Experiments in Objet Trouvé Browsing

Simon Colton*, Jeremy Gow*, Pedro Torres*, and Paul Cairns†

* Department of Computing, Imperial College, London, UK.
† Department of Computer Science, University of York, UK.

Abstract. We report on two experiments to study the use of a graphic design tool for generating and selecting image filters, in which the aesthetic preferences that the user expresses whilst browsing filtered images drives the filter generation process. In the first experiment, we found evidence for the idea that intelligent employment of the user's preferences when generating filters can improve the overall quality of the designs produced, as assessed by the users themselves. The results also suggest some user behaviours related to the fidelity of the image filters, i.e., how much they alter the image they are applied to. A second experiment tested whether evolutionary techniques which manage fidelity would be preferred by users. Our results did not support this hypothesis, which opens up interesting questions about how user preferences can be intelligently employed in browsing-based design tools.

1 Introduction

The Objet Trouvé (Found Object) movement in modern art gained notoriety by incorporating everyday objects – often literally found discarded in the streets – into visual art and sculpture pieces. This is a two stage process, whereby the original object is first found, and then manipulated into a piece of art. This process is analogous with certain practices in computer-supported graphic design. In particular, both amateur and expert designers will often find themselves browsing through libraries of image filters; or brush shapes; or fonts; or colour palettes; or design templates; etc. Once they have found some possibilities, these are pursued further and manipulated into a final form. This analogy with Objet Trouvé methods is most pronounced in the field of evolutionary art, where artistic images (or more precisely the image generating processes) are evolved, e.g., [5]. Here, the software initially leads the user, through its choices of processes to employ and the way in which it combines and/or mutates those processes as the session progresses. However, as the user begins to exert their aesthetic preferences through their choices, the software should enable them to quickly turn their found processes into a final form. We investigate here the behaviour of "amateur creators" [2] when using such design software. Our motivation is to ultimately build software which acts as a creative collaborator in design processes.

We present here the results from two experiments where participants were asked to undertake graphic design tasks using a simple design tool which allows users to browse and select image-filtered versions of a source image in an Objet Trouvé fashion. Various techniques may be used to supply new filters on demand. As described in section 2, our tree based image filtering method enables evolutionary, database lookup and image retrieval techniques to be used in providing

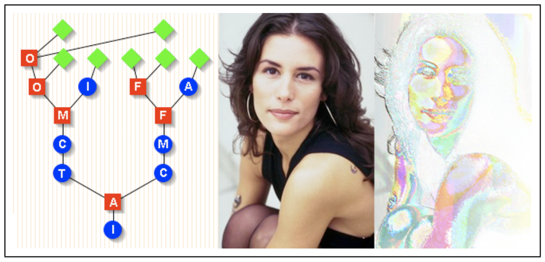

Fig. 1. Filter tree with transforms in blue circles: (A)dd colour, (C)onvolution, (I)nverse, (M)edian and (T)hreshold; and compositors in red squares: (A)nd, (F)ade, (M)in and (O)r. Image inputs are in green diamonds. An original image and filtered version are shown.

a user with new filters. We incorporated six such techniques into a very pared-down user interface which enables the user to undertake simple graphic design tasks. As described in section 3, the first experiment was designed to test the hypothesis that employing the user's current choices to intelligently determine what to show them next is more effective than a random selection method. The data revealed that users ascribe on average a higher score to designs produced by the intelligent methods than produced randomly. In addition, by studying user preferences towards the six generation methods, we hypothesised certain user behaviours, largely involving their preferences towards more conservative image filters, i.e., ones with high *fidelity* which don't radically change the original image. Studying these behaviours enabled us to design a second experiment involving evolutionary techniques where evolved filters are supplemented with other filters. As described in section 4, this experiment tested the hypothesis that choosing the supplementary filters to manage the overall average fidelity of the filters would be more effective than choosing them randomly. The data did not support this hypothesis, which opens up interesting questions about how to analyse the behaviour of a user to improve the quality of the content they are shown as their browsing session progresses. In section 5, we suggest more intelligent methods for future Objet Trouvé design approaches.

2 Image Filtering

An image filter such as blurring, sharpening, etc. manipulates the bitmap information of an original digital image into bitmap information for a filtered version of the image. We represent image filters as a tree of fundamental (unary) image transforms such as inverse, lookup, threshold, colour addition, median, etc., and (binary) image compositors such as add, and, divide, max, min, multiply, or, subtract, xor, etc. In the example tree of figure 1, the overall filter uses 7 transform steps and 6 compositor steps, and the original image is input to the tree 7 times. Using this representation, image filters can be generated randomly, as described in [3]. We used such random generation to produce a library of 1000 hand-chosen filters, compiled into 30 categories according to how the filtered images look, e.g., there are categories for filters which produce images that are blurred, grainy, monotone, etc. The time taken to apply a filter is roughly proportional to the size of the input image multiplied by the size of the tree. Over the entire library of filters, the average number of nodes in a tree is 13.62, and the average time – on a Mac OS X machine running at 2.6Ghz – to apply a filter to an image of 256 by 384 pixels is 410 milliseconds.

2.1 Filter Generation Methods

As described in the experiments below, we investigate how best to supply a user with a set of novel filters (N) given a set of filters (C) for which they have already expressed an interest. We have implemented the following methods.

• **Database methods.** These two methods use the 30 hand-constructed categories from our image filter library. The *Random From Category (RFC)* method supplies filters for N which are chosen randomly from the library. To do this, firstly a category is chosen at random, and then a filter is chosen at random from the category. Given that some of the categories have up to 100 filters in them, we found that choosing evenly between the categories gave the user more variety in the filters shown than simply choosing from the 1000 filters at random (which tends to bias towards filters in the most popular few categories – which can look fairly similar). Whenever a filter has been shown to the user, it is removed from a category, so it will not be shown again, and if a category has been exhausted, then a new one is chosen randomly. The *More From Categories (MFC)* method takes each filter in C with an equal probability, finds the library category from which it came and then chooses a filter from this category at random to add to N. As before, filters which have been shown to the user are removed from the category. Exhausted categories are re-populated, but when a filter is used from the category, the filter is mutated (see below), to avoid repetitions.

• **Image retrieval methods.** An alternative way to retrieve filters from the library is to search for filters which are closest to the user choices in terms of colour or texture. We call these the *Colour Search (CS)* and *Texture Search (TS)* retrieval methods. We modified standard image retrieval techniques to perform this search, using information derived offline about how the filters alter the colour histogram and texture of some standard images, as described in further detail in [11]. While the search techniques work with approximate information about filters (to perform efficiently), they are fairly accurate, i.e., in [11], we report a 93% probability of retrieving a given filter in a set of 10. Again, a record of the library filters shown to the user is kept to avoid repetitions.

• **Evolutionary methods.** Representing image filters as trees enables both the crossing over of branches into offspring from each of two parents, and the mutation of trees. To perform crossover, we start with two parent trees (called the *top* and *bottom* parent), and we choose a pair of nodes: N_t on the top parent and N_b on the bottom parent. These are chosen randomly so that the size of the tree above and including N_t added to the size of the tree gained from removing the tree above and including N_b from the bottom parent is between a predefined minimum and maximum (3 and 15 for the experiments here). After 50 failed attempts to choose such a pair of nodes, the two trees are deemed incompatible, and a new couple is chosen. If they are compatible, however, the tree above and including N_t is substituted for the tree above and including N_b to produce the offspring. An example crossover operation is shown in figure 2a. We have implemented various mutation techniques, which alter the filter by different amounts, with details given in [3]. For the experiments here, we employ a mutation method which randomly mutates a single transform/compositor node in the filter tree

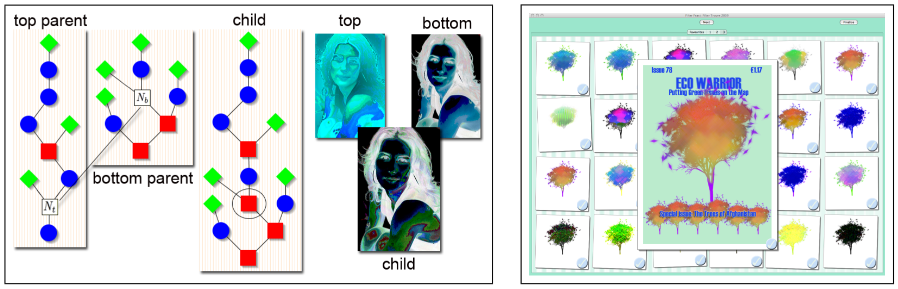

Fig. 2. a) Example crossover operation. b) Filter Trouvé design interface screenshot.

into a different transform/compositor node. This almost guarantees that a visible change will occur, and through experience we have found that it can produce a range of changes to the filter, from mild to fairly strong, depending on where the mutated node is in the tree. With the *Cross-Over (CO)* method, we choose pairs of filters randomly from the set of chosen filters C and perform one-point crossover to produce new filters for N. With the *Mutate (MT)* method, filters from C are chosen randomly and mutated as above to produce filters for N. Note that filters generated by the CO or MT methods are checked to determine if they produce single colour images or the same image as another child of the same parents. If so, the filter is rejected and another one is generated.

2.2 The Filter Trouvé User Interface

The user interface employed for the experiments described below is portrayed in figure 2b. In each session, the user works on a single design which involves filtering a single image which is incorporated (possibly multiple times) into a design, such as a magazine cover, etc. The user is shown the expression of 24 filters on the image in successive screens. They can click on a filtered image to see it in the design (in figure 2b, a filtered version of a tree image is shown in a magazine cover design). The user can also click a tick sign on each image to express their preference for it. When a user has chosen the images they like from a sheet, they click on the 'next' button which supplies them with another sheet of 24 images. At the end of a session, after clicking a 'finalise' button, users are shown all the designs that they ticked one by one in a random order. For each design, they are asked to judge it by choosing one of: (1) would definitely not use it (2) would probably not use it (3) not sure (4) would probably use it, and (5) would definitely use it. These choices essentially provide a score between 1 and 5 for each design. When each design has been given a score, the session ends. There has been considerable interest recently in creativity support tools like Filter Trouvé, which allow users to quickly generate, explore and compare multiple alternatives [9]. Our design embraces Shneiderman's principle of "low thresholds" for novices, but intentionally avoids the "high ceilings and wide walls" of advanced and comprehensive functionality. Filter Trouvé is designed for the *amateur creator* [2], who is not motivated to gain domain expertise, as opposed to the *novice*, who intends to become an expert. However, we see no reason why Objet Trouvé methods could not support other user groups.

3 Experiment 1

To recap, we are interested in comparing methods for presenting users with successive sets of image filters, so that they can drive a browsing session via their aesthetic choices. In this experiment, we investigate whether using techniques which choose new filters based on the user's choices perform better than techniques which choose them randomly. To do this, we compared the Random From Category (RFC) generation technique with a hybrid technique which we call *Taster*. This supplies the user with: 4 filters from MFC; 4 from CO; 4 from MT; 3 from CS; 3 from TS and 6 from RFC. Note that we included filters returned by RFC in the Taster method, as we found in initial tests that users appreciate the variety provided by RFC. Note also that providing only 3 images from the CS and TS methods was a mistake – while we had intended to provide 4 images for each of these methods, the mistake does not affect the conclusions we draw. The two hypotheses we proposed were:

• The Taster method will produce better images than the RFC method, as measured by the scores ascribed by participants to their designs.
• Taster will be quicker to use than RFC, as measured by the time to complete tasks and number of images viewed before deciding they have finished the task.

We asked 29 participants with varying levels of graphic design experience (no professionals) to undertake 4 design tasks using the Filter Trouvé interface. The design tasks were: a gallery installation, where a filtered image of a cityscape was included four times with wooden frames; a magazine cover, which involved a filtered image of a woman's face behind text; a Facebook profile, which involved a filtered version of man's face; and a book cover, where a filtered version of a (haunted) house appears on the front and back. We instructed participants to tick any filters they liked on a sheet and to stop when either around 10 minutes had passed, or they felt they had enough designs they were pleased with, or they felt the search was futile and wanted to stop. We balanced the two experimental conditions (i.e., the Taster method and the RFC method) in such a way that each participant had both conditions twice. This meant that there were either 14 or 15 participants in each pairing of design task and condition. The measures for each task were the time taken to complete the task, the number of sheets viewed by the participant, the number of ticks and expansions (viewing the filtered image in the overall design) a participant makes, and the score for each design.

3.1 Quality and Efficiency Results

Some summary statistics about Taster and RFC are presented in table 1. The data were analysed using SPSS v17.0. With all measures, it was noted that there was substantial positive skew in many of the task conditions. For this reason, non-parametric tests were used to compare the conditions. Additionally, as each participant completes four separate tasks but not in a full-factorial design, the measures for each task are considered separately as a between-subjects design for the two conditions. However, to account for possible correlations between the performance on the different tasks, we make a Bonferroni correction. Thus, for all

Condition	Mean Score	Mean Ticks	Mean Expands	Mean Time (s)	Mean Sheets
RFC	3.23	17.59	43.07	497.21	9.48
Taster	3.47	15.97	39.53	463.55	6.97

Table 1. Mean score per chosen design; average number of ticks per design task; mean number of expansions per design task; mean time per design task; mean number of sheets of 24 images per design task, for both RFC and Taster conditions.

Design Task	RFC	Taster
Gallery	536 (251)	504 (123)
Magazine	409 (92.0)	473 (242)
Facebook	373 (179)	372 (128)
BookCover	673 (298)	508 (230)

Design Task	RFC	Taster
Gallery	6.87 (2.48)	4.71 (2.37)
Magazine	7.73 (2.69)	6.07 (2.76)
Facebook	12.1 (5.72)	10.5 (6.29)
BookCover	11.50 (5.20)	6.33 (2.82)

Table 2. a) mean (standard deviation) of task times in seconds for each task in each condition. b) mean (sd) of number of sheets viewed for each task in each condition.

tests, we use the Mann-Whitney test with significance level $\alpha = 0.05/4 = 0.0125$, and SPSS was used to produce exact p values. The mean and standard deviations of the task times are given in table 2a. There are modest differences between the means, but overall there are no significant differences: for the gallery task, $U = 102$, $p = 0.914$; for the magazine task, $U = 95.5$, $p = 0.691$; for the Facebook task, $U = 104$, $p = 0.974$; and for the book cover task, $U = 57.5$, $p = 0.038$. Note that the book cover task is tending towards being completed significantly quicker for the Taster condition. For the number of filters viewed, in all tasks, participants viewed fewer sheets in the Taster condition than in the RFC condition. The means and standard deviations are shown in table 2b. Significant differences are seen for Gallery ($U = 49$, $p = 0.011$) and for BookCover ($U = 40$, $p = 0.003$) but not for Magazine ($U = 63.5$, $p = 0.067$) or Facebook ($U = 86.5$, $p = 0.429$).

Analysing scores is more complicated, as each participant was able to tick and therefore score as many designs as they wished. The number of ticks depends on personal strategies for using the system, hence it would be useful to see if participants differed in the number of ticked designs in one condition over the other. As tasks were fully counterbalanced across the two conditions, for each participant, the number of scores produced was summed across tasks in each condition. In the RFC condition, participants ticked a mean of 17.59 designs, whereas in the Taster condition, the mean is lower, at 15.97. A Wilcoxon Signed Ranks test indicates that these differences are not significant ($Z = -1.14$, $p = 0.261$). Taking instead the average (mean) of all scores for each participant over two tasks in a single condition, the mean score in the RFC condition is 3.23 and in the Taster condition, it is higher, at 3.47. The difference in mean scores over the two conditions is significant (Wilcoxon $Z = -2.649$, $p = 0.007$). It is worth noting though that looking only at the number of score 5s, i.e., the images people would definitely use, a similar analysis showed no significant difference.

In summary, users will on average be more satisfied (in terms of scores) with designs produced by Taster. However, they will not necessarily tick more designs nor give more designs the maximum score in a session. While users will not necessarily finish quicker, they might be presented with fewer images in carrying out a design task with Taster than RFC, but this is task dependent.

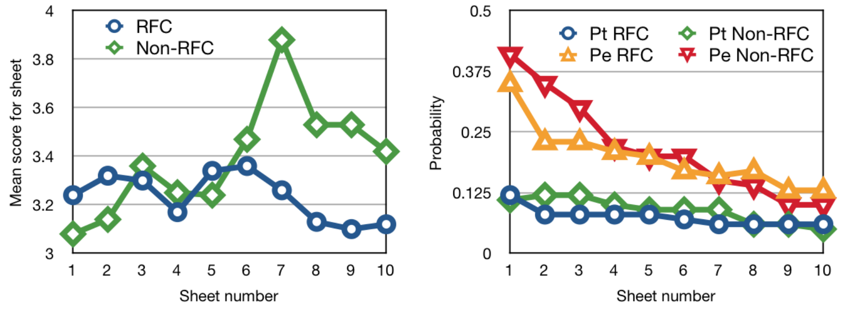

Fig. 3. a) Mean sheet scores for the RFC and the Non-RFC submethods in the Taster method. b) tick and expand probabilities per sheet for RFC and Non-RFC submethods.

3.2 User Behaviour Analysis

As Taster is a combination of six filter selection submethods, we can look at the individual contribution each submethod makes. Splitting the submethods into those using the participant's choices (Non-RFC) and the RFC method, looking at figure 3a, we see that for Non-RFC methods, the mean score for designs ticked in sheets 1 to 5 is 3.23, whereas in sheets 6 to 10, the mean score is 3.4. The same effect is not present in the sheets produced by the RFC method. This suggests that users may appreciate the ability to drive the search with their tick choices. Let us define the *probability of ticking,* $p_t(n)$, for sheet number n as the number of images ticked on the n-th sheet across all sessions divided by 24 (the number of images shown on any sheet). Plotting this in figure 3b, we see that for both RFC and Non-RFC methods, $p_t(n)$ consistently falls as n increases from 1 to 10. Defining the probability of expanding a design similarly as $p_e(n)$, we see that this also decreases over sheets 1 to 10. This suggests that, in general, participants became more discerning about the images they expanded and ticked as they progressed through the first 10 sheets. After sheet 10, the pattern is less clear, perhaps due to the small number of sessions that lasted that long. Table 3 shows that ranking the methods by mean image score is equivalent to ranking them by p_t and (almost) by p_e. This suggests a ranking by popularity, i.e., if participants are ticking/expanding more designs during a session, they will give a higher score on average to the designs they choose. We also note that the two evolutionary techniques (CO and MT) perform the best in terms of the mean score that users ascribe to the designs they produce.

To further analyse the difference between the submethods, we investigated the fidelity of the image filters. For a given filter f, we let $D(f)$ denote the average Euclidean RGB-distance between pairs of pixels in the original and filtered image at the same co-ordinate, normalised by division by the maximum RGB-distance possible. Note that we have experimented with measures based on the HSV colour model, but we saw little difference in the results. For a given design session, S, we let $D(S)$ denote the average of $D(f)$ over all the filters f shown to the user in the session. We let $T(S)$ be the average of $D(f)$ over all the filters ticked by the user in session S. We also denote $P(S)$ as the average Euclidean RGB-distance between pairs of filters (f_1, f_2) in a session S, where f_1 is a filter

Rank	Method	Mean Score	p_t	p_e	Mean $D(S)$	Mean $T(S)$	Mean $P(S)$
1	CO	3.92	0.20	0.34	0.33	0.25	0.32
2	MT	3.73	0.17	0.33	0.35	0.27	0.33
3	CS	3.30	0.09	0.27	0.37	0.30	0.37
4	RFC	3.13	0.07	0.20	0.47	0.35	0.51
5	MFC	3.13	0.06	0.16	0.45	0.35	0.5
6	TS	3.00	0.04	0.19	0.47	0.37	0.51

Table 3. Taster submethods ranked by mean score; probability of being ticked (p_t) or expanded (p_e); mean distance from original $D(S)$; mean distance of ticked filters from original $T(S)$; mean distance from the ticked filters in the previous sheet $P(S)$.

ticked by the user in sheet n and f_2 is any of the 24 filters *shown* to the user in sheet $n+1$. Table 3 shows $D(S)$, $T(S)$ and $P(S)$ for the submethods used in the Taster sessions. We see that the mean score increases as $P(S)$ decreases, hence participants seem to appreciate filters more if they are more similar to those ticked in the previous sheet. Also, the mean score decreases as $D(S)$ increases, which suggests that participants may have preferred more conservative image filters, i.e., which change the original image less. This is emphasised by the fact that in all but 3 of the 116 design sessions, $T(S)$ was less than $D(S)$, i.e., participants ticked more conservative filters on average than those presented to them in 97% of the design sessions. The extent of this conservative nature differs by design task: Gallery: $D(S) = 0.44$, $T(S) = 0.34$; Magazine: $D(S) = 0.44$, $T(S) = 0.28$; Facebook: $D(S) = 0.46$, $T(S) = 0.30$; BookCover: $D(S) = 0.45$, $T(S) = 0.35$. Participants were particularly conservative with the Magazine and Facebook tasks, as these require the filtering of faces, which was generally disliked (as expressed through some qualitative feedback we recorded).

4 Experiment 2

To explore the observation that scores seem to be correlated with the fidelity of the filters, we implemented further retrieval techniques which manage the overall fidelity of filters presented to users. In particular, we implemented another hybrid technique, Evolution (EVO), which returns 8 filters produced by CO, 8 filters produced by MT and 8 filters produced by RFC. This choice was motivated by the fact that CO and MT were appreciatively the best submethods from experiment 1. We produced two variants of EVO to test against it. Firstly, the EVO-S method replaces the 8 filters produced by RFC with filters chosen from the library in such a way that the average $D(f)$ value for the filters on each sheet remains static at 0.25. This choice was motivated by 0.27 and 0.25 being the mean of the $D(f)$ values over the ticked filters produced by the CO and MT submethods respectively. The EVO-D method is the second variant. In order to supply filters for sheet number n, this method calculates the average, A, of $D(f)$ over the ticked filters on sheet $n-1$. Then, EVO-D chooses 8 filters from the library to replace those produced by the RFC submethod in EVO, in such a way that they each have a $D(f)$ value as close to A as possible.

The aim of the second experiment was to test the hypothesis that EVO-S, EVO-D or both would be an improvement on the plain EVO method. A similar experimental setup as before was employed, involving 24 participants, asked to undertake 6 new design tasks, namely: more stationery; another gallery; another magazine cover, shown in figure 3; a poster; a calendar; and a menu (note that we used no faces in the designs, to avoid any biasing as in the first experiment). The EVO, EVO-S and EVO-D methods were balanced around the six design tasks evenly, so that each participant was given each method twice. The results are shown in table 4. A statistical analysis revealed that the hypotheses that EVO-S or EVO-D is better (in terms of eciency and mean score) are not supported by the data. In fact, EVO has a higher mean score and lower mean time than both the other methods. We speculate that EVO-S and EVO-D's balancing of the average image fidelity results in many very high fidelity filters (i.e. very low RGB distance) being introduced to achieve the balance, and that in general these filters are less satisfying to the user than the random selection used by EVO.

Method	D(f)	Score	Ticks	Exps	Time	Sheets
EVO	0.28	3.45	18.9	30.6	347.1	6.3
EVO-S	0.25	3.32	22.2	32.3	392.0	6.7
EVO-D	0.27	3.15	20.1	27.8	362.2	6.4

Table 4. Statistics for exp. 2: Mean RGB distance per design (fidelity); Mean score per chosen design; mean ticks per design task; mean expansions per design task; mean time (s) per design task; mean sheets viewed per design task.

5 Conclusions and Further Work

To the best of our knowledge, there has been little study of user behaviour with browsing systems for creative tasks such as evolutionary art [7]. A notable exception is [5], where user interaction with the NEvAr evolutionary art tool is described. We introduce the phrase *Objet Trouvé browsing* to acknowledge the push and pull between software leading the user and the user leading the software in such systems. This raises the question of whether software could learn from the user, or more ambitiously: take a creative lead in design projects. Such behaviour might be appreciated by novice or amateur designers perhaps lacking inspiration. We are taking deliberately small steps towards such creative systems with experiments involving amateur designers to understand the nature of both the different methods and user behaviour with respect to those methods. In particular, we started with the straightforward hypothesis that using intelligent techniques to deliver new image filters based on those chosen by the user would be an improvement over supplying the filters randomly. (The truth of this is rather taken for granted in evolutionary art and image retrieval systems).

Working with image filters enables us to compare and contrast methods from different areas of computing: database, image retrieval and evolutionary methods in browsing for resources (filters) in design tasks. In experiment 1, we found that more intelligent methods will lead to greater satisfaction in the designs produced and may lead to the completion of the design task with less effort (i.e., having to consider fewer possibilities). We also observed some user behaviours such as becoming more discerning as a session progresses and appreciating the progression afforded by the intelligent techniques. Furthermore, while there is some correlation between filter fidelity and user satisfaction, we were unable to harness this for improved browsing techniques, as shown in experiment 2. Simply giving users *more of what they like*, whether statically or dynamically is not sophisticated enough, raising interesting questions about managing novelty.

We plan to study other browsing systems, e.g., [10], which employs emotional responses to web pages, and other evolutionary image filtering systems, e.g., that of Neufeld, Ross and Ralph (chapter 16 of [7]), which uses a fitness function based on a Bell curve model of aesthetics. Moreover, to improve our experiments, we will study areas of computer supported design such as: the influences of reflection and emergence [8]; the use of analogy and mutation [4]; and how serendipity can be managed [1]. We will test different browsing mechanisms involving different image analysis techniques such as edge information and moments, and measures based on novelty, such as those prescribed in [6]. Despite the failure of the EVO-D method in experiment 2, we believe that software which dynamically employs information about a user's behaviour to intelligently suggest new artefacts can improve upon less sophisticated methods. In particular, we intend to use the data from experiments 1 and 2 to see whether various machine learning techniques can make sensible predictions about user preferences during image filtering sessions. Our ultimate goal is to build and investigate software which acts as a creative collaborator, with its own aesthetic preferences and goals, able to work in partnership with both amateur and expert designers.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable input. This work is supported by EPSRC grants EP/F067127 and TS/G002835.

References

1. M Andre, M Schraefel, J Teevan, and S Dumais. Designing for (un)serendipity. In *Proceedings of the 7th ACM Creativity and Cognition Conference*, 2009.
2. S Boyd Davis and M Moar. The amateur creator. In *Proceedings of the 5th ACM Conference on Creativity and Cognition.*, 2005.
3. S Colton and P Torres. Evolving approximate image filters. In *Proceedings of the Evolutionary Art and Music Workshop, EvoMUSART*, 2009.
4. J Gero and M Maher. Mutation and analogy to support creativity in computer-aided design. In *Proceedings of CAAD Futures*, 1991.
5. P Machado and A Cardoso. NEvAr – The assessment of an evolutionary art tool. In *Proc. AISB Symp. on Creative and Cultural Aspects of AI and Cog. Sci.*, 2000.
6. G Ritchie. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines*, 17:67–99, 2007.
7. J Romero and P Machado, editors. *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music.* Springer, 2007.
8. D Schon and G Wiggins. Kinds of seeing and their functions in designing. *Design Studies*, 13(2):135–156, 1992.
9. B Schneiderman. Creativity support tools: Accelerating discovery and innovation. Communications of the ACM, 50(12):20–32, 2007.
10. I Takeshi, O Jun, M Koji, and M Satoshi. Studies on user adaptive browsing system on the internet. *SIG-KBS*, 43:81–86, 1999.
11. P Torres, S Colton, and S Rüger. Experiments in example-based image filter retrieval. In *Proceedings of the SAMT Cross-Media Workshop*, 2008.