among the participants. For the Books photograph, $\chi^2(8, N = 30) = 107, p < 0.001$. Kendall's W is 0.446 indicating moderate agreement among the participants. This demonstrates that there is a significant correlation between the rankings awarded by the participants and therefore that the differences between the mean rankings can be taken to have significance.

The means of the ranks assigned to the images are presented in Table 6.4. The Friedman test with Fisher's LSD for rank post hoc test performed on the ranks assigned to the Horse images shows that the original photograph is the least preferred

Table 6.4: Mean (standard deviation) of participant ranks awarded to the test images after being processed by participant developed recipes

Recipe			Photograph	
Photograph used	Interface	Algorithm	Horse	Books
None (original photograph)			8.57 (0.73)	7.53 (1.66)
Horse	Evolutionary	Compound	2.13 (1.14)	6.27 (1.76)
Books	Evolutionary	Compound	6.00 (1.58)	2.28 (1.57)
Horse	Direct	Compound	2.03 (1.22)	5.83 (1.70)
Books	Direct	Compound	6.30 (1.47)	3.27 (1.46)
Horse	Evolutionary	Piecewise	2.87 (1.28)	6.57 (1.81)
Books	Evolutionary	Piecewise	6.47 (1.87)	3.43 (2.15)
Horse	Direct	Piecewise	4.00 (1.76)	6.23 (2.94)
Books	Direct	Piecewise	6.63 (1.27)	3.10 (2.12)

version of the image, $\chi^2 = 171.2, p < 0.001$; post hoc comparison with the next lowest ranked image, $t(232) = 5.020, p < 0.001$. A similar test on the ranks assigned to the Books image shows that whilst the unprocessed image is the least preferred image it is not quite significantly worse than the second lowest ranked image, $\chi^2(8) = 107.0, p < 0.001$, post hoc comparison to the next lowest ranked image, $t(232) = 1.806, p = 0.072$.

The Friedman test does not detect interaction effects between the factors (the image the recipe was developed on, the process used, and the interface used). To compare the effects and interactions of these factors the original image was removed from the rankings, the ranks adjusted to reflect this, and an ART with three-way ANOVA (see Section 2.2.4) was applied to the ranks assigned to each set of images (Table 6.5). It can be seen from Table 6.5 that for both the Horse image and the Books image that the image a recipe was developed on had a significant effect on its performance. It can be seen from Table 6.4 that a recipe performs better on the image it was developed on. Other significant effects were found for the ranks assigned to the Horse image, though their meaning is hard to discern from Table 6.4 and it is necessary to interpret the ART data directly. For the main effect of interface used the mean ART rank for the evolutionary interface is 109.58 and that of the direct interface is 131.42, hence it is concluded that the evolutionary interface produced more satisfactory recipes than the direct interface. Similarly, for the main effect of process used the mean ART rank for the compound process is 98.65 and that of the piecewise process is 142.34, hence it is concluded that the compound process produced more satisfactory recipes than the piecewise process. The interaction effects are harder to interpret. For the interaction between the source

Table 6.5: ART with three-way ANOVA for the participant rank assignments in the comparison of recipes developed on the Horse and Books images, the evolutionary and direct interfaces, and the compound and piecewise image processes

	Horse		Books	
Effect	$F(1, 232)$	p-value	$F(1, 232)$	p-value
Source image	377.998	< 0.001	135.240	< 0.001
Interface	5.946	0.016	0.022	0.883
Process	25.943	< 0.001	0.704	0.402
Source image * interface	0.485	0.487	0.199	0.656
Source image * process	6.189	0.014	0.786	0.376
Interface * process	2.946	0.087	0.210	0.647
Source image * interface * process	5.340	0.021	1.742	0.188

image and the image process used the compound process on the Horse image had a mean ART rank of 113.17, the compound process on the Books image had a mean ART rank of 124.43, the piecewise process on the Horse image had a mean ART rank of 120.27, and the compound process on the Books image had a mean ART rank of 124.13. This can be interpreted as being due to the Horse image benefiting from local contrast enhancement, which only the compound process provides, to a greater extent than the Books image. The interaction between all three effects appears to be a residual from the main and other interaction effects.

The means of the time taken to develop the recipe for easch combination of image, interface, and process are presented in Table 6.6. The ART three-way ANOVA process was applied to the times taken to create the recipes. The results are presented in Table 6.7. It can be seen that the only significant effect was the interaction between interface and image: the recipes developed on the Horse image using the evolutionary interface had a mean ART rank of 111.08, the recipes developed on the Horse image using the direct interface had a mean ART rank of 136.71, the recipes developed on the Books image using the evolutionary interface had a mean ART rank of 122.43, and the recipes developed on the Books image using the direct interface had a mean ART rank of 111.73. These mean ranks can be taken to mean that it was quicker to develop recipes on the Horse image using the evolutionary interface and quicker to develop recipes on the Books image using the direct interface.

Table 6.6: Mean (standard deviation) of times taken to develop recipes

	Recipe		
Photograph used	Interface	Algorithm	Time
Horse	Evolutionary	Compound	115.50s (60.99s)
Books	Evolutionary	Compound	139.73s (75.06s)
Horse	Direct	Compound	100.73s (46.45s)
Books	Direct	Compound	118.47s (60.21s)
Horse	Evolutionary	Piecewise	108.53s (63.87s)
Books	Evolutionary	Piecewise	157.87s (101.83s)
Horse	Direct	Piecewise	127.57s (61.50s)
Books	Direct	Piecewise	103.87s (40.12s)

Table 6.7: ART with three-way ANOVA for the time taken to develop recipes in the comparison of recipes developed on the Horse and Books images, the evolutionary and direct interfaces, and the compound and piecewise image processes

Effect	$F(1, 232)$	p-value
Source image	2.440	0.120
Interface	2.310	0.130
Process	0.176	0.674
Source image * interface	4.010	0.044
Source image * process	0.254	0.615
Interface * process	0.353	0.553
Source image * interface * process	2.104	0.148

6.4.4 Comparison between participants' perceptions and rankings and timings

The participants' perceptions of how long the recipes took to develop were compared to the actual times and the participant assigned rankings using the chi-square test of independence (see Section 2.2.4). For the times taken $\chi^2(1, N = 30) = 1.35, p = 0.245$. For the satisfaction with the results $\chi^2(1, N = 30) = 0.176, p = 0.675$. There is no significant correlation between the participants' perceptions of which interface was fastest or provided the best processed images. This implies that the time taken to develop the recipes and the comparative performance of the recipes are not reliable measures of user satisfaction.

6.4.5 Discussion

Participants would often concentrate on a single part of the image such as the area to the viewer's right of the man's face in the 'Atlas' photograph. This is likely to be why the participants used the zoom function far more than anticipated. If it had been known that the participants would use the zoom function to the extent they did, more time would have been devoted to making the zoom function more user friendly. For example, with some effort the zoom could have been implemented so that if a section of the photograph was enlarged then the same part of the photograph would remain enlarged after adjustment of the sliders (in the case of the direct interfaces) or when a new generation of processed photographs was displayed (in the case of the evolutionary interface) as opposed to the zoom being reset so that the entire photograph was displayed. The fact that the zoom factor reset after every alteration to the parameters (in the direct interface) or after the generation of a new population (in the evolutionary interface) did serve to encourage the participants to evaluate each image as a whole rather than concentrate on a single section.

Some participants decided to finish developing their photograph on the evolutionary interface if the population as a whole was worse than the previous generation. Many participants expressed the desire for a 'back' button on the evolutionary interface, particularly when using the piecewise intensity transfer process on the practice photograph.

6.5 Conclusion

In this experiment two different image processes for enhancing contrast in photographs were compared; a compound process and piecewise intensity transfer function. Two different ways of manipulating the inputs to the processes were also compared; an IEA and an interface which allowed direct manipulation of the values via a set of sliders.

The results show that whilst it is possible to develop a recipe on one photograph and use it to improve another photograph it is better to develop recipes on the photographs upon which they are to be used.

The participants generally preferred recipes developed using the composite process over those developed using the piecewise process for enhancing the Horse image. No such difference was detected for the books image. This finding suggests that for some images developing image enhancement processes with compound processes yield better results than intensity transfer only processes such as those used in [106] and [130].

The participants generally preferred the Horse image processed by recipes developed using the evolutionary interface to those developed using the direct interface. No such difference was detected for the books image. This finding suggests that for some images the evolutionary interface yields better results. This agrees with conclusions regarding preferences of IEA developed images reported in [130] and [77].

A difference was found between the evolutionary interface and the manual interface with regards to the time taken to develop recipes on the Horse image. This finding suggests that for some images the evolutionary interface is quicker than the manual interface. This would agree with the conclusion in [130] that an evolutionary interface is quicker than a manual one.

Chapter 7

Comparison of search spaces and search operators for the evolutionary development of facial composites

7.1 Introduction

When a crime is investigated, investigators have an array of tools they can bring to bear. The tools employed depend on the nature of the crime and the circumstances surrounding it. There are occasions in which a crime is witnessed by people who see the perpetrator's face but do not know the identity of the perpetrator. In such cases it is often useful to create a pictorial likeness of the suspect from eyewitness accounts [80]. Ideally, someone recognises the person in the likeness, knows their identity, and relates the identity of the suspect to the investigators. The likeness can also serve to gather more information from people who, for example, were unaware they had seen anything of significance. These people may remember seeing the person represented in the likeness at some point around the time the crime was committed and can come forward and provide more information.

The earliest approach to developing a likeness was to use a sketch artist. Sketch artists have been used to create facial likenesses for over 100 years and are still commonly used in the USA [129] . In this approach, the artist interviews the witness to obtain a facial description so that a likeness can be created. The sketch undergoes a series of alterations and refinements until the witness is satisfied with the result. The sketch method is relatively slow and requires a skilled artist in order to be effective.

Identikit [27] was released in 1959 and negated the need for a sketch artist. Identikit consisted of a library of line drawings of various parts of the face printed on transparencies which were combined to make a *composite* facial likeness. An operator would manipulate the transparencies, moving and replacing them as necessary, based on feedback from the witness. This approach was later extended with Identikit II and Photofit [93] each of which consisted of a library of parts of the face taken from photographs instead of line drawings. The physical nature of these composite tools limited the number of facial features that could be included as a greater number of features would require more storage space and also require more time for the operator to search through the library for the required features.

Computerised composite tools such as EFIT were developed in the 1980s. They still relied on a library of face parts which meant that the operator still had to search for the appropriate parts when constructing a composite. However, the computerised tools enabled the operator to resize and rotate the component parts, allowing a far greater range of faces to be composed.

With the exception of the sketch artist method, the methods listed above are component based; composites are created by assembling parts of faces to make a whole. Evaluations of the composite process revealed that they were not particularly effective at creating recognisable faces [18], [9] (cited in [37]). Psychological research suggests that human beings recognise faces not by their individual components but as a whole [127, 24]. The appearance of one facial feature, such as the nose, can alter the appearance of another facial feature, such as the eyes. It is also known that people are better able to recognise faces than they can recall and describe them. An alternative approach to having a witness recall details of parts of the face and having the operator change individual features is to use the human capability for the recognition of faces. One way this can be achieved is to present a number of faces to the witness and allow the witness to choose the face(s) which bear the closest resemblance to the suspect. These faces could then be used as the basis for generating more faces and again the witness chooses those faces which bear the closest resemblance to the suspect and so on until the witness is satisfied with the composite. This *holistic* approach is used in two commercial systems developed in the early 2000s; EvoFIT [37] and EFIT-V (originally called EigenFIT) [41].

The holistic approach requires some means of encoding faces such that it is possible for a witness to create a likeness of the suspect with relative ease. A search space or, more appropriately, a *face-space* is required in which faces can be represented parametrically. Here the term face-space is used to mean a mathematical construct which models aspects of the psychological concept of a face-space as presented by Valentine [135]. Valentine proposed that faces are mentally represented as points in a multidimensional face-space. Each face that a person is familiar with occupies a point in their face-space. Faces which are similar in appearance are located at points near to each other in their face-space; faces which look less alike are located at points farther from each other.

Computer representations of faces (i.e. face images) exist in a very high dimensional space. If the face images are h pixels high and w pixels wide then the face images have $h \times w$ dimensions (or $h \times w \times 3$ dimensions if they are colour images). The majority of possible images in this space will not be discernible as anything other than noise — they will not resemble faces at all. Some means of defining a subset of the image space is needed which includes images that resemble faces but not images that do not. This is achieved through the use of a *face model*. Such a model is constructed using a number of face images as a training set. All faces constructed by the face model are derived from the training set. In the early work of Sirovich and Kirby [112, 64] the faces in the training set were aligned on their axes of symmetry and the axes upon which the eyes lay. The faces were also adjusted so that the width of each face was the same. The face-model was constructed from the training set through the use of *principal components analysis* (PCA). PCA is a mathematical technique for transforming data expressed in an n-dimensional form to a different form of n dimensions or fewer. The transformed expression organises the axes such that the first dimension or *principal component* (PC) expresses as much of the variability in the data as possible. The second PC expresses as much of the remaining variability in the data as possible and so on. If the data is highly correlated, it is possible to express most of the variation in the data in very few PCs. PCA offers a means of compressing data expressed in a large number of dimensions to an expression requiring far fewer dimensions with very little loss of information.

Sirovich and Kirby's approach lead to some blurring of features. Craw and Cameron [21] addressed this issue by warping the faces in the training set to a mean face shape before applying PCA. Cootes et al. [20] extended Craw and Cameron's approach by using the training set to build a shape model and a texture model and then combining the two to create a face model. This is similar to the approaches used in EvoFIT and EFIT-V and is summarised in Section 7.2.

All of the faces that could possibly be created by a face model constitute that face model's face-space. Some means of searching a face-space for a likeness to the face a witness has in their mind is required. The iterative process of allowing the witness to select one or more faces from a number of faces and then using the selections to create more faces immediately suggests the use of an IEA. To assess whether the use of an IEA may be appropriate, the IEA suitability questions of Section 2.2.3 are addressed:

- *Is the subjective fitness function unimodal in the search space?* Yes, there is a single point in the search space which provides the closest match to any given face. However, human recognition, and consequently evaluation, of the faces is somewhat noisy and thus search methods which assume unimodality should be used with caution.

- *What other approaches to the problem are available?* The face-space could be searched by changing the location of the point in the face-space being considered directly. This is what was done by Brunelli and Mich [12] in their prototype holistic composite software 'SpotIt!'. The problem with direct manipulation such as this is that a single PC may affect more than one aspect of the composite; for example, one PC could affect face width and skin tone. Conversely, a single face property such as, for example, face width may be affected by a number of PCs.

- *Is the search space large?* Yes. For example, the face-spaces used in EFIT-V have 60 dimensions. In a PCA face-space, most of the variation is in the first few PCs, so movement of a face point along the axes of the higher dimensions in the face-space has very little effect on the resulting composite. If the conservative estimate is made that only the first 15 PCs have any effect on the composite and variation along any single PC results in only three distinct faces, the face-space still contains over 10^7 possible faces.

Considering these points it is reasonable to conclude that using an IEA to search the face-space for a match to the suspect's face is a justified approach.

Whether it is constructed holistically or componentwise a facial composite requires some details to be added manually. For example, details such as tattoos, birthmarks, and scars. Furthermore, the PCA face model approaches described cannot cope well with fine detailed features such as hair and beards, these need to be added using overlays in the same way as in the componentwise approach. As well as being able to add details such as these, EvoFIT and EFIT-V both include controls that enable direct manipulation of the composites. Both have controls relating to semantic notions so that a witness can have the operator make a face appear to be, for example, more (or less) 'friendly' or 'hard'. Direct adjustments of particular features such as making the eyes wider or the lips thinner can be also performed. It can be seen that the IEA is not required to obtain an exact match and indeed it is generally unable to do so. The purpose of the IEA can therefore be viewed as being one of searching for the best face possible within the confines of the face-space. After a good match is found within the face-space direct methods are used to complete the composite.

There has been much work done to improve the quality of the facial composites developed using PCA-based face models. The development of the face models themselves and the addition of semantic and direct manipulation tools to EvoFIT and EFIT-V have already been mentioned. Bruce et al. [11] found that a composite which was itself a combination of four composites created by different people achieved the same recognition rates as the best composite used to make the combined composite. Frowd et al. [35] found that having a witness create two composites of a suspect's face lead to improved recognition rates. Valentine et al. [136] found that composites created from those of four different witnesses showed better recognition rates than those created from four composites created by the same witness which in turn had better recognition rates than the individual composites. Recently Frowd et al. [38] reported how changes in the way witnesses were interviewed could lead to composites which had better recognition rates.

The emphasis on all of the work above has been toward creating composites which are more likely to be recognised. This is the most important measure of improvement to the process of creating a facial composite but it is not the only one. Selection of an appropriate evaluation method and population size can reduce fatigue and make the process less difficult for a witness. In the early stages of development of both EvoFIT and EFIT-V, full scale rating was trialled as an evaluation method [49, 90] but was abandoned in both cases. For EvoFIT the population size was determined by the number of faces that could be comfortably displayed on a monitor [37]. For EFIT-V the population size was set to 9 because it was thought that a population size of 9 offered a good compromise between convergence speed and usability [39].

Selection of an appropriate IEA, associated operators, and the values of any associated parameters may also have an effect on the composite creation process.

IGAs were selected for both EvoFIT and EFIT-V (even though EFIT-V does not use recombination, it does not use the defining feature of an ES: self-adaptive step size). EvoFIT uses the simple IGA introduced in Chapter 5 though with a different mutation operator [34]. EFIT-V uses an SMM-IGA, though other algorithms were considered early in its development [90]. In each case, the mutation parameter values and other aspects of the algorithm were set with the aid of mathematical models of human evaluation akin to the virtual user of Chapter 4. This is not surprising as it would be a time consuming and laborious process to optimise such parameters using human evaluation. The choice of EA and perhaps the associated mutation and recombination operators does warrant some human comparison. Very little work has been done to compare the performances of different IEAs for use in the creation of facial composites. A series of small experiments evaluating the performances of various nature-inspired metaheuristic algorithms have been conducted [69, 4]. The results indicate that the choice of algorithm has some, but not much, effect on the recognition rates of the composites.

Work on the development and comparison of mutation and recombination operators is an active aspect of research in EAs, however, comparison of these operators with regards to IEAs is virtually unknown. With the time and effort required to perform any form of comparison at all regarding IEAs this is not surprising. The only work that could be found concerning the comparison of mutation or recombination operators in IEAs is by Oinuma et al. [87] in which four recombination operators were compared on a face image beautification task. The details of how the experiment to compare the operators was conducted is unclear, but the final output of the IEA using each of the operators was compared to a manually beautified face image using the mean square error. It was concluded that there was a difference between the recombination operators and the one proposed in the work was found to be the best.

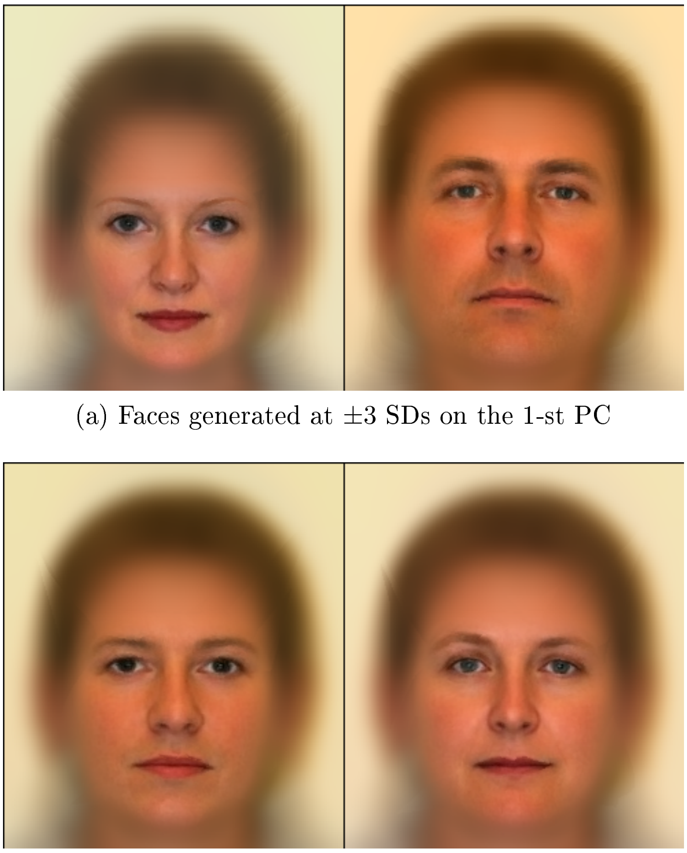
The imperfect nature of human face recognition and the need for direct manipulation of the composites means that not all of the PCs make a significant contribution to the composites. EvoFIT uses the first 71 PCs [37] and EFIT-V uses the first 60 PCs [115]. It is quite possible that so many PCs are not required and the face-space can be constructed from fewer PCs with no perceptible difference in the performances of the face-spaces. From a mathematical perspective each PC accounts for less of the variation in the faces than the previous one. For example, Figure 7.1 shows two pairs of faces generated from the PCA face-space used for the experiments in this chapter. The first shows a pair of faces whose positions in the PCA face-space are at ±3 standard deviations (SDs) on the first PC. The second pair has positions at ±3 SDs on the 30-th PC. The variation accounted for by a particular PC may be due to aspects of the face images which do not have an effect on identity such as face tilt and facial expression. Human perception of identity is less sensitive to these factors and thus PCs whose variation is mainly due to these factors are likely to be less important perceptually than they are to the mathematical prioritisation. Using human evaluation to select the PCs to be used in an n-dimensional face-space may provide a face-space which perceptually accounts for more variation in the faces than one which simply uses the first n PCs.

In the first experiment reported in this chapter a 12-dimensional 'human reduced' face-space is constructed using human evaluation of the differences between pairs of faces from the 'large' 30-dimensional face-space. In Chapters 4 and 5 a trivial colour matching task in which the search spaces had only three dimensions (red, green, and blue or L, a, and b) were used to compare different IEAs and search spaces. The creation of facial composites provides an opportunity to compare the performance of different mutation and recombination operators in a more realistic high dimensional task. The second experiment in this chapter compares the performances of two different mutation operators and two different recombination operators. In the third experiment a 'mathematically reduced' face-space in which only the first 12 PCs are used is constructed. The performances of searches using the different variation operators and the large, the human reduced, and the mathematically reduced face-spaces are compared using a task which requires participants to create composites from memory. Creating composites from memory is a without target task and as such the ability of the participants to memorise the target faces will add noise to the data collected.

7.2 Theory

The face model used in the experiments reported in this chapter was created in a similar manner to that laid out by Cootes et al. [20]. The process is outlined briefly here.

Photographs of faces are gathered to be used as a training set. Ideally, the photographs are taken in identical lighting conditions with each face expressing the same neutral expression and looking directly at the camera with no tilting of the



(a) Faces generated at ±3 SDs on the 1-st PC

(b) Faces generated at ±3 SDs on the 30-th PC

Figure 7.1: The pairs of faces at ±3 SDs on the 1-st and 30-th PCs

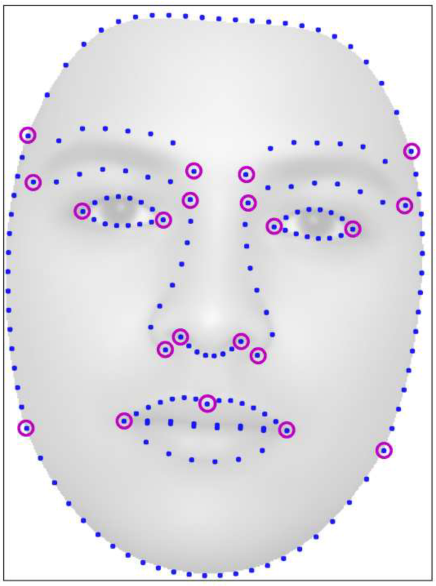


Figure 7.2: Example of a landmarked face image

head. The photographs used in the training set to build the face model used in the experiments reported in this chapter is composed of 27 males and 63 females of various ages.

A number of points common to all of the photographs are landmarked. These common points are facial features such as the corners of the eyes, the bottom of the chin, and the outline of the eyebrows. An example of a landmarked face is given in Figure 7.2. The set of landmarks on a particular face form a face shape and therefore there is one face shape for each face in the training set. In the face model created for the experiments in this chapter each face shape consists of 190 two-dimensional landmarks and thus the resulting shape model has 380 dimensions.

The mean face shape $\bar{\mathbf{s}}$ is found by aligning the face shapes using an iterative Procrustes alignment process [105]. PCA is used to reduce the 380-dimensional shape model to a smaller number of dimensions. Any face shape $\mathbf{s}$ can be approximated to $\widehat{\mathbf{s}}$ in the shape model using

$$\widehat{\mathbf{s}} = \mathbf{P}_s \mathbf{b}_s + \bar{\mathbf{s}} \tag{7.1}$$

where $\mathbf{P}_s$ are the PCs of the shape model ordered from most important (the PCs which account for the most variance in the data) to least important and $\mathbf{b}_s$ are parameters that determine how the shape PCs are combined to make the face shape.

In order to create the texture model, each photograph in the training set is partitioned using its landmark points and Delaunay triangulation [25]. Piecewise affine transforms are used to warp the texture information (the pixel values of the photographs in the training set) from each training photograph's face shape to the mean face shape to form normalised texture patterns. PCA is then used to find a texture model with fewer dimensions than that formed by the tens of thousands of pixels within each normalised texture pattern. As with the face shapes, any face texture $\mathbf{g}$ may be approximated using

$$\widehat{\mathbf{g}} = \mathbf{P}_g \mathbf{b}_g + \bar{\mathbf{g}}. \tag{7.2}$$

where $\mathbf{P}_g$ are the PCs of the face texture ordered from the most important to least important and $\mathbf{b}_s$ are parameters that determine how the texture PCs are combined to make the face texture.

Finally, a face-model is created from the combined shape and texture models using PCA to further reduce the number of dimensions in the final face-space. The appearance model parameters, $\mathbf{c}$, of any face can be approximated to $\widehat{\mathbf{c}}$ using

$$\widehat{\mathbf{c}} = \mathbf{Q}^T \begin{bmatrix} w\mathbf{b}_s \\ \mathbf{b}_g \end{bmatrix} \equiv \mathbf{Q}^T \begin{bmatrix} w\mathbf{P}_s^T(\widehat{\mathbf{s}} - \bar{\mathbf{s}}) \\ \mathbf{P}_g^T(\widehat{\mathbf{g}} - \bar{\mathbf{g}}) \end{bmatrix} \tag{7.3}$$

where $\mathbf{Q}$ are the appearance PCs of the training set ordered from the most important to the least important and w is a weighting scale that scales the shape parameters such that equal significance is assigned to shape and texture.

New faces can be created by setting the values of an n-dimensional parameter vector $\mathbf{c}$ and performing the above process in reverse. Starting with the extraction of $\mathbf{b}$

$$\mathbf{b} = \sum_{i=1}^{n} \mathbf{q_i} c_i \tag{7.4}$$

where $\mathbf{q}_i$ is the i-th column of matrix $\mathbf{Q}$ in Equation 7.3. The shape and texture parameters $\mathbf{b}_s$ and $\mathbf{b}_g$ are extracted from $\mathbf{b}$ and are used in Equations 7.1 and 7.2 to find the shape parameters $\mathbf{s}$ and texture parameters $\mathbf{g}$. The pixel intensities in $\mathbf{g}$ are rearranged into a two-dimensional (or three-dimensional for colour images) array of pixels which then form an intermediate face image with the mean face shape. At this stage the face texture could be warped according to the shape model parameters $\mathbf{s}$ and displayed. An example of this is shown in Figure 7.3(a). Preliminary testing revealed that aspects of the edge of the face image which were due to the landmarking process had a dominant unwarranted effect on the perception of the face. To counter this effect the generated face texture was inserted into a softened background. The background was calculated as the mean of the training set and included the average hair style The background is shown in Figure 7.3(b). Face textures generated by the face model were inserted into the background. The perimeter of face texture was then blended into the background (as shown in Figure 7.3(c)). The resulting image was subsequently warped according to the shape parameters, $\mathbf{s}$, to form the final face image as shown in Figure 7.3(d).

7.3 Experiment 1: Identifying the most perceptually significant PCs

Thirty-two participants were used in this experiment. The majority of the participants were undergraduate students in the School of Physical Sciences at the University of Kent. The remainder of the participants comprised postgraduate students and staff. The participants were instructed to sort pairs of faces that were created in the large (30-dimensional) PCA face-space into their order of dissimilarity. These orderings were used to establish the most appropriate PCs to be used in the human reduced face-space.

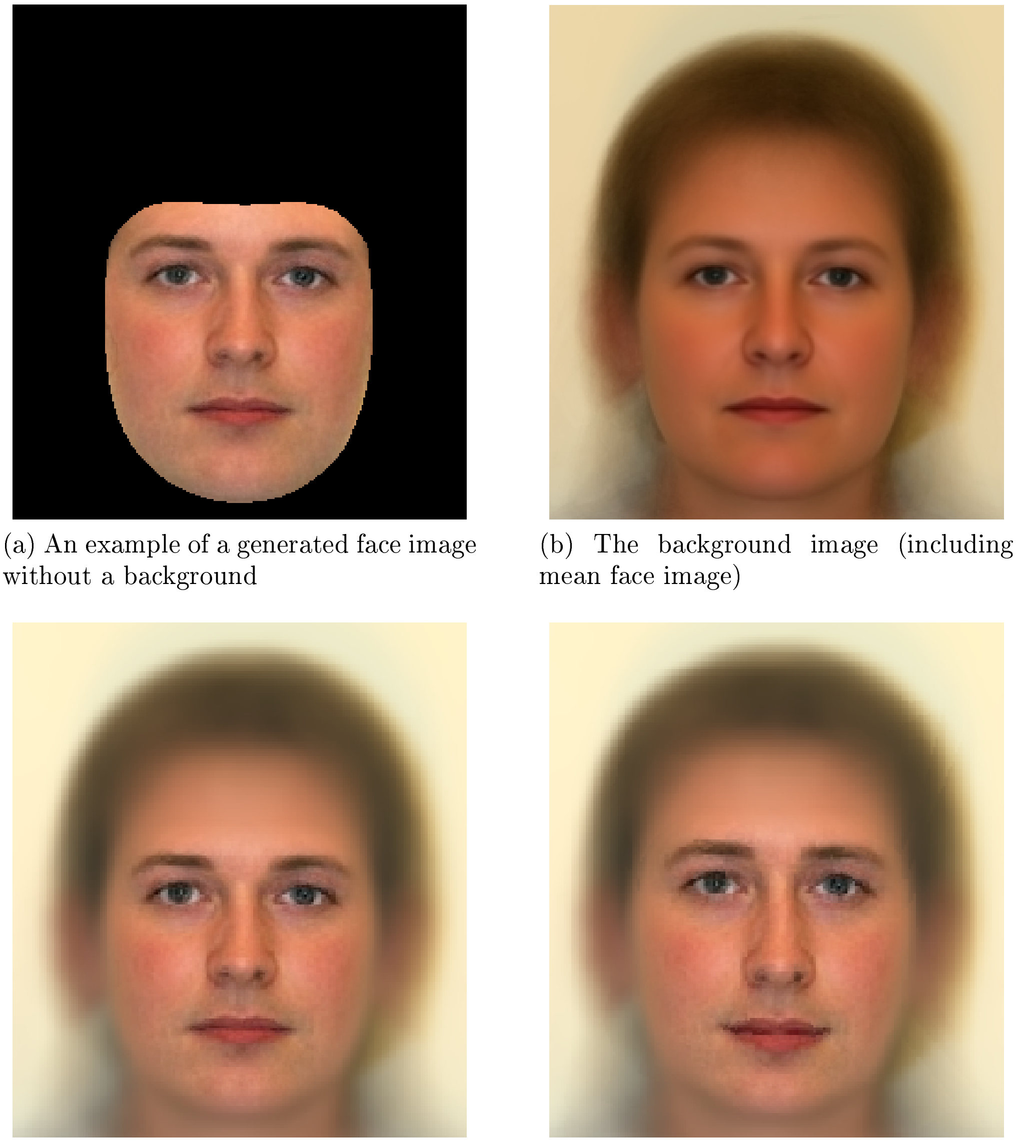
7.3.1 Method

Set-up

Thirty pairs of faces were generated from the large (30-dimensional) PCA face-space. If a face's representation in the 30-dimensional large face-space is given by $\mathbf{c} = (c_1, c_2, \ldots, c_i, \ldots, c_{30})$, then each pair of points $(\mathbf{c_{+k}}, \mathbf{c_{-k}})$ has coordinates defined by

$$c_{\pm i} = \begin{cases} \pm 3 & \text{SDs if } i = k \\ 0 & \text{SDs otherwise} \end{cases} \tag{7.5}$$

At the start of each run of the experiment the pairs of faces were arranged randomly in a grid six pairs high by five pairs wide (Figure 7.4). The participants were instructed to group the twelve pairs of faces which 'exhibited the most within pair dissimilarity' (Figure 7.5). Once the participants had done this they were instructed to sort the twelve pairs of faces from the most similar to the least similar (Figure 7.6). When they had finished, the order of the pairs of faces was recorded using the numbers on the backs as identifiers (Figure 7.7). The participants were not asked to sort all of the pairs of faces into order because preliminary testing revealed that after the first fifteen or so most dissimilar pairs had been ordered it became



(a) An example of a generated face image without a background

(b) The background image (including mean face image)

(c) Face image on background after blurring but before warping

(d) Final face image on background after warping

Figure 7.3: Adding a background to the face images to remove peripheral landmarking artefacts

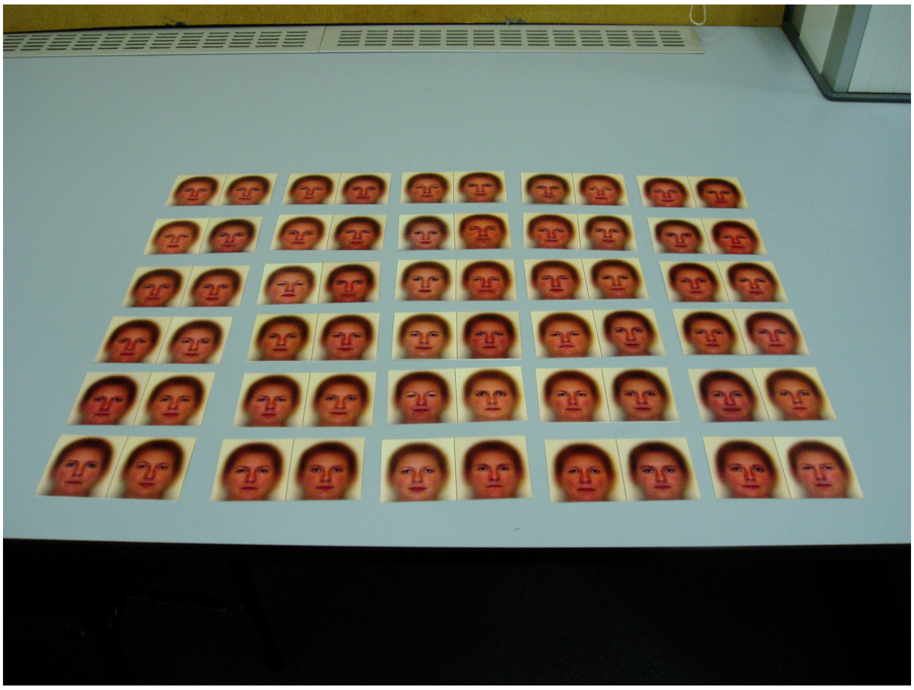


Figure 7.4: Initial layout of the sorting task

difficult to decide which pairs were most different and thus the data gathered for the less dissimilar pairs would have been very noisy.

7.3.2 Results

Each pair of faces was awarded a score of 12 for each occasion they were selected as the most dissimilar pair, 11 for each occasion they were selected as the second most dissimilar pair and so on. The total scores for all of the pairs are given in Table 7.1. The PCs are listed in order of perceived difference in Table 7.2.

The Spearman correlation coefficient between the mathematical ordering of the PCs and the human ordering of the PCs is $\rho = 0.8260$, $p < 0.001$. Whilst it can be seen that the correlation is strong, there are noticeable differences between the perceptual and variance based orderings. Of particular note is the high importance placed on the 15-th PC by the participants and the relatively low importance placed on the 8-th PC. The pairs of faces corresponding to the 8-th and 15-th PCs are shown in Figure 7.8

Whilst the PCA ordering of the PCs is different to that of the participants', the goal of this part of the experiment was to decide which PCs should be used to build the human reduced face-space. From Table 7.2, it can be seen that the twelve most significant PCs perceptually are $1, 2, 3, 4, 5, 6, 7, 9, 13, 14, 15$, and 18. These are the

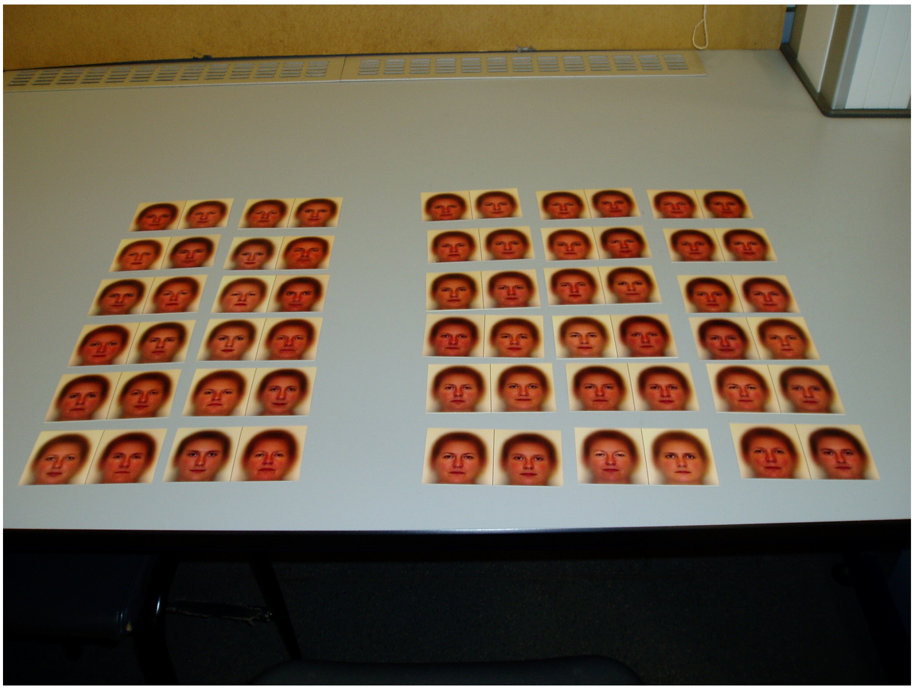


Figure 7.5: The twelve pairs of faces that exhibit the most within pair dissimilarity have been identified (to the left of the picture)

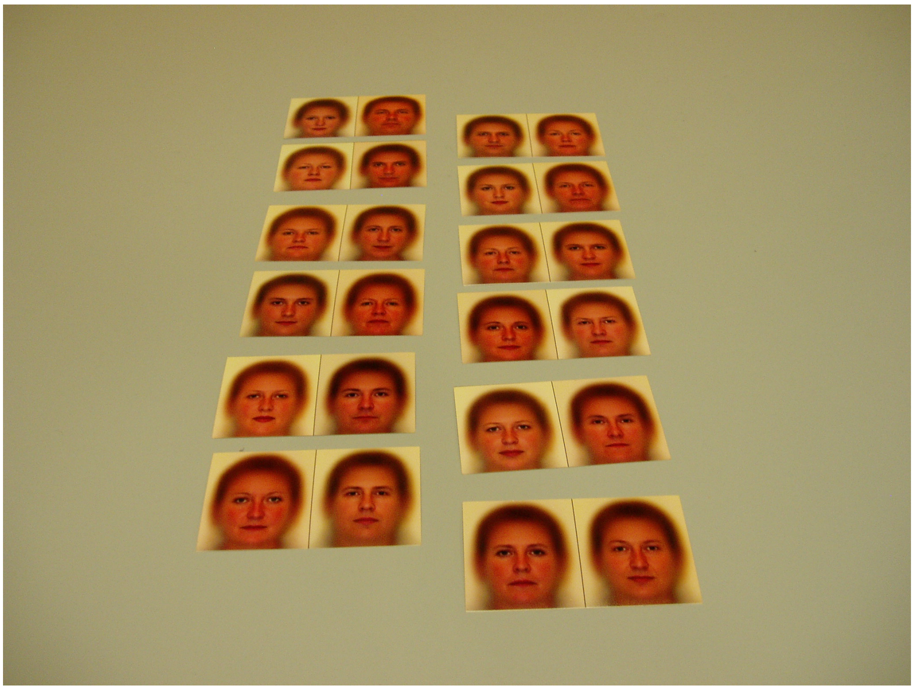


Figure 7.6: The twelve most dissimilar pairs sorted by within pair dissimilarity, top-left to bottom-left, top-right to bottom-right

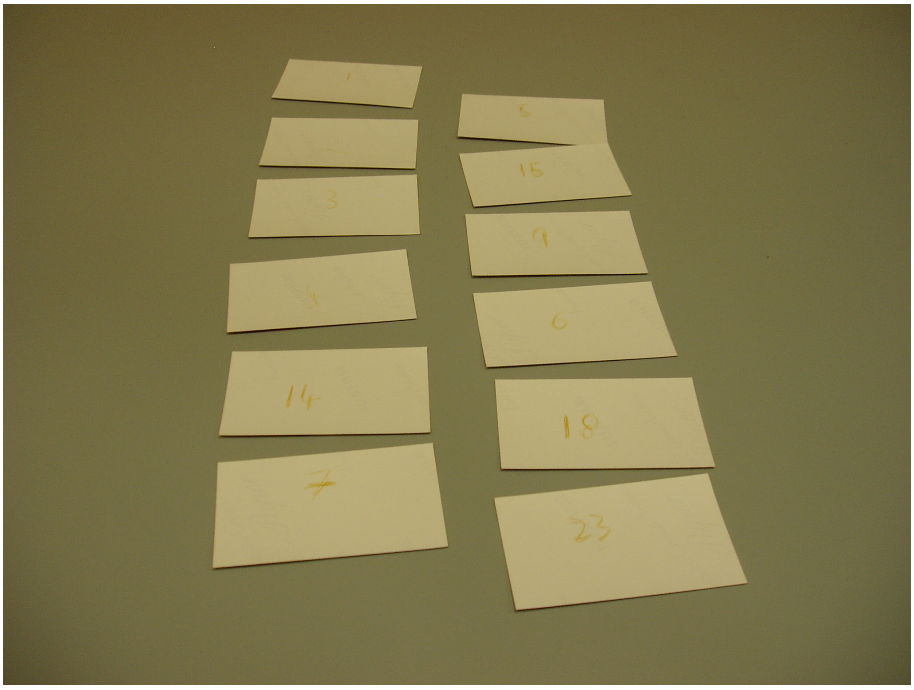
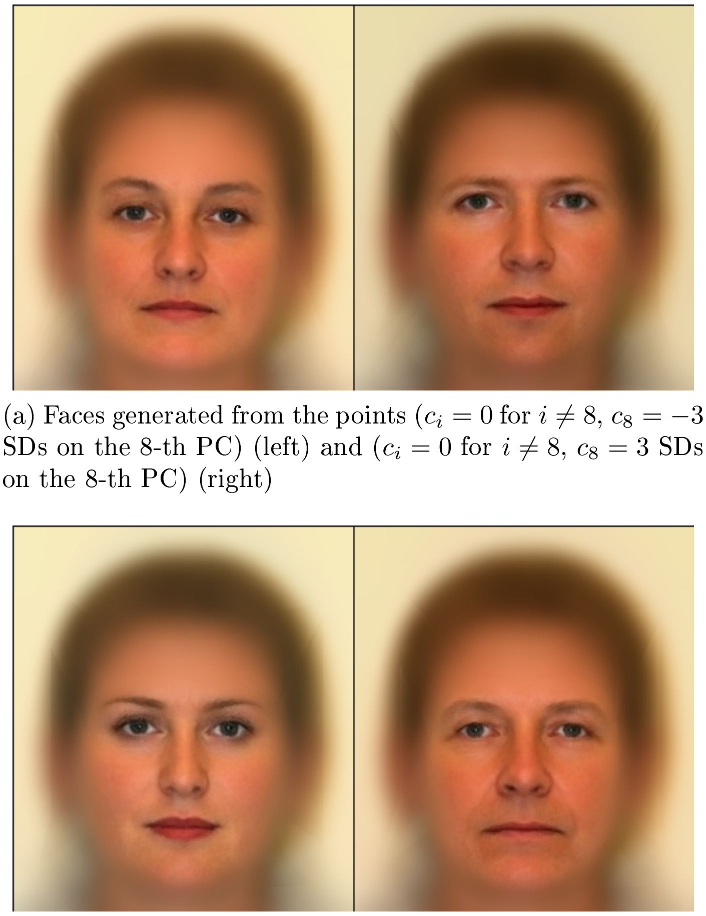


Figure 7.7: The reverse side of the pairs of faces. The numbers identify which PC each pair comes from

Table 7.1: Total dissimilarity scores for the pairs of faces

PC	Score	PC	Score	PC	Score
1	371	11	22	21	26
2	326	12	4	22	6
3	293	13	88	23	44
4	175	14	154	24	2
5	265	15	194	25	28
6	70	16	28	26	4
7	177	17	19	27	0
8	37	18	60	28	4
9	59	19	5	29	0
10	32	20	0	30	3



(a) Faces generated from the points ($c_i = 0$ for $i \neq 8$, $c_8 = -3$ SDs on the 8-th PC) (left) and ($c_i = 0$ for $i \neq 8$, $c_8 = 3$ SDs on the 8-th PC) (right)

(b) Faces generated from the points ($c_i = 0$ for $i \neq 15$, $c_{15} = -3$ SDs on the 15-th PC) (left) and ($c_i = 0$ for $i \neq 15$, $c_{15} = 3$ SDs on the 15-th PC) (right)

Figure 7.8: Example of PCs in which human evaluation and mathematical ordering disagreed. The 15-th PC was considered more important than the 8-th

Table 7.2: PCs ranked according to human perception of importance

Rank	PC	Rank	PC	Rank	PC
1	1	11	18	21	22
2	2	12	9	22	19
3	3	13	23	23	12
4	5	14	8	24	26
5	15	15	10	25	28
6	7	16	16	26	30
7	4	17	25	27	24
8	14	18	21	28	20
9	13	19	11	29	27
10	6	20	17	30	29

PCs to be used to build the human reduced face-space. It can be seen that eight of the twelve PCs in the human reduced face-space are in the first twelve PCs of the large PCA face-space.

7.4 Experiment 2: Comparison of recombination and mutation methods in the facial composite task

Two recombination methods were compared: uniform crossover and arithmetic crossover, and two mutation methods were compared: Gaussian addition and Gaussian replacement.

7.4.1 The mutation and recombination methods used

Uniform crossover is the recombination method used for the simple IGA in Chapters 5 and 6. In the implementations used in this thesis, two parents are used to create one offspring. The value of each gene in an offspring has an equal chance of coming from either parent. In the implementation of arithmetic crossover used in this experiment the value of each gene in an offspring is the mean of the values for that gene in the parents. Two offspring produced by the same parents in arithmetic crossover will always be identical (before mutation). However, two offspring by the same parents in uniform crossover are likely to be different. It was expected that uniform crossover is better at maintaining diversity in the population whereas arithmetic crossover is better at aiding convergence.

Gaussian addition is the name given to the mutation method used in all of the work in this thesis to this point. After recombination (if any) all of the offspring's gene values are mutated by the addition of a zero-mean Gaussian distributed random number with a standard deviation set either by the algorithm (as in Chapters 3 and 4) or by the user (as in Chapters 5 and 6). In Gaussian addition the mutated gene value is given by

$$c_i' = c_i + \sigma_i \cdot m \cdot N(0,1) \qquad (7.6)$$

where σ_i is the standard deviation of the i-th PC, m is the mutation factor set by the user on the interface, and $N(0,1)$ is a random number from the Gaussian distribution. *Gaussian replacement* is the name given in this work to an analogous method to the uniform mutation operator. In uniform mutation, there is some probability p_m for each gene in an offspring's genotype that its value will undergo be replaced by a uniformly distributed random value where $c_i, c_i' \in [\text{Lower limit}, \text{Upper limit}]$. The Gaussian replacement operator is similar except that c_i' is a random number taken from $N(0,1)$. c_i' has the further restriction that it is bounded by the hyperrectangle which designates the edge of the search space, that is $c_i, c_i' \in [-2.5, 2.5]$ SDs. Gaussian replacement is used instead of uniform mutation because whilst having a single coordinate at the edges of the bounding hyperrectangle will still produce plausible faces, having even a few such values will produce faces with artefacts. For example, Figure 7.9 shows a composite face with the genotype ($c_2 = 2$ SDs, $c_3 = 2$ SDs, $c_5 = -2$ SDs, $c_{13} = 2$ SDs, $c_i = 0$ SDs otherwise). Gaussian replacement tends to mutate the offspring toward the centre of the face-space. The value of p_r used in this experiment has a minimum of 0 and a maximum of $5/n$ where n is the number of dimensions in the face-space. This may seem high but it serves to allow the search to be taken to appropriate parts of the face-space quickly. The participant could reduce the mutation slider when they thought the composites were looking more like the target face.

The 12-dimensional human reduced face-space was used in this experiment. This face-space was chosen because it is not thought that the face-space used would have an effect on the relative performances of the different recombination and mutation operators. It was assumed, however, that searching a lower-dimensional face-space would lead to a face match more quickly that searching in a high-dimensional face-

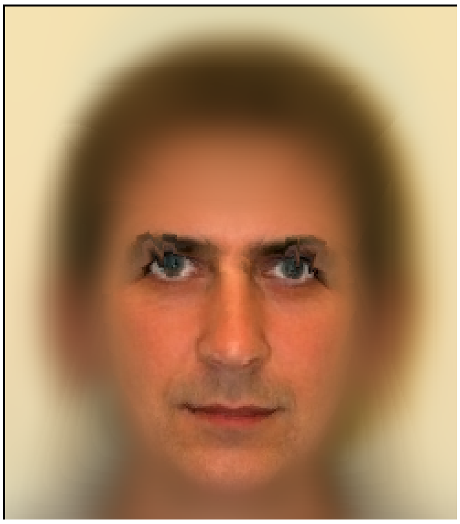


Figure 7.9: Example of artefacts in a facial composite. In this case it can be seen that the region between the upper eyelids and the eyebrows have non-plausible colouration

space and thus induce less fatigue in the participants. The validity of this assumption was tested in the third experiment of this chapter (see Section 7.5).

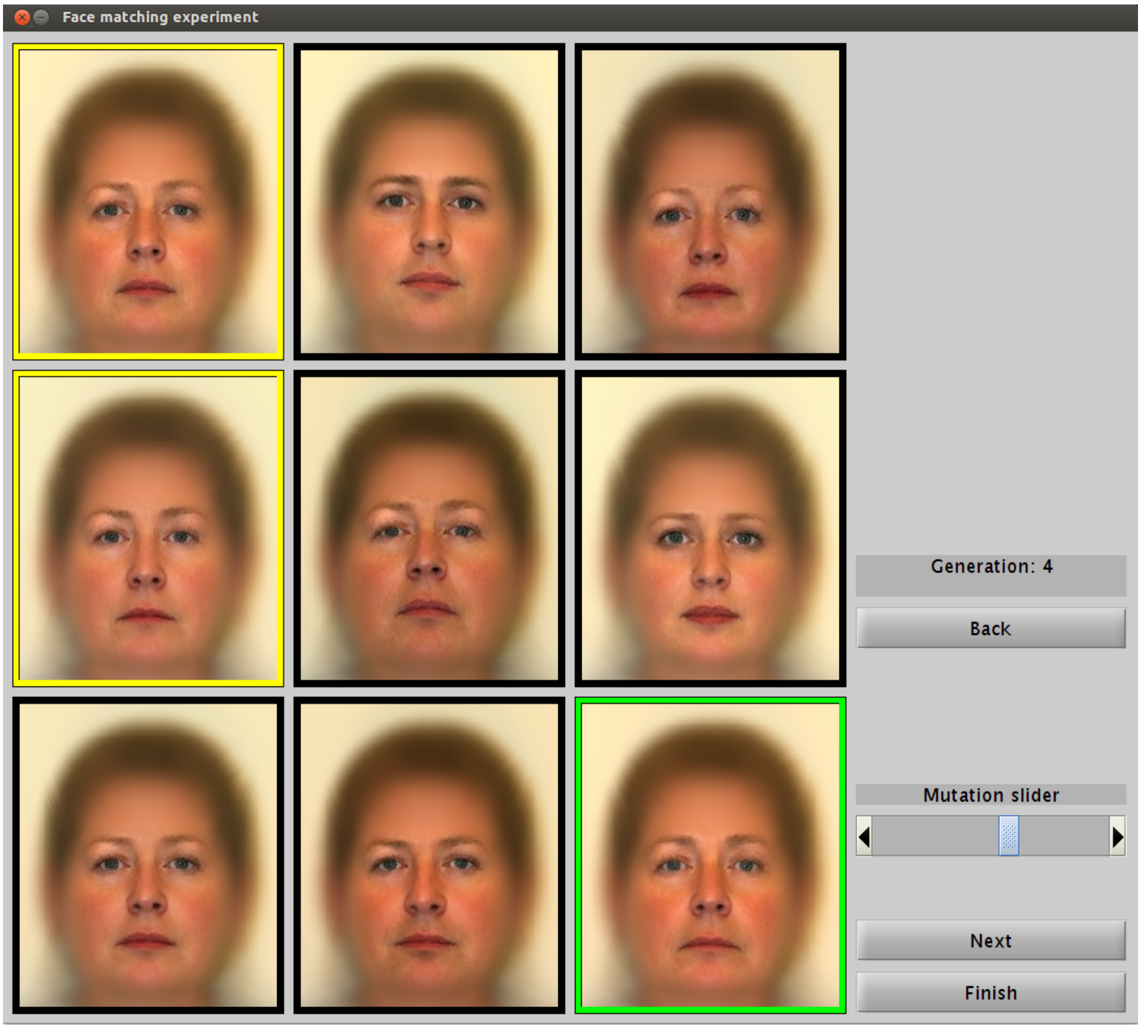
7.4.2 Method

The interface

The interface used for the experiment was adapted from that used for the IEAs runs of the contrast enhancement experiment of Chapter 6. The population consisted of a 3×3 grid of composites. To avoid the problem described in Section 5.4.3 whereby participants did not make effective use of the mutation slider the slider was automatically decremented by 0.03 (the range of the slider is $[0, 1]$) per generation. Trial runs of the experiment suggested that the value of the mutation parameter should change linearly with the slider position as was the case for the experiment reported in Chapter 5 as opposed to the square of the slider's value as was the case for the experiment reported in Chapter 6. A 'back' button was added to the interface which enabled the participant to go back to the previous generation and make alternative selections or adjust the mutation slider if they were not satisfied with the current generation. A screenshot of the interface is given in Figure 7.10.

At the start of each run the participants had to remember the target face and then try to recreate the face from memory using the facial composite process. At the start of each run the target face was displayed on the monitor for 10 seconds.

Figure 7.10: Screenshot of the interface for the facial composite tasks



The target face was not shown to the participants again until the end of the run.

Every generation the participants would choose the composite that best resembled the target face and select it by clicking on it using the left mouse button. The participants also had the option of selecting any composites that they thought were also good, anywhere from zero to eight. This could be achieved by clicking on the composites using the right mouse button. A green border was placed around the composite the participant preferred, a yellow border for those composites the participant thought were also good, and a black border for those composites that were not selected. Once they were satisfied that they had selected the best match and any other matches they considered to be good, the participant would go to the next generation by pressing the 'Next' button. The selected composite was carried forward into the next generation. The preferred and other selected composites were used in creating the next generation of composites. The participants would continue the process until they thought they had successfully recreated the target face, or until they thought no further improvement was possible, by clicking on the 'Finish' button.

Test set-up

Fifteen participants were used in this experiment. The majority of the participants were postgraduate students in the School of Physical Sciences at the University of Kent, the remainder of the participants were staff or undergraduate students. Participants required only a basic level of computer literacy. Testing each combination of recombination and mutation operator required $2 \times 2 = 4$ runs per participant.

At the start of the experiment the participants were read a script telling them about the task and how to use the interface. They then did a practice run using the recombination and mutation combination they were going to be using for the first run of the recorded part of the experiment. The target face for the practice run was the mean face, that is, the face whose genotype is located at the centre of the face-space. The target faces were chosen such that they were equidistant from the centre of the face-space. The genotypes of the target faces, as represented in the large face-space, are given by the following equations. In each case i is the PC (dimension) number ($i = 1, 2, \ldots, 29, 30$) and the location of the point relative to each PC is given in the number of standard deviations (SDs) that the point is along that PC.

Face 1

$$c_i = \begin{cases} 1.25 & \text{SDs for } i = 1, 2, 3, 4, 5, 6, 7, 9, 13, 14, 15, 18 \\ 0 & \text{otherwise} \end{cases} \tag{7.7}$$

Face 2

$$c_i = \begin{cases} -1.25 & \text{SDs for } i = 1, 2, 3, 4, 5, 6, 7, 9, 13, 14, 15, 18 \\ 0 & \text{otherwise} \end{cases} \tag{7.8}$$

Face 3

$$c_i = \begin{cases} 1.25 & \text{SDs for } i = 1, 3, 5, 7, 13, 15 \\ -1.25 & \text{SDs for } i = 2, 4, 6, 9, 14, 18 \\ 0 & \text{otherwise} \end{cases} \tag{7.9}$$

Face 4

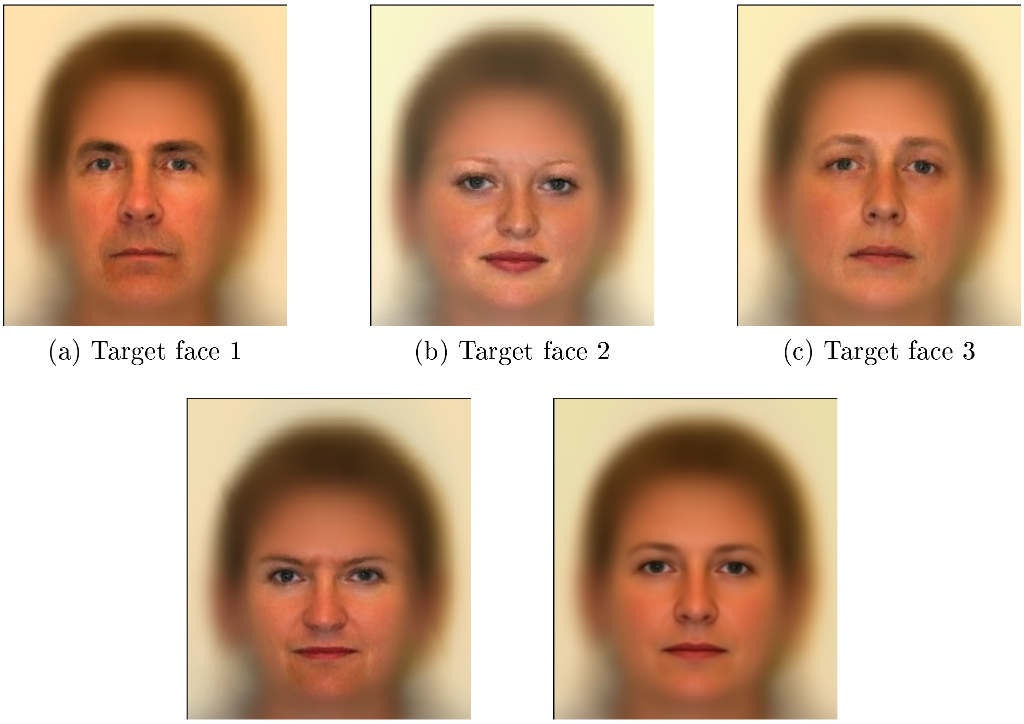
$$c_i = \begin{cases} -1.25 & \text{SDs for } i = 1, 3, 5, 7, 13, 15 \\ 1.25 & \text{SDs for } i = 2, 4, 6, 9, 14, 18 \\ 0 & \text{otherwise} \end{cases} \tag{7.10}$$

Practice face

$$c_i = 0 \text{ for all } i \tag{7.11}$$

The faces themselves are presented in Figure 7.11.

The initial population was designed to be roughly evenly distributed in the human reduced face-space. To start with, 1000 points were generated at random in the human reduced face-space. The points were generated using a twelve-dimensional uniform distribution with the limits being at ±2.5 SDs on each axis. K-means clustering [78] using the squared Euclidean distance as the distance metric was used to group the generated points into nine clusters. Normally, it is the grouping of the generated points that is of interest however in this application it is the centroids, the mathematical centres of the clusters, that were required. The centroids of the nine clusters were used as the genotypes of the initial population of faces.



(a) Target face 1 (b) Target face 2 (c) Target face 3

(d) Target face 4 (e) Practice face

Figure 7.11: The target faces used in the second experiment

```
1 -- Very poor likeness between faces
2 or 3 -- Few similarities
4 or 5 -- Some similarities
6 or 7 -- Many similarities
8 or 9 -- Faces could be easily confused
10 - Faces are identical
```

Figure 7.12: Scale used for similarity rating of facial composites

Data gathered

At the end of every run, the participants were shown the composite they had just created and were asked to rate its similarity to the target (as they remembered it) with reference to the scale[1] shown in Figure 7.12. Immediately after rating their composite the participants were shown the target face alongside their composite and asked to rate the similarity between their composite and the target again.

Three sets of objective data were gathered: the time taken to create the composites, the number of generations it took to create the composites, and the number of times the back button was used. The number of times the back button was used may

[1] This scale was presented by Frowd in [34]

provide an indication that the participants are having difficulty with the interface or that the mutation or recombination operators are inappropriate for the task.

7.4.3 Results

Comparisons between the operators

The means and standard deviations of the measured variables (number of generations, time taken, number of times the back button was used, participant rating of their composite without reference to the target, participant rating of the their composite with reference to the target) are given in Table 7.3. Each of the measured variables were subjected to ART with two-way ANOVA (see Section 2.2.4) having two mutation operators (Gaussian additive and Gaussian replacement) and two recombination operators (uniform crossover and arithmetic crossover) (Table 7.4). It can be seen that the main effects of mutation operator and recombination operator were not significant for any of the measured variables, nor was the interaction of the two operators significant.

Correlations between the measured variables

The Spearman's correlation coefficients between the measured variables were calculated for each combination of mutation operator and crossover operator (Table 7.5). It can be seen that there was a strong correlation between the number of generations and the time taken, which is to be expected given that each new population of faces takes about three seconds to create. For two of the four operator pairs there was a statistically significant correlation between the with comparison and the without comparison similarity ratings, and although not statistically significant there was a moderate correlation for the other two operator pairs. Comments from participants during experimentation suggests that they had some idea of how well they had remembered the target faces and this would influence how they rated the composites without comparison to the target; that is, they were rating their recollection of the target face rather than the composite. The correlation between the without comparison and with comparison ratings can be interpreted as a measure of how correct the participants were in general with regards to their abilities to remember the target faces. There was a statistically significant negative correlation between the time taken and the without comparison rating for one of the operator pairs and, with reference to the remaining operator pairs, reason to conclude that there was a

Table 7.3: Means (standard deviations) of the measured variables in the comparison of the Gaussian replacement and Gaussian mutation operators and the uniform crossover and arithmetic crossover recombination operators in the creation of facial composites

Mutation	Recombination	Generations	Back count	Time taken	Without rating	With rating
Gaussian replacement	uniform	10.6 (5.10)	0.73 (1.33)	195s (91.5s)	6.27 (1.22)	4.40 (2.10)
Gaussian replacement	arithmetic	12.5(8.64)	0.47 (0.74)	222s (155s)	5.47 (2.00)	5.07 (2.19)
Gaussian additive	uniform	11.5 (4.73)	0.87 (1.41)	220s (71.1s)	6.07 (1.03)	4.60 (2.41)
Gaussian additive	arithmetic	9.73 (2.49)	0.47 (0.64)	188s (66.2s)	6.07 (1.49)	4.40 (2.32)

Table 7.4: ART with two-way ANOVA for the measured variables in the comparison of the Gaussian replacement and Gaussian mutation operators and the uniform crossover and arithmetic crossover recombination operators in the creation of facial composites

	Mutation		Recombination		Interaction	
Variable	$F(1,56)$	p-value	$F(1,56)$	p-value	$F(1,56)$	p-value
Generations	0.025	0.874	0.041	0.840	0.826	0.367
Back Count	0.153	0.670	0.368	0.547	0.055	0.816
Time taken	0.427	0.516	0.553	0.460	0.851	0.360
Without comparison rating	0.132	0.718	0.510	0.478	0.771	0.384
With comparison rating	0.425	0.517	0.214	0.645	0.571	0.529

moderate negative correlation between the time taken and the without target comparison. This correlation can be attributed to the participant's memory of the face too; the more confidence a participant had in their ability to remember the target face the more generations it would take to achieve a satisfactory composite. There was one statistically significant correlation between time taken and the with target rating but as two of the ρ values were nearly zero it cannot be concluded that there was a correlation between the time taken and the with target ratings.

7.5 Experiment 3: Comparison of face-spaces in the facial composite task

Three face-spaces were compared in this experiment: A face-space constructed from the first 30 PCs of the PCA analysis (the large face-space), a face-space constructed from the first twelve PCs (the mathematically reduced face-space), and a face-space constructed form the twelve most perceptually important PCs (the human reduced face-space). The results of the second experiment showed no significant difference between the operators on any of the recorded measures. Arithmetic crossover and Gaussian additive mutation were the operators chosen for this experiment. The face-spaces were reduced in size for this experiment. This was done as a consequence of trying to reduce the number of artefacts in the composites and preventing the more unrealistic faces from being generated. As well as the bounding hyperrectangle whose faces were perpendicular to ±2.5 SDs on each PC, the genotypes generated were also subject to the condition

$$\sqrt[3]{\sum_{i}^{n} |c_i|^3} < 3.5 \tag{7.12}$$

Where n is the number of dimensions in the search space.

7.5.1 Method

This experiment was nearly identical to the assessment of the operators in the second experiment. The interface was identical though a text box advising participants to take a few moments to rest was displayed after the composites were rated at the end of each run. As there were only three test conditions (large face-space, human reduced face-space, and mathematically reduced face-space) each participant

Table 7.5: Correlations between the measured variables in the comparison between mutation and crossover operators in IGAs applied to a facial composite task

		Mutation and	Spearman's correlation coefficients				
		crossover operators	Generations	Back count	Time taken	Without target rating	With target rating
p-values	Generations	Replacement uniform	—	0.497	0.891	-0.074	-0.180
		Replacement arithmetic	—	0.469	0.771	-0.282	0.113
		Addition uniform	—	0.429	0.687	-0.280	-0.068
		Addition arithmetic	—	0.027	0.748	-0.191	0.242
	Back count	Replacement uniform	0.060	—	0.245	-0.133	-0.157
		Replacement arithmetic	0.078	—	0.306	-0.191	-0.425
		Addition uniform	0.111	—	0.242	-0.012	-0.288
		Addition arithmetic	0.924	—	-0.235	0.452	0.412
	Time taken	Replacement uniform	< 0.001	0.379	—	-0.114	-0.236
		Replacement arithmetic	< 0.001	0.267	—	-0.380	-0.034
		Addition uniform	< 0.001	0.384	—	-0.536	-0.552
		Addition arithmetic	< 0.001	0.399	—	-0.457	-0.018
	Without target rating	Replacement uniform	0.793	0.637	0.687	—	0.683
		Replacement arithmetic	0.309	0.495	0.162	—	0.351
		Addition uniform	0.313	0.967	0.040	—	0.508
		Addition arithmetic	0.496	0.091	0.087	—	0.643
	With target rating	Replacement uniform	0.521	0.577	0.396	0.005	—
		Replacement arithmetic	0.688	0.114	0.903	0.200	—
		Addition uniform	0.810	0.299	0.033	0.053	—
		Addition arithmetic	0.385	0.127	0.949	0.010	—

performed two runs for each of the test conditions so that they performed $2 \times 3 = 6$ runs. The initial populations for each of the face-spaces were constructed in the same way as that for the second experiment. The target faces were chosen to be equidistant from the centre of the 30-dimensional face-space. The genotypes of the target faces, as represented in the large face-space, are given by the following equations. In each case i is the PC (dimension) number ($i = 1, 2, \ldots, 29, 30$) and the location of the point relative to each PC is given in the number of standard deviations (SDs) that the point is along that PC.

Face 1

$$c_i = \begin{cases} 0.75 & \text{SDs for for all } i \end{cases} \tag{7.13}$$

Face 2

$$c_i = \begin{cases} -0.75 & \text{SDs for for all } i \end{cases} \tag{7.14}$$

Face 3

$$c_i = \begin{cases} 0.75 & \text{SDs for odd } i \\ -0.75 & \text{SDs for even } i \end{cases} \tag{7.15}$$

Face 4

$$c_i = \begin{cases} 0.75 & \text{SDs for } i = 1,2,5,6,9,10,13,14,17,18,21,22,25,26,29,30 \\ -0.75 & \text{SDs for } i = 3,4,7,8,11,12,15,16,19,20,23,24,27,28 \end{cases} \tag{7.16}$$

Face 5

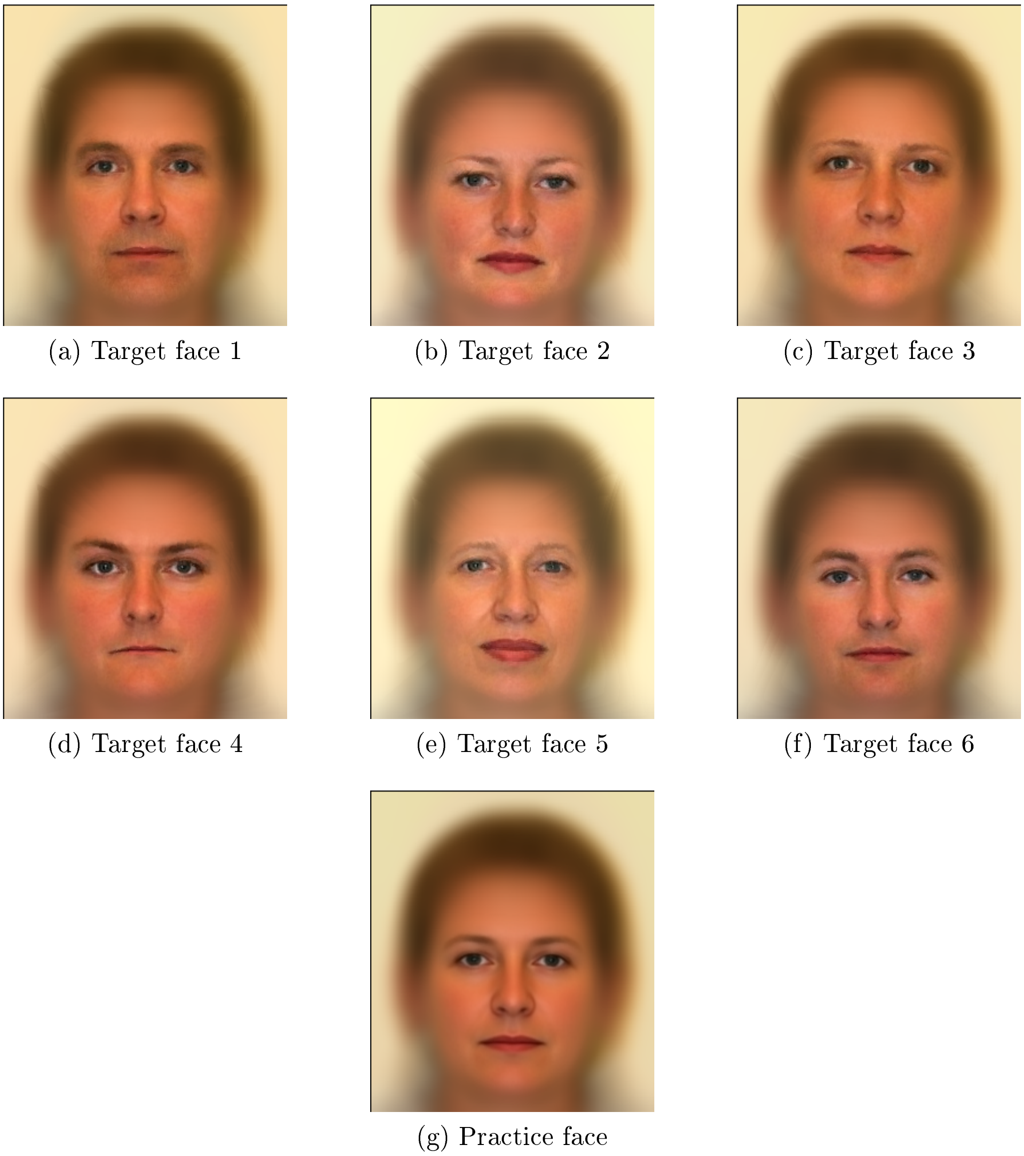
$$c_i = \begin{cases} -0.75 & \text{SDs for } i = 1,2,5,6,9,10,13,14,17,18,21,22,25,26,29,30 \\ 0.75 & \text{SDs for } i = 3,4,7,8,11,12,15,16,19,20,23,24,27,28 \end{cases} \tag{7.17}$$

Face 6

$$c_i = \begin{cases} -0.75 & \text{SDs for odd } i \\ 0.75 & \text{SDs for even } i \end{cases} \tag{7.18}$$

Practice face

$$c_i = 0 \text{ for all } i \tag{7.19}$$



(a) Target face 1

(b) Target face 2

(c) Target face 3

(d) Target face 4

(e) Target face 5

(f) Target face 6

(g) Practice face

Figure 7.13: The target faces used in the face-space experiment

The target faces themselves are presented in Figure 7.13.

Twenty-one participants were used for this experiment. The majority of the participants were postgraduate students in the School of Physical Sciences at the University of Kent, with the remainder being staff or undergraduate students. As in the previous experiment, the genotypes of the target faces were chosen to be equidistant from the centre of the face-space. The same data were gathered as in the second experiment of this chapter.

7.5.2 Results

Comparisons between the treatments

Each participant performed the face matching task twice using each of the large (30-dimensional), human reduced, and mathematically reduced face-spaces.The average of each of the measured variables (number of generations, time taken, number of times the back button was used, participant rating of their composite without reference to the target, participant rating of the their composite with reference to their target) over the two runs was found. These averages were treated as a single run for the purposes of the calculating the means and standard deviations, so that each participant was treated as having performed only one run using each of the face-spaces. The means and standard deviations of the measured variables over all of the runs for each of the algorithms are presented in Table 7.6.

Performing Friedman's test on each of the measured variables showed that the differences between the face-spaces were not significant for any of the measured variables (number of generations: $\chi^2(2) = 2.11, p = 0.349$, number of times the 'back' button was used: $\chi^2(2) = 0.54, p = 0.765$, time taken: $\chi^2(2) = 2.14, p = 0.343$, without comparison rating: $\chi^2(2) = 2.37, p = 0.306$, and with comparison rating: $\chi^2(2) = 0.71, p = 0.700$).

Correlations between the measures

The Spearman's correlation coefficients between the measured variables were calculated. The correlation coefficients were calculated for each of the face-spaces. Table 7.7 shows the correlation coefficients and their p-values. As with the operator comparison experiment, there was a very strong correlation between time taken and the number of generations, and evidence of a moderate correlation between the with and without target ratings. The one statically significant and two moderate correlations between time taken and the number of times the back button was used indicates a moderate correlation between these two measured variables. A similar correlation is observed between the number of generations and the number of times the back button was used. A simple explanation is that the longer a run took the more generations the run had gone to and hence the more likely it was that the back button was used during the run. There was a significant negative correlation between the without target rating and the time taken for the large face-space and weaker correlations for the other two face-spaces. As before, it is reasonable to conclude that

Table 7.6: Means (standard deviations) of the measured variables in the comparison of the large, human reduced and mathematically reduced face-spaces in the creation of facial composites

Face-space	Generations	Back count	Time taken	Without target rating	With target rating
Large	10.7 (4.73)	0.50 (0.55)	205s (80.3s)	5.81 (1.13)	4.10 (1.25)
Human reduced	9.38 (4.31)	0.36 (0.42)	186s (91.8s)	6.02 (1.08)	3.95 (1.33)
Mathematically reduced	10.5 (4.75)	0.48 (0.56)	193s (85.6s)	5.86 (1.16)	4.12 (1.82)

there is a moderate negative correlation between the without target rating and time taken due to the confidence of the participants' had in their abilities to remember the target faces.

7.6 Conclusion

A human evaluation based reduced face-space for use with an IEA in the creation of facial composites was derived from a larger PCA based face-space. The performances of searches for faces in the human reduced face-space was compared to those of a mathematically reduced face-space and the larger face-space. The human reduced face-space was also used in the comparison between different mutation and recombination operators in the simple IGA.

The prioritisation of the PCs with regards to human evaluation was found to be similar but different to that of the PCA. The human reduced face-space was found to share eight of its twelve PCs with the mathematically reduced face-space.

No significant differences in the performances of the operators was detected. The difficult nature of the facial composite task means that the data collected was noisy. It may be that the choice of mutation and recombination operators could make a difference on a less cognitively demanding task. The lack of detected difference challenges the finding of Oinuma et al. [87] that the choice of recombination operator can make a difference to the performance of an IEA.

No significant differences in the performances of the search spaces was detected. This result suggests that commercial facial composite software such as EFIT-V [115] and EvoFIT [37] can use face-spaces with far fewer dimensions with no loss of performance.

Table 7.7: Correlations between the measured variables in the comparison between face-spaces

			Spearman's correlation coefficients				
		Face-space	Generations	Back count	Time taken	Without target rating	With target rating
p-values	Generations	Large	—	0.537	0.818	-0.377	0.010
		Human reduced	—	0.416	0.845	-0.219	-0.102
		Mathematically reduced	—	0.304	0.851	-0.205	0.081
	Back count	Large	0.012	—	0.341	-0.049	0.347
		Human reduced	0.066	—	0.469	0.028	0.237
		Mathematically reduced	0.181	—	0.380	0.119	-0.031
	Time taken	Large	< 0.001	0.130	—	-0.460	-0.060
		Human reduced	< 0.001	0.032	—	-0.359	-0.074
		Mathematically reduced	< 0.001	0.089	—	-0.196	0.016
	Without target rating	Large	0.092	0.834	0.036	—	0.306
		Human reduced	0.340	0.903	0.110	—	0.467
		Mathematically reduced	0.374	0.607	0.395	—	0.475
	With target rating	Large	0.663	0.123	0.797	0.177	—
		Human reduced	0.660	0.302	0.749	0.033	—
		Mathematically reduced	0.728	0.893	0.946	0.029	—

Chapter 8

Summary and conclusion

In this final chapter, the main results of the thesis are summarised, conclusions are drawn, and suggestions are made for avenues of further investigation.

8.1 Summary

Chapter 1 introduced some of the general problems of using image enhancement and creation tools. The basic difference between an EA an IEAs was stated. A survey of a sample of work in the field of IEAs demonstrated that existing work is deficient when it came to providing robust comparisons between IEAs introduced and existing IEAs or between IEAs and other approaches.

In Chapter 2 the basic parts of an EA were described. The basic parts of an IEA were presented with emphasis on the differences between an EA and an IEA. The concept of fatigue and its influence on the design of IEAs was discussed.

In Chapter 3 an experiment using an IEA, the SMM-IES, to optimise filters for treating salt and pepper noise in colour images is reported. It was found that participant developed filters worked better than each other on the particular images they were developed on and that a previously developed GA filter performed poorly due to the use of the MAE as the fitness function used during its development.

In Chapter 4 a hyperplane-IES was introduced and was compared to the SMM-IES and a dummy-IES using a colour matching task. It was found that the hyperplane-IES and the dummy-IES took more time to achieve a colour match than the SMM-IES. It was concluded that in this instance the interface had a greater effect on the search than the underlying EAs.

Chapter 5 reported a second experiment using the colour matching task. A simple IGA was introduced and compared to the hyperplane-IGA. Use of the CIELAB and sRGB colour spaces as search spaces was also compared. It was concluded that the effort of constructing a perceptually uniform search space was unlikely to be rewarded for the kinds of tasks that IEAs are generally used for.

In Chapter 6 two different methods of setting the input values of image processes were compared: an IEA and a direct interface. Two different contrast enhancement processes were used: a compound process and an intensity transfer function process. It was concluded that for some images the IEA approach leads to better enhanced images than a direct interface and that for some images the compound process can achieve better images than an intensity transfer function process.

In Chapter 7 a series of experiments using IEAs to create facial composites was reported. In the first experiment a human evaluation based search space was created. In the second experiment two mutation operators and two recombination operators were compared. In the third experiment the human evaluation based search space was compared to two others. No differences were found between the performances of the operators or the search spaces. It was concluded that this lack of difference was due to the difficult nature of the task.

8.2 Conclusion

The poor performance of the filter developed using an EA based training image approach [74] in Chapter 3 demonstrates that any IQMs used to measure the quality of images need have their suitability for the task evaluated before they are employed. The use of an IEA negates the need for an IQM and thus removes the problem of needing to find a satisfactory IQM.

The results of Chapters 3 and 6 in which it was observed that participants showed preferences for photographs which had been processed using image processes developed on the photographs upon which they had been developed demonstrates that there is a need for image content to be taken into account when applying image enhancement processes.

The large amount of noise observed in the data over all of the experiments due to the variability in the abilities and the temperaments of the participants calls into question the necessity of using a virtual user to optimise parameters in an IEA as was attempted in Chapter 4 and as was previously done in [34, 90]. Differences found by experiments in which comparisons were performed exclusively by a virtual user such as [125] and [55] should not be regarded as evidence that one IEA is better than another, only as an indication that further experimentation using human participants may find such differences.

The significant difference between the time taken to achieve a colour match using the hyperplane-IES and dummy-IES and the SMM-IES in Chapter 4 combined with the lack of such differences between the IGAs of Chapter 5 and the operators in Chapter 7 leads to the conclusion that how users evaluate members of the population, as investigated in [145, 34], has a greater effect on the performance of an IEA than the choice of parameters, operators, or the underlying EA.

In Chapter 6 it was found that the IEA approach produced better enhanced images than the slider based approach on one of the two images enhanced. Further investigation is required before it can be said that using an IEA will generally result in better enhanced images than a direct interface. An experiment is proposed in Section 8.3 the aim of which is to determine if using an IEA will generally result in better enhanced images. The results of Chapter 6 do tentatively support the findings in [130] and [77] that an IEA will provide better image enhancement results than direct manipulation of the input variables to the image enhancement processes.

It was found in Chapter 6 that the compound process produced better enhanced images than the piecewise intensity transfer function process on one of the two images enhanced. Further investigation is required before it can be said that using the compound process will generally result in better enhanced images than the intensity transfer process. An experiment is proposed in Section 8.3 the aim of which is to determine if the compound process will generally result in better enhanced images. The results of Chapter 6 do suggest that the compound process is an improvement over intensity transfer function processes like those used in [130] and [106].

In Chapter 6 it was found that the IEA attained a satisfactory image quicker than the direct interface on one of the two images enhanced. Further investigation is required before it can be said that using an IEA is generally quicker than a direct interface. The experiment is proposed in Section 8.3 to determine if an IEA will generally result in better enhanced images can also be used to determine if an IEA is generally quicker. The results of Chapter 6 do tentatively support the findings in [130] and [77] that an IEA will attain a satisfactory image quicker than direct manipulation of the input variables to the image enhancement processes.

In the work presented in Chapter 5 no statistically significant differences between the performances of the hyperplane-IGA based on the SMM-IEA presented is [41] and the simple IGA based on the IGA developed by Frowd for use in EvoFIT [34] were detected. Similarly, no significant differences between the performances of the recombination and mutation operators compared in Chapter 7 were detected. These results indicate that the IEA approach is robust to the choice of mutation and recombination operators and to a certain extent the choice of algorithm.

In Chapter 7 no differences were found between the face-spaces and this was attributed, in part, to the difficult nature of the task. In Chapter 5 no significant differences between the colour spaces were detected for the without target task. From these results it is concluded that IEAs are, to some extent, insensitive to the search space used. This conclusion fails to support the assertion made in [121] that a psychologically based search space will produce better results. The difference between the colour spaces with regards to the number of generations required to attain the target colour in the with target task of Chapter 5 suggests further testing to see if this effect can be observed for other tasks. This idea is expanded upon in Chapter 8.3

8.3 Future work

The work presented in this thesis suggests a number of avenues for future work. This work is divided here into two categories. The first is a list of specific experiments designed to confirm the less robust conclusions and observations drawn from the work in this thesis, the second is more general and identifies areas that may prove fruitful within the context of current work involving IEAs.

8.3.1 Specific experiments arising from the thesis

When working with human participants the amount of data that can be collected is severely limited. This is particularly apparent in Chapter 6 in which data from the processing of only two images were gathered. It is difficult to justify stating that an effect, namely that the IEA approach and the compound image process are better for processing images than the direct interface and the piecewise intensity transfer function process. To establish if these effects are observed in general two experiments are proposed. In the first, the compound and intensity transfer processes are compared using an IEA to enhance four or five images (plus one practice image). In the second, an IEA and a direct interface are compared in the same manner.

In the discussion of Chapter 4 it was suggested that the self adaptive step size aspect of an IEA was detrimental to the search and as a consequence manually adjustable mutation was used for the remainder of the experiments. A comparison between the self adaptive step size method, manual adjustment (with automatic decrementation), and a combination of the two in which the step size is self adaptive but users can adjust it if the need arises, would establish whether the suggestion was correct.

When building the search space for Chapter 6 some of the input values to the image processes in the compound process were logarithmically scaled to form a search space that better represented the psychological search space. To establish whether this was useful or if an IEA is generally robust enough for this to be unnecessary an experiment in which an IEA is used to enhance images in two search spaces, one with scaled inputs and one without, performed over four or five images is suggested.

8.3.2 Avenues for research in the wider field

In Chapter 4 and Chapter 5 a colour matching task very similar to those of [10] and [16] was used to compare IEAs and search spaces. The colour matching task is cognitively simple, easy to implement, and quick to perform but has too few dimensions to provide an accurate representation of tasks for which an IEA may be suitable. The work of Gong [43, 42, 46, 45, 44] in IEAs uses a fashion design task to compare algorithmic design options. This task is not simple to implement and the shape aspect of the clothing is easily ‘separable’, that is, it would almost certainly be easier to select the desired garments form a panel as opposed to being selected using an IEA. A good test task would have a sufficiently large number of dimensions to warrant the use of an IEA, be easy to implement, be easy to understand, and not be separable. Unfortunately, no such task appears to have been developed so far.

The lack of observed differences between the operators in Chapter 7, lack of differences between algorithms in Chapter 5 and the observation that rejecting individuals in the experiment of Chapter 5 significantly adds to the time taken without improving the performance of the search suggests that the development of interfaces for IEAs is a more fruitful endeavour than the development of underlying algorithms. A more robust experiment based on that of Yoon et al. [145] would provide a more informative comparison between rating methods. Takenouchi et al. [126] compared an IDE with a pairwise comparison interface to a IGA using a full scale rating interface and found that the IDE provided a superior performance. A two-way experiment with underlying algorithm as one factor and interface as the second would reveal which factor best accounts for the superior performance of the IDE pairwise comparison algorithm.

The main goal of most work in IEAs is the reduction of fatigue. The reasons for wanting to reduce fatigue are to make completing tasks using an IEA quicker and, for more difficult tasks such as the creation of facial composites, to make it more likely that a satisfactory result is obtained. A recent approach to reducing fatigue that shows promise is the use of surrogate fitness functions [46, 45, 44]. Surrogate fitness functions are developed using fitness data collected from the user. The surrogate fitness functions are used to make evaluations on the behalf of the user. The fitness evaluations collected, however, tend to be full scale rating on a continuous scale. Providing evaluations using a full scale rating induces fatigue to a greater extent than simpler methods. Comparisons between the performances of the surrogate fitness function approach and the simpler evaluation methods favoured in this thesis would form an interesting comparison between a simple interface and a complex underlying algorithm. Developing the surrogate fitness function approach to work with limited fitness evaluation data may provide better results that either approach could achieve individually.

One of the main motivations for this work was to enable people who know little about image enhancement to be able to use it to enhance images. It is unlikely that people would want to have to upload their photographs to a personal computer in order to enhance them. Work has been done to apply IEAs to image processing on smartphones [61, 70]. Extending the work presented in this thesis along the avenues of research suggested in this section may lead to the use of IEAs becoming a viable alternative to the direct manipulation approaches currently used on mobile devices.

Bibliography

[1] IEC 61966-2-1: Multimedia spystems and equipment — colour measurement and manaement, part 2-1: Colour management — default rgb colour space — srgb, 1999.

[2] ISO 15076-1:2005, image technology colour management — architecture, profile format, and data structure: Part 1, 2005.

[3] Adobe. Adobe photoshop. `http://www.adobe.com/products/photoshop/family/?promoid=BPDEK`.

[4] Akbal, T., Demir, G. N., Kanlikiliçer, A. E., Kus, M. C., and Ulu, F. H. Interactive nature-inspired heuristics for automatic facial composite generation. In *Genetic and Evolutionary Computaton Conference Undergraduate Student Workshop* (2006).

[5] Astola, J., Haavisto, P., and Neuvo, Y. Vector median filters. *IEEE Proceedings 78*, 4 (April 1990), 678–689.

[6] Avcibaş, I., Sankur, B., and Sayood, K. Statisical evaluation of image quality measures. *Journal of Electronic Imaging 11* (2002), 206–223.

[7] Beyer, H.-G. *The theory of evolution strategies.* Springer, 2001.

[8] Brace, N., Kemp, R., and Snelgar, R. *SPSS for Psychologists*, third ed. Palgrave Macmillan, Basingstoke, 2006.

[9] Brace, N., Pike, G., and Kemp, R. Investigating E-FIT using famous faces. In *Traditional Questions and New Ideas* (2000), A. Czerederecka, T. Jaskiewicz-Obydzinska, and J. Wójcikiewicz, Eds., Kraków: Institute of Forensic Research Publishers.

[10] Breukelaar, R., Emmerich, M., and Bäck, T. On interactive evolution strategies. In *Applications of Evolutionary Computing* (2006), F. R. et al, Ed., vol. 3907, Springer-Verlag, pp. 530–541.

[11] Bruce, V., Ness, H., Hancock, P. J. B., Newman, C., and Rarity, J. Four heads are better than one: combining face composites yields improvements in face likeness. *Journal of Applied Psychology 87*, 5 (2002), 894.

[12] Brunelli, R., and Mich, O. SpotIt! an interactive identikit system. *Graphical Models and Image Processing 58*, 5 (1996), 399–404.

[13] Buades, A., Coll, B., and Morel, J.-M. A non-local algorithm for image denoising. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2005), vol. 2, IEEE, pp. 60–65.

[14] Burger, W., and Burge, M. J. *Digital Image Processing: An Algorithmic Introduction using Java.* Springer, 2008.

[15] Chatterjee, P., and Milanfar, P. Is denoising dead? *IEEE Transactions on Image Processing 19*, 4 (2010), 895–911.

[16] Cheng, C. D., and Kosorukoff, A. Interactive one-max problem allows to compare the performance of interactive and human-based genetic algorithms. In *Proceedings of the 6th annual conference on Genetic and evolutionary computation* (2004), Springer, pp. 983–993.

[17] Cho, S.-B. Towards creative evolutionary systems with interactive genetic algorithm. *Applied Intelligence 16*, 2 (2002), 129–138.

[18] Christie, D., and Ellis, H. Photofit constructions versus verbal descriptions of faces. *Journal of Applied Psychology 66*, 3 (1981), 358–363.

[19] Conover, W. J. *Practical nonparametric statistics*, second ed. Wiley, New York, 1980.

[20] Cootes, T. F., Edwards, G. J., and Taylor, C. J. Active appearance models. In *Proceedings of the European Conference on Computer Vision* (1998), H. Burkhardt and B. Neumann, Eds., vol. 2, Springer, pp. 484–498.

[21] Craw, I., and Cameron, P. Parameterising images for recognition and reconstruction. In *British Machine Vision Conference* (London, 1991), P. Mowforth, Ed., Springer Verlag, pp. 367–370.

[22] Cree, M. Observations on adaptive vector filters for noise reduction in color images. *IEEE Signal Processing Letters 11* (2004), 140–143.

[23] Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on Image Processing 16*, 8 (2007), 2080–2095.

[24] Davies, G., and Christie, D. Face recall: An examination of some factors limiting composite production accuracy. *Journal of Applied Psychology 67*, 1 (1982), 103.

[25] De Berg, M., Van Kreveld, M., Overmars, M., and Cheong, O. *Computational Geometry: Algorithms and Applications.* Springer, 2008.

[26] de Castella, T. Five ways the digital camera changed us. http://www.bbc.co.uk/news/magazine-16483509, January 2012. Accessed 07/01/2015.

[27] Dunleavy, P. Identikit, 1959,1975.

[28] Eberhart, R., and Kennedy, J. A new optimizer using particle swarm theory. In *Proceedings of the Sixth International Symposium on Micro Machine and Human Science* (1995), IEEE, pp. 39–43.

[29] Eiben, A. E., and Smith, J. E. *Introduction to Evolutionary Computing.* Springer, 2003.

[30] Eisenmann, J., Schroeder, B., Lewis, M., and Parent, R. Creating choreography with interactive evolutionary algorithms. In *Applications of Evolutionary Computation.* Springer, 2011, pp. 293–302.

[31] Fogel, D. *Artificial intelligence through simulated evolution.* Wiley-IEEE Press, 2009.

[32] Fogel, L. J., Owens, A. J., and Walsh, M. J. Artificial intelligence through a simulation of evolution. In *Biophysics and Cybernetic Systems: Proceedings of the 2nd Cybernetic Sciences Symposium* (Washington, D.C., 1965), M. Maxtield, A. Callahan, and L. J. Fogel, Eds., Spartan Books, pp. 131–155.

[33] Foi, A., and Boracchi, G. Foveated self-similarity in nonlocal image filtering. In *IS&T/SPIE Electronic Imaging* (2012), International Society for Optics and Photonics, pp. 829110–829110.

[34] Frowd, C. D. *EvoFIT: A Holistic, Evolutionary Facial Imaging System.* PhD thesis, Department of Psychology, University of Stirling, 2001.

[35] Frowd, C. D., Bruce, V., Plenderleith, Y., and Hancock, P. J. B. Improving target identification using pairs of composite faces constructed by the same person. In *IEE Conference on Crime and Security* (London, 2006), IET, pp. 386–395.

[36] Frowd, C. D., and Hancock, P. J. B. EvoFIT. http://www.evofit.co.uk.

[37] Frowd, C. D., Hancock, P. J. B., and Carson, D. Evofit: A holistic, evolutionary facial imaging technique for creating composites. *ACM Transactions in Applied Perception 1*, 1 (2004), 19–39.

[38] Frowd, C. D., Skelton, F., Hepton, G., Holden, L., Minahil, S., Pitchford, M., McIntyre, A., Brown, C., and Hancock, P. J. B. Whole-face procedures for recovering facial images from memory. *Science & Justice 53*, 2 (2013), 89–97.

[39] George, B., Gibson, S. J., Maylin, M. I., and Solomon, C. J. EFIT-V — interactive evolutionary strategy for the construction of photo-realistic facial composites. In *Proceedings of the 10th annual conference on Genetic and evolutionary computation* (New York, NY, USA, 2008), ACM, pp. 1485–1490.

[40] Ghosh, P., and Mitchell, M. Segmentation of medical images using a genetic algorithm. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation* (2006), ACM, pp. 1171–1178.

[41] Gibson, S. J., Solomon, C. J., and Pallares Bejarano, A. Synthesis of photographic quality facial composites using evolutionary algorithms. In *British Machine Vision Conference 2003* (2003), R. Harvey and J. A. Bangham, Eds., vol. 1, pp. 221–230.

[42] Gong, D., and Guo, G. S. Interactive genetic algorithms with interval fitness of evolutionary individuals. *Dynamics of Continuous, Discrete and Impulsive Systems, Series B: Complex Systems and Applications-modeling, Control and Simulations 14*, s2 (2007), 446–450.

[43] Gong, D., Hao, G.-S., Zhou, Y., and Sun, X.-Y. Interactive genetic algorithms with multi-population adaptive hierarchy and their application in fashion design. *Applied Mathematics and Computation 185*, 2 (2007), 1098–1108.

[44] Gong, D., Yang, L., Sun, X., and Li, M. Applying knowledge of users with similar preference to construct surrogate models of IGAs. In *IEEE Congress on Evolutionary Computation* (2012), IEEE, pp. 1–8.

[45] Gong, D., Yao, X., and Yuan, J. Interactive genetic algorithms with individual fitness not assigned by human. *Journal of Universal Computer Science 15*, 13 (2009), 2446–2462.

[46] Gong, D., Yuan, J., and Ma, X. Interactive genetic algorithms with large population size. In *IEEE Congress on Evolutionary Computation* (2008), IEEE, pp. 1678–1685.

[47] Gonzalez, R. C., and Woods, R. E. *Digital image processing*, second ed. Prentice Hall, 2002.

[48] Gorai, A., and Ghosh, A. Hue-preserving color image enhancement using particle swarm optimization. In *Recent Advances in Intelligent Computational Systems* (2011), IEEE, pp. 563–568.

[49] Hancock, P. J. B. Evolving faces from principal components. *Behavior Research Methods, Instruments, & Computers 32*, 2 (2000), 327–333.

[50] Harvey, N. R., Brumby, S. P., Perkins, S., Szymanski, J. J., Theiler, J., Bloch, J. J., Porter, R. B., Galassi, M., and Young, A. C. Image feature extraction: GENIE vs conventional supervised classification techniques. *IEEE Transactions on Geoscience and Remote Sensing 40*, 2 (2002), 393–404.

[51] Hashemi, S., Kiani, S., Noroozi, N., and Moghaddam, M. E. An image contrast enhancement method based on genetic algorithm. *Pattern Recognition Letters 31*, 13 (2010), 1816–1824.

[52] Haupt, R. L. Optimum population size and mutation rate for a simple real genetic algorithm that optimizes array factors. In *IEEE Antennas and Propagation Society International Symposium* (2000), vol. 2, IEEE, pp. 1034–1037.

[53] He, Z., Yen, G. G., and Zhang, J. Fuzzy-based pareto optimality for many-objective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation 18* (2014), 269–285.

[54] Holland, J. H. Genetic algorithms and the optimal allocation of trials. *SIAM Journal on Computing 2*, 2 (1973), 88–105.

[55] Hornby, G. S., and Bongard, J. C. Accelerating human-computer collaborative search through learning comparative and predictive user models. In *Proceedings of the fourteenth international Genetic and evolutionary computation conference* (2012), ACM, pp. 225–232.

[56] Hoseini, P., and Shayesteh, M. G. Hybrid ant colony optimization, genetic algorithm, and simulated annealing for image contrast enhancement. In *IEEE Congress on Evolutionary Computation* (2010), IEEE, pp. 2840–2845.

[57] Howell, D. C. *Statistical Methods for Psychology.* Wadsworth, Belmont CA, 2010.

[58] Hsieh, M.-H., Cheng, F.-C., Shie, M.-C., and Ruan, S.-J. Fast and efficient median filter for removing 1–99% levels of salt-and-pepper noise in images. *Engineering Applications of Artificial Intelligence 26* (2012), 1333–1338.

[59] Hsu, F.-C., and Huang, P. Providing an appropriate search space to solve the fatigue problem in interactive evolutionary computation. *New Generation Computing 23*, 2 (2005), 115–127.

[60] Jakša, R., Nakano, S., and Takagi, H. Image filter design with interactive evolutionary computation. In *IEEE International Conference on Computational Cybernetics* (Siofok, Hungary, August 2003), pp. 29–31. ISBN 963 7154 175.

[61] Jung, T.-M., Lee, Y.-S., and Cho, S.-B. Mobile interface for adaptive image refinement using interactive evolutionary computing. In *Proceedings of the IEEE World Congress on Computational Intelligence* (2010).

[62] Kamalian, R., Yeh, E., Zhang, Y., Agogino, A. M., and Takagi, H. Reducing human fatigue in interactive evolutionary computation through fuzzy systems and machine learning systems. In *IEEE International Conference on Fuzzy Systems* (2006), IEEE, pp. 678–684.

[63] Kelly, J., Papalambros, P. Y., and Seifert, C. M. Interactive genetic algorithms for use as creativity enhancement tools. In *AAAI Spring Symposium: Creative Intelligent Systems* (2008), pp. 34–39.

[64] Kirby, M., and Sirovich, L. Application of the karhunen-loève procedure for the characterization of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence 12*, 1 (1990), 103–108.

[65] Koga, S., Inoue, T., and Fukumoto, M. A proposal for intervention by user in interactive genetic algorithm for creation of music melody. In *International Conference on Biometrics and Kansei Engineering* (2013), IEEE, pp. 129–132.

[66] Kowaliw, T., Dorin, A., and McCormack, J. Promoting creative design in interactive evolutionary computation. *IEEE transactions on evolutionary computation 16*, 4 (2012), 523.

[67] Koza, J. R. Hierarchical genetic algorithms operating on populations of computer programs. In *International Joint Conference on Artificial Intelligence* (1989), IJCAI, pp. 768–774.

[68] Kumar, N. N., and Ramakrishna, S. An efficient approach of removing the high density salt and pepper noise using stationary wavelet transform. *Global Journal of Computer Science and Technology 12*, 5 (2012), 43–47.

[69] Kurt, B., Etaner-Uyar, A. S., Akbal, T., Demir, N., Kanlikilicer, A. E., Kus, M. C., and Ulu, F. H. *Lecture Notes in Computer Science*, vol. 4105.

Springer-Verlag, 2006, ch. Active appearance model-based facial composite generation with interactive nature inspired heuristics, pp. 183–190.

[70] Lee, M.-C., and Cho, S.-B. Interactive differential evolution for image enhancement application in smart phone. In *IEEE Congress on Evolutionary Computation* (2012), IEEE, pp. 2411–2416.

[71] Levitan, B., and Kauffman, S. Adaptive walks with noisy fitness measurements. *Molecular Diversity 1*, 1 (1995), 53–68.

[72] Lewis, M., and Ruston, K. Aesthetic geometry evolution in a generic interactive evolutionary design framework. *New Generation Computing 23*, 2 (2005), 171–179.

[73] Li, X., Tang, K., Omidvar, M. N., Yang, Z., Qin, K., and China, H. Benchmark functions for the CEC 2013 special session and competition on large-scale global optimization. *Gene 7* (2013), 33.

[74] Lukac, R., Plataniotis, K. N., and Venetsanopoulos, A. N. Color image denoising using evolutionary computation. *International Journal of Imaging Systems and Technology 15*, 5 (2005), 236–251.

[75] Lukac, R., Smolka, B., Martin, K., Plataniotis, K., and Venetsanopoulos, A. Vector filtering for color imaging. *IEEE Signal Processing Magazine 1* (2005), 74–86.

[76] Luo, W. An efficient detail-preserving approach for removing impulse noise in images. *IEEE Signal Processing Letters 13* (2006), 413–416.

[77] Ma, J., and Takagi, H. Design of composite image filters using interactive genetic programming. In *Third International Conference on Innovations in Bio-Inspired Computing and Applications* (2012), IEEE, pp. 274–279.

[78] MacQueen, J. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability* (1967), vol. 1, California, USA, pp. 281–297.

[79] Mádar, J., Abonyi, J., and Szeifert, F. Interactive particle swarm optimization. In *Proceedings of the 5th International Conference on Intelligent Systems Design and Applications* (2005), IEEE, pp. 314–319.

[80] McQuiston-Surrett, D., Topp, L. D., and Malpass, R. S. Use of facial composite systems in us law enforcement agencies. *Psychology, Crime & Law 12*, 5 (2006), 505–517.

[81] Mitchell, M. *An Introduction to Genetic Algorithms.* MIT Press, 1996.

[82] Mitchell, T. M. *Machine Learning.* McGraw-Hill, London, 1997.

[83] Munteanu, C., Morales, F. C., and Ruiz-Alzola, J. Speckle reduction through interactive evolution of a general order statistics filter for clinical ultrasound imaging. *IEEE Transactions on Biomedical Engineering 55*, 1 (2008), 365–369.

[84] Munteanu, C., and Rosa, A. Towards automatic image enhancement using genetic algorithms. In *Proceedings of the Congress on Evolutionary Computation* (2000), vol. 2, IEEE, pp. 1535–1542.

[85] Munteanu, C., and Rosa, A. Evolutionary image enhancement with user behaviour modeling. In *Proceedings of the ACM symposium on Applied computing* (2001), ACM, pp. 316–320.

[86] Munteanu, C., and Rosa, A. Gray-scale image enhancement as an automatic process driven by evolution. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 34*, 2 (2004), 1292–1298.

[87] Oinuma, J., Arakawa, K., and Harashima, H. Evaluation of genetic algorithm for interactive evolutionary face image beautifying system. In *6th International Symposium on Communications, Control and Signal Processing* (2014), IEEE, pp. 594–597.

[88] Otobe, K., Tanaka, K., and Hitafuji, M. Image processing and interactive selection with Java based on genetic algorithms. In *Proceedings of the 3rd IFAC/CIGR Workshop on Artificial Intelligence in Agriculture* (1998), pp. 83–88.

[89] Pal, S. K., Bhandari, D., and Kundu, M. K. Genetic algorithms for optimal image enhancement. *Pattern Recognition Letters 15*, 3 (1994), 261–271.

[90] Pallares-Bejarano, A. *Evolutionary Algorithms for Facial Composite Synthesis.* PhD thesis, University of Kent, 2006.

[91] Pallares-Bejarano, A., Gibson, S. J., Maylin, M. I. S., and Solomon, C. J. Eigenfit - an evolutionary approach to facial composite construction. In *18th International Symposium on The Forensic Sciences* (2006). Proceedings contains abstract only, keynote talk.

[92] Pallez, D., Collard, P., Baccino, T., and Dumercy, L. Eye-tracking evolutionary algorithm to minimize user fatigue in IEC applied to interactive one-max problem. In *Proceedings of the 2007 GECCO conference companion on Genetic and evolutionary computation* (2007), ACM, pp. 2883–2886.

[93] Penry, J. Photo-fit. *Forensic Photography 3*, 7 (1974), 4–10.

[94] Pitas, I., and Venetsanopoulos, A. N. Order statistics in digital image processing. *Proceedings of IEEE 80*, 12 (December 1992), 1892–1921.

[95] Pizer, S. M., Amburn, E. P., Austin, J. D., Cromartie, R., Geselowitz, A., Greer, T., ter Haar Romeny, B., Zimmerman, J. B., and Zuiderveld, K. Adaptive histogram equalization and its variations. *Computer vision, graphics, and image processing 39*, 3 (1987), 355–368.

[96] Plataniotis, K., Sri, V., Androutsos, D., and Venetsanopoulos, A. An adaptive nearest neighbor multichannel filter. *IEEE Transactions on Circuits and Systems for Video Technology 6(6)* (1996), 699–703.

[97] Plataniotis, K. N., Androutsos, D., Sri, V., and Venetsanopoulos, A. N. Nearest-neighbour multichannel filter. *Electronics Letters 31(22)* (1995), 1910–1911.

[98] Plataniotis, K. N., Androutsos, D., and Venetsanopoulos, A. N. Adaptive fuzzy systems for multichannel signal processing. *Proceedings of the IEEE 87* (1999), 1601–1622.

[99] Plataniotis, K. N., Androutsos, D., Vinayagamoorthy, S., and Venetsanopoulos, A. N. Color image processing using adaptive multichannel filters. *IEEE Transactions on image processing 6* (1997), 933 – 949.

[100] Poli, R., and Cagnoni, S. Genetic programming with user-driven selection: Experiments on the evolution of algorithms for image enhancement. In *Proceedings of the Second International Conference on Genetic Programming* (1997), Morgan Kaufmann, pp. 269–277.

[101] Pratt, W. K. *Digital image processing.* John Wiley & Sons, New York, 1978.

[102] Quiroz, J. C., Louis, S. J., and Dascalu, S. M. Interactive evolution of XUL user interfaces. In *Proceedings of the 9th annual conference on Genetic and evolutionary computation* (2007).

[103] R., A. Facebook has 220 billion of your photos to put on ice. `http://gigaom.com/2012/10/17/facebook-has-220-billion-of-your-photos-to-put-on-ice/`, October 2012. Accessed 26/03/2014.

[104] Rechenberg, I. Cybernetic solution path of an experimental problem,(royal aircraft establishment translation no. 1122, bf toms, trans.). *Farnsborough Hants: Ministery of Aviation, Royal Aircraft Establishment 1122* (1965).

[105] Rohlf, F. J., and Slice, D. Extensions of the procrustes method for the optimal superimposition of landmarks. *Systematic Biology 39*, 1 (1990), 40–59.

[106] Saitoh, F. Image contrast enhancement using genetic algorithm. In *IEEE International Conference on Systems, Man, and Cybernetics* (1999), vol. 4, IEEE, pp. 899–904.

[107] Shao, L., Yan, R., Li, X., and Liu, Y. From heuristic optimization to dictionary learning: a review and comprehensive comparison of image denoising algorithms. *IEEE Transactions on Cybernetics 44*, 7 (2014), 1001–1013.

[108] Sheikh, H. R., Sabir, M. F., and Bovik, A. C. A statistical evaluation of recent full reference image quality assessment. *IEEE Transactions on Image Processing 15* (2006), 3440–3451.

[109] Sheikh, H. R., Wang, Z., Cormack, L., and Bovik, A. C. LIVE image quality assessment database release 2. `http://live.ece.utexas.edu/research/quality`.

[110] Shyu, M.-S., and J.-J., L. A genetic algorithm approach to color image enhancement. *Pattern Recognition 31* (1998), 871–880.

[111] Singh, T., Kharma, N., Daoud, M., and Ward, R. Genetic programming based image segmentation with applications to biomedical object detection. In *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation* (2009), pp. 1123–1130.

[112] Sirovich, L., and Kirby, M. Low-dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America A 4*, 3 (1987), 519–524.

[113] Solomon, C. J., and Breckon, T. *Fundamentals of Digital Image Processing: A Practical Approach with Examples in Matlab*. John Wiley, 2011.

[114] Solomon, C. J., Gibson, S. J., and Maylin, M. I. S. *A New Computational Methodology for the Construction of Forensic, Facial Composites*, vol. 5718/2009. Springer-Verlag LNCS, August 2009, pp. 67–77.

[115] Solomon, C. J., Gibson, S. J., and Mist, J. J. Interactive evolutionary generation of facial composites for locating suspects in criminal investigations. *Applied Soft Computing 13.7* (2013), 3298–3306.

[116] Spiegel, M. R., and Stephens, L. J. *Statistics*, fourth ed. McGraw-Hill, London, 2011.

[117] Srinivas, M., and Patnaik, L. M. Genetic algorithms: A survey. *Computer 27*, 6 (1994), 17–26.

[118] Steinwart, I. Christmann, A. *Support Vector Machines.* Springer, 2006.

[119] Storn, R., and Price, K. Differential evolution — a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization 11*, 4 (1997), 341–359.

[120] Stroock, D. *Probability Theory.* Cambridge University Press, 2011.

[121] Sugimoto, F., and Honda, N. A human interface to search and draw facial images in mind by using psychometrical space model of faces. In *IEEE International Fuzzy Systems Conference Proceedings* (1999), vol. 3, IEEE, pp. 1585–1590.

[122] Sun, X., Gong, D., Jin, Y., and Chen, S. A new surrogate-assisted interactive genetic algorithm with weighted semisupervised learning. *IEEE Transactions on Cybernetics 43*, 2 (2013), 685–698.

[123] Sun, X., Gong, D., and Zhang, W. Interactive genetic algorithms with large population and semi-supervised learning. *Applied Soft Computing 12*, 9 (2012), 3004–3013.

[124] Takagi, H. Interactive evolutionary computation: Fusion of the capabilities for EC optimization and human evaluation. *Proceedings of the IEEE 89*, 9 (2001), 1275–1296.

[125] Takagi, H., and Pallez, D. Paired comparison-based interactive differential evolution. In *World Congress on Nature & Biologically Inspired Computing* (2009), IEEE, pp. 475–480.

[126] Takenouchi, H., Tokumaru, M., and Muranaka, N. Performance evaluation of interactive evolutionary computation with tournament-style evaluation. In *IEEE Congress on Evolutionary Computation* (2012), IEEE, pp. 1–8.

[127] Tanaka, J. W., and Farah, M. J. Parts and wholes in face recognition. *Quarterly Journal of Experimental Psychology 46A* (1993), 225–245.

[128] Tanaka, M., Sasaki, Y., Miki, M., and Hiroyasu, T. Crossover method for interactive genetic algorithms to estimate multimodal preferences. *Applied Computational Intelligence and Soft Computing 2013* (2013).

[129] Taylor, K. T. *Forensic Art & Illustration.* CRC Press, 2000.

[130] Tokuda, Y., Hashino, H., Ohashi, G., Tsukada, M., Kobayashi, R., and Shimodaira, Y. Image quality enhancement support system by gamma correction using interactive evolutionary computation. In *IEEE International Conference on Systems, Man and Cybernetics* (2007), IEEE, pp. 2906–2910.

[131] Tomasi, C., and Manduchi, R. Bilateral filtering for gray and color images. In *Sixth International Conference on Computer Vision* (1998), IEEE, pp. 839–846.

[132] Trahanias, P. E., and Venetsanopoulos, A. N. Vector directional filters — a new class of multichannel image processing filters. *IEEE Transactions on Image Processing 2* (1993), 528–534.

[133] Ueda, Y., Kuramoto, Y., Kubota, R., Suetake, N., and Uchino, E. An interactive genetic algorithm-based image sharpening system considering user's liking. In *IEEE Symposium on Computational Intelligence for Engineering Solutions* (2013), IEEE, pp. 91–96.

[134] Vafaie, H., and Imam, I. F. Feature selection methods: genetic algorithms vs. greedy-like search. In *Proceedings of International Conference on Fuzzy and Intelligent Control Systems* (1994).

[135] Valentine, T. A unified account of the effects of distinctiveness, inversion, and race in face recognition. *The Quarterly Journal of Experimental Psychology 43A* (1991), 161–204.

[136] Valentine, T., Davis, J. P., Thorner, K., Solomon, C., and Gibson, S. Evolving and combining facial composites: Between-witness and within-witness morphs compared. *Journal of Experimental Psychology: Applied 16*, 1 (2010), 72.

[137] Verma, O. P., Kumar, P., Hanmandlu, M., and Chhabra, S. High dynamic range optimal fuzzy color image enhancement using artificial ant colony system. *Applied Soft Computing 12*, 1 (2012), 394–404.

[138] Vijaykumar, V. R., Vanathi, P. T., and Kanagasabapathy, P. Fast and efficient algorithm to remove gaussian noise in digital images. *IAENG International Journal of Computer Science 37*, 1 (2010), 78–84.

[139] Visionmetric. EFIT-V. http://www.visionmetric.com.

[140] Walsh, P., and Gade, P. Terrain generation using an interactive genetic algorithm. In *IEEE Congress on Evolutionary Computation* (2010), IEEE, pp. 1–7.

[141] Wang, Z., and Bovik, A. *Modern Image Quality Assessment.* Morgan and Claypool, 2006.

[142] Whigham, P. A., Aldridge, C., and de Lange, M. Constrained evolutionary art: Interactive flag design. In *IEEE Congress on Evolutionary Computation* (2009), IEEE, pp. 2194–2200.

[143] Wobbrock, J. O., Findlater, L., Gergle, D., and Higgins, J. J. The aligned rank transform for nonparametric factorial analyses using only anova procedures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2011), ACM, pp. 143–146.

[144] Yang, X.-S. Firefly algorithms for multimodal optimization. In *Stochastic algorithms: foundations and applications*. Springer, 2009, pp. 169–178.

[145] Yoon, D.-M., and Kim, K.-J. Comparison of scoring methods for interactive evolutionary computation based image retouching system. In *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference companion* (2012), ACM, pp. 617–618.