Brock University

Department of Computer Science

Virtual Photography Using Multi-Objective Particle Swarm Optimization

William Barry and Brian J. Ross

Technical Report # CS-14-01
January 2014

Brock University
Department of Computer Science
St. Catharines, Ontario Canada L2S 3A1
www.cosc.brocku.ca

Virtual Photography Using Multi-Objective Particle Swarm Optimization

William Barry
william.barry.mail@gmail.com

Brian J. Ross
bross@brocku.ca

ABSTRACT

Particle swarm optimization (PSO) is a stochastic population-based search algorithm that is inspired by the flocking behaviour of birds. Here, a PSO is used to implement swarms of cameras flying through a virtual world in search of an image that satisfies a set of compositional constraints, for example, the rule of thirds and horizon line rules. To effectively process these multiple, and possible conflicting, criteria, a new multi-objective PSO algorithm called the sum of ranks PSO (SR-PSO) is introduced. The SR-PSO is useful for solving high-dimensional search problems, while discouraging degenerate solutions that can arise with other approaches. Less user intervention is required for the SR-PSO, as compared to a conventional PSO. A number of problems using different virtual worlds and user-supplied constraints were studied. In all cases, solution images were obtained that satisfied the majority of given constraints. The SR-PSO was shown to be superior to other algorithms in solving the high-dimensional virtual photography problems studied.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Experimentation

Keywords

particle swarm optimization, virtual photography, multi-objective optimization

1. INTRODUCTION

In recent years, research has investigated the creation of high-level tools for assisting photographers in the production of images adhering to artistic rules of composition. *Automated photography* refers to algorithmic techniques for the unsupervised automation of photographic composition, for images taken in both virtual and real-life environments. We adopt the term *virtual photography* to refer to automatic photography within a virtual 3D environment. The automated composition of such images is important, since images should comply to established norms of artistic composition. This technology is also applicable to automated editing of digital photographs. This may result in intelligent photographic assistants built into cameras that modify images to make them more compositionally pleasing and "professional". A related problem is *virtual world exploration* [14], where agents autonomously seek out desirable subject matter within the virtual world.

Particle Swarm Optimization (PSO) is commonly used in virtual photography. Ermetici *et al.* [9, 19] use a PSO to generate images in a virtual environment complying to a set of supplied constraints. These constraints specify the visibility of different game avatars, and their orientation with respect to the camera. Preuss *et al.* [25] apply a niching evolutionary algorithm to the virtual photography problem, in search for a diversity of results that satisfy compositional constraints. Bares and Kim [6] automate rules of composition by considering subject size, view angle, location, depth, exclusion, and occlusion. Bares [5] expanded this work with a *Virtual 3D Camera Assistant* for 3D virtual photography. Virtual photography has been applied to cinematography to assist in composition. Abdullah *et al.* [1] use a PSO to optimize cinematographic images via composition rules. Examples of real-life photographic editing include [3, 4, 8, 22, 27].

This paper presents a new technique for virtual photography. Like Ermetici *et al.* [9, 19], a PSO algorithm is used to harness a swarm of cameras to autonomously explore a virtual 3D world. Each camera will move through the world in search of an image that satisfies a supplied set of aesthetic rules. Since there are usually multiple compositional requirements to be satisfied, a multi-objective approach is taken here, in order to reconcile the multiple aesthetic constraints of images. When composing a picture with a camera, many things must be considered. The subject(s) of interest must be present, and the amount of image real estate it covers must be considered. The subject should be positioned close to one of the four rule of thirds loci. Other objects of interest will have similarly considerations. The horizon rule states that it should reside at a one-third or two-thirds vertical position in the image. The colour palette of the image also may be of consideration. An advantage of multi-objective search is that the user is usually not required to specify how different objective scores are to be measured

.

with respect to each other.

We introduce a new multi-objective PSO algorithm called the Sum of Ranks PSO. The SR-PSO borrows a multi-objective strategy used in high-dimensional genetic algorithms [7], which are used to optimize problems with a large number of conflicting objectives. Besides its ability to handle many problem constraints, images obtained with it are less likely to be *outlier* solutions (ones that satisfy a small minority of objectives).

Obvious applications for this research may be within video games, virtual reality portals (Google Streetview), and 3D building design software. For example, today's games strive to generate complex, realistic virtual worlds. These worlds also contain objects of interest to the game, which require time and effort for creating specific cameras to focus upon. Using the strategy proposed in this paper, game developers could have PSO agents automatically discover objects of interest, and return dramatically composed images of them. More advanced applications of this technology are conceivable. For example, one could imagine the PSO swarm to be autonomous robot helicopters [23].

The paper is organized as follows. Multi-objective problem solving is discussed in Section 2. Section 3 introduces the SR-PSO algorithm, as well as other PSO algorithms used in comparisons in later sections. The composition rules used in this research are discussed in Section 4. The system architecture is described in Section 5. Results and comparative analyses are given in Section 6. Comparisons to related research is given in Section 7. Section 8 gives conclusions and directions for future research.

2. MULTI-OBJECTIVE PROBLEMS

A multi-objective optimization problem is one in which two or more objectives must be optimized [11]. These multiple objectives often interact in complex, nonlinear, and conflicting ways. Often, the best one can hope for is that the majority of objectives can be balanced and satisfied to an acceptable degree.

2.1 Pareto Ranking

The most popular multi-objective scoring strategy is Pareto ranking [13]. Pareto ranking is based on the concept of Pareto dominance.

$$\begin{aligned} A\ dominates\ B \leftrightarrow \quad & (\forall_{obj}\, f_{obj}(A) \leq f_{obj}(B)) \\ & \wedge (\exists_{obj}\, f_{obj}(A) < f_{obj}(B)) \end{aligned}$$

where $f_{obj}(P)$ is the score of objective obj in agent P, $1 \leq obj \leq n$ objectives, and f defines a minimization problem. A dominates B if it is superior to B in at least one objective, and equal to B in the remaining objectives.

Using the definition of Pareto dominance, a population of agents is examined. The undominated agents are assigned the Pareto rank of 1 and removed. The undominated agents from the remaining population are assigned the rank of 2. This continues until all agents are ranked. Lower ranks are preferred. Example rankings are given in Table 1.

Pareto ranking becomes ineffective with more than 5 objectives, because most of the population becomes undominated. In such cases, the sum of ranks scoring strategy should be considered.

	Objectives				Ranks			
	O_1	O_2	O_3	P	R_1	R_2	R_3	SR
1)	10	5	80	1	5	1	2	2.25
2)	6	50	80	3	4	3	2	2.55
3)	3	60	20	3	3	4	1	2.1
4)	2	40	20	2	2	2	1	1.4
5)	1	40	20	1	1	2	1	1.2
max:					5	4	2	

Table 1: Example of sum of ranks calculation. O_i are raw scores (lower preferred), P are the Pareto ranks, R_i is the rank of O_i, and SR is the (normalized) sum of ranks.

2.2 Sum of Ranks

The sum of ranks (or average rank) was originally proposed for solving high-dimensional multi-objective problems [7, 12]. Consider a problem with K objectives, each of which must be minimized. Each individual j in the population is assigned a rank vector, in which each objective score for the objective is given a rank value $r(1 \leq r \leq N)$ for population size N: $R^j = < r_1, ..., r_K >$. Individuals with tied scores have the same rank value. These rank scores are always set relative to the current population. Hence, a rank $r_i = 1$ will be found in the vector of the individual that has the best score in objective i in the population. The sum of ranks score S^j for each individual in the population is then:

$$SR^j = \sum_{i=1}^{K} \frac{w_i r_i}{max_i}$$

where w_i is an optional weight (default 1.0), and max_i is the maximum rank for each objective. Normalizing the ranks by max makes the rankings uniform amongst the different objectives.

Table 1 shows some example calculations. Rows 1-5 represent the population. Objective scores are to be minimized. Note how the ranks R_i replace the raw objective scores with rank orderings. SR is the sum of the normalized ranks in that row, which are found by dividing each R_i by the maximum rank for that objective (column). Note that the relative orderings of the SR scores and Pareto ranks (P) differ.

3. PARTICLE SWARM OPTIMIZATION

3.1 The Basic PSO Algorithm

Particle swarm optimization (PSO) is a population based search algorithm inspired by the social behaviour known as flocking [15]. Swarm individuals (particles, agents) have two important attributes: velocity and position. These two attributes are best thought of as a K size vector, for a problem with K values to optimize. Throughout a simulation, each individual updates its own position by its own velocity, and the velocity is modified by one value from within the population and one value from itself. The influence from the population is a global best *gbest*, which is the best individual from within the swarm. The influence from an individual agent is its personal best *pbest*, which is the best solution that the individual has seen.

More formally, the different attributes that contribute to the change of a particles position or state are:

$\vec{p_i}$:	particle position or state
$\vec{v_i}$:	particle velocity
w:	an inertia value to control the velocity
$\vec{pbest}$:	particle's personal best solution
r_1:	random number between (0,1)
c_1:	constant constraint for *pbest*
$\vec{gbest}$:	swarm's best solution
r_2:	random number between (0,1)
c_2:	constant constraint for *gbest*

Using these attributes, individuals in a swarm update their velocity:

$$\vec{v_i} = w\vec{v_i} + c_1 r_1(\vec{pbest} - \vec{p_i}) + c_2 r_2(\vec{gbest} - \vec{p_i}) \quad (1)$$

A particle's position is then updated:

$$\vec{p_i} = \vec{p_i} + \vec{v_i} \quad (2)$$

Algorithm 1: Particle swarm optimization

```
1  for $i = 1$ to $N$ do
2      $\mathbf{p}_i \leftarrow$ randomize;
3      $\mathbf{pbest}_i \leftarrow \mathbf{p}_i$;
4      if $f(\boldsymbol{p}_i) < f(\boldsymbol{gbest})$ then
5          $\mathbf{gbest} \leftarrow \mathbf{p}_i$;
6      end
7      $v_i \leftarrow$ randomize;
8  end
9  while Termination criteria not met do
10     for $i = 1$ to $N$ do
11         for $d = 1$ to $D$ do
12             Generate random numbers $r_1$, $r_2$;
13             Update $\mathbf{v}_i$ with Eqn. (1);
14         end
15         Update $\mathbf{p}_i$ with Eqn. (2);
16         if $f(\boldsymbol{p}_i) < f(\boldsymbol{pbest}_i)$ then
17             $\mathbf{pbest}_i \leftarrow \mathbf{p}_i$;
18             if $f(\boldsymbol{p}_i) < f(\boldsymbol{gbest})$ then
19                 $\mathbf{gbest} \leftarrow \mathbf{p}_i$;
20             end
21         end
22     end
23 end
```

Algorithm 1 outlines the basic PSO. The algorithm assumes a swarm size of N. Termination criteria will either be a maximum number of iterations of the algorithm, or the detection of a particle whose fitness is within a minimal requirement. The conventional PSO outlined is distinguished by the fitness evaluator f returning a single-value numeric result, where lower values are preferred. Should the problem in question have multiple constraints to satisfy, then a weighted sum formula is used:

$$f(\mathbf{p}) = \sum_{j=1}^{K} w_j f_j$$

where there are K constraints $\mathbf{p} = (f_1, ..., f_K)$, and w_j is a weight factor.

The next two sections present alternative ways of calculating the fitness of a particle with multiple constraints.

3.2 The SR-PSO Algorithm

A new PSO algorithm, the sum of ranks PSO (SR-PSO), is introduced. It is largely the same as the PSO in Algorithm 1. The main difference is the method by which fitness scores are found in the multi-objective search space, where it replaces the weighted sum formula (line (13)) with the sum of ranks scoring method discussed in Section 2. In order to find $pbest_i$ and *gbest*, the SR-PSO uses *archives* of particle's multi-objective score vectors for determining sum of ranks scores. Each particle has its own archive of personal best vectors, while the swarm has an archive recording best vectors seen overall. The sum of ranks calculation is applied to these archives, and the lowest sum of ranks scores are used to identify the **pbest** for each particle, and **gbest** for the swarm.

To update the simulation best **gbest**, the sum of ranks algorithm determines the current **gbest** from the population and appends this agent to the simulation best archive. If there is more than one current best when analyzing the population (multiple individuals have the same sum of ranks value), all the tying agents are added to the archive. This archive is then sorted by the sum of ranks, and the best from the archive is chosen as the simulation best. If there is more than one best, one of these is randomly selected. Similarly, each particle has a personal archive, which is processed similarly to determine the **pbest** value for a particle. Since archives might grow very large during long runs, a user-supplied size limit is placed on the archives.

3.3 The Pareto PSO

The MOP PSO algorithm of Mostaghim and Teich [24] was selected as a comparison algorithm. Briefly, this algorithm assigns each agent in the swarm a σ value, which is a slope value. Equation 3 solves the σ value for a 2-dimensional problem:

$$\sigma = \frac{f_1^2 - f_2^2}{f_1^2 + f_2^2} \quad (3)$$

N-dimensional problems have higher-dimensional slopes computed. A global archives of Pareto undominated swarm best solutions is maintained, along with their σ values. To determine the **gbest** for a particle, the Pareto undominated particle in the archives with the closest slope to that particle is determined, and is selected as the guide for the particle. If an agent is already considered to be Pareto optimal then it uses itself as the *gbest*. The rationale for this approach is that such a selected guide will direct the particle more directly towards the Pareto front.

The **pbest** is determined in the same manner as the conventional PSO in Section 3.1.

3.4 Bootstrapping

An option called *bootstrapping* is available. Image evaluation rules in Section 4 such as object detection and rule of thirds are not measurable if an object of interest is not seen in an image. When this happens, only rules such as colour palette matching and horizon line return meaningful scores, and the swarm will optimize only these rules, and ignore the others. With bootstrapping, the PSO will repeatedly randomize particles until at least one particle has spotted an object of interest. Once such an object is found, object detection and rule of thirds will be measurable, and this will improve PSO performance across multiple objectives.

4. COMPOSITION RULES

4.1 Rule of Thirds

The rule of thirds is one of the most important composition rules[18]. An image is subdivided into thirds horizontally and vertically. The areas around the four intersections are positions where dominant objects of interest should reside. To implement the rule of thirds, the pixel mask for objects of interest is determined (Section 4.3 below). If an object of interest is found in the image, the centroid for its visibly rendered pixels (n total) is determined:

$$C_{x,y} = \frac{\sum [x,y]}{n}$$

Then the Euclidean distances between $C_{x,y}$ and each of the four rule of thirds locations is determined. The least distance found is assigned as the rule of thirds score.

4.2 Horizon Line

The horizon line rule also uses the concept of proportion of thirds. The horizon is the delineation line where the sky meets the land. The horizon line rule states that the horizon should reside at a proportion of either one-third or two-thirds vertically in an image[17]. The implementation of the horizon line rule first requires a mask image to be rendered, with the mask indicating the ground and sky in the scene. The horizon line, if visible, is easily determined from this mask. Note that cameras are not permitted to roll, and so the horizon will always be horizontal. Next, the shortest vertical distance between the horizon and one of the horizon rule lines (at the one-third and two-thirds positions) is calculated. The smaller of these distances is used as the horizon rule score.

4.3 Object of Interest

Subject matter is an important factor in photography. The automation of object recognition is an advanced computer vision topic beyond the scope of this research. Since we are working in a virtual 3D environment, however, we exploit the system's knowledge of the models rendered in an image. Each object is rendered with a colour-coded mask value. This mask image is examined it to see whether a particular object of interest is rendered within it. Furthermore, the area of that object relative to the total image can be determined. Using this idea, we can specify a desired area that an object of interest should have within an image.

4.4 Colour Palette

Colour is an important aspect of aesthetics, and a means to identify subject matter. The *Colour Histogram Quadratic Matching* (CHISTQ) algorithm [26] is used to determine the colour match between a rendered image from the PSO and the target colour image. A colour target image is supplied to the system. This image specifies a desired colour palette, as well as relative amount of each colour. A quantized colour histogram is generated for the colour target image at the beginning of a run. When evaluating a PSO image, the image is processed to generate another quantized colour histogram. Then the quadratic distance between the histograms is computed, which measures each colour of one histogram with every colour in the other histogram, producing a relaxed measurement of colour correspondence.

5. SYSTEM ARCHITECTURE

Table 2: Rotation and field of view constraints

$Rotation_X$	$30° \leq R_x \leq 150°$
$Rotation_Y$	N/A (camera roll not allowed)
$Rotation_Z$	$0° \leq R_z \leq 360°$
FoV	$40° \leq fov \leq 72°$

Each agent in the swarm is a camera in a virtual environment capable of generating an image based on the direction it is facing. Three properties modify the camera: *Location* (x,y,z), *Rotation*, and *Field of View (FoV)*. To prevent gimbal lock (rotation issues from using Euler angles) in the camera from the *Rotation* and undesired rendering artifacts from the *FoV*, constraints were included (Table 2). These properties are stored in a vector, which every PSO agent uses to determine the camera location and viewpoint in the environment.

We implemented a virtual 3D environment with 3ds Studio Max [2]. A custom plug-in was written in *C#* and *MaxScript* to manipulate an agent's camera in the scene and render the current viewport of the camera. Three independent modules (Image Analysis, Particle Swarm Optimization and 3ds Max) are loaded and executed in parallel. The Image Analysis component analyzes an agent's image and return fitness values. The PSO algorithm uses these fitness values to update the swarm, by modifying every agent's state vector in the swarm. Then each agents vector is transferred to 3ds Max. 3ds Max updates the *Location*, *Rotation*, and *FoV* of the respective camera within the world, and renders a new image. Once all agents are updated, the process iterates until the simulation is complete.

6. RESULTS

Table 3: PSO parameters

Number of Runs	30
Swarm size	25
Max iterations	100
Inertia	0.8
Personal best constraint	0.45
Global best constraint	0.5

Table 4: Fitness ranges for objectives. Low scores preferable.

Fitness Objective	Fitness Range
Object Detection (OD)	$0 \leq f \leq 153600$
Rule of Thirds (ROT)	$0 \leq f \leq 800$
Horizon Line (HZ)	$0 \leq f \leq 240$
Colour Similarity (CS)	$0.0 \leq f \leq 1.0$

The virtual worlds used here are typical of those found in video games and computer animations. The problem specifications used have between 6 to 10 objectives. These are potentially challenging problems for the swarm to satisfy, given that conflicting objectives will arise.

Table 3 are the PSO parameters used for all these examples, and Table 4 give the score ranges.

6.1 Sunrise

Table 5: Sunrise objectives

1.	Object detection: boat on land (10%)
2.	Object detection: boat in water (10%)
3.	Object detection: Sun (5%)
4.	Rule of thirds: boat on land
5.	Rule of thirds: boat in water
6.	Rule of thirds: sun
7.	Horizon line
8.	Colour similarity

This example of an ocean sunrise scene uses 8 objectives in total (Table 5). There are two boats and a Sun as objects of interest, all of which should be composed with the rule of thirds. Figure 1(a) shows one view of the environment.

Two result images are shown in Figure 1 (b) and (c), and their objective scores are shown in Table 6. Image (b) is typical of the results obtained, and has tried to satisfy most of the requirements. However, it is difficult applying the rule of thirds to both ships and the sun simultaneously, and so the boat in the water is off on this. On the other hand, image (c) is an example of an anomalous result image. It has missed both ship objects (see the poor OD and ROT scores for the ships in Table 6), but has obtained better Sun ROT and horizon scores. Although the sum of ranks scoring usually discourages such anomalies, they can sometimes arise. Therefore, PSO algorithms are almost always executed many times on a problem, in order to generate a set of solutions for consideration.

Images (d) through (f) in Figure 1 are screen captures from an animation of the swarm in action. The initial state of the swarm is shown in (d), where the particles are scattered randomly throughout the environment. As the simulation proceeds in (e), the swarm begins to converge as the particles detect objects of interest. The swarm converges in (f). It is common to see the final swarm converge together, which shows convergence in the PSO search.

A comparison of the bootstrapped sum of ranks (SRB), bootstrapped Pareto PSO from [24] (PB), and normal weighted-sum PSO bootstrapped (NB) was done. Given the many objectives to consider, a statistical analysis was undertaken to clearly summarize the performance. Each algorithm was run 30 times, and 30 solutions were obtained. (For the Pareto PSO, an arbitrary rank 1 solution was taken from each run.) The non-parametric Mann-Whitney U test was used to test when one algorithm performed statistically better than another on a particular objective (95% confidence level). The number of times each algorithm outperformed another on a each objective was tallied (Table 7). The table shows that both the sum of ranks PSO and conventional PSO outperformed the Pareto PSO on half or more of the objectives. The sum of ranks PSO also did better than the conventional PSO on half the objectives. This gives evidence that the sum of ranks PSO is indeed suitable for high-dimensional multi-objective search, as seen in the problem studied.

Table 6: Objective scores for result images in Figure 1.

Objective	(b)	(c)
OD boat land (%)	17.0	–
OD boat water (%)	19.3	–
OD Sun (%)	9.9	9.9
ROT boat land	0.33	800.0
ROT boat water	10.87	800.0
ROT Sun	75.89	22.32
HZ	18	2
CS	0.011	0.019

Table 7: Frequency that algorithm beat others on Sunrise scene objectives (95% confidence). Table values represent # objectives in which algorithm A beat algorithm B (maximum 10).

	B		
A	SRB	NB	PB
SRB	-	4	5
NB	1	-	4
PB	0	0	-

6.2 Space

Table 8: Space objectives

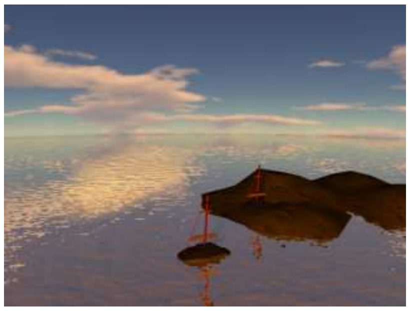
1.	Object detection: boat on land (10%)
2.	Object detection: boat in water (10%)
3.	Object detection: red moon (5%)
4.	Object detection: blue moon (5%)
5.	Rule of thirds: boat on land
6.	Rule of thirds: boat in water
7.	Rule of thirds: red moon
8.	Rule of thirds: blue moon
9.	Horizon line
10.	Colour similarity

This example enhances the previous scene, by replacing the Sun with two moons. The objectives are similarly defined (Table 8). Figure 2 shows 3 solution images, and Table 9 shows the scores. There is no perfect solution, given that the swarm struggles to position all four objects at one of the rule of thirds loci, and satisfy object detection requirements. All three solutions show different compositional compromises in satisfying the competing criteria.

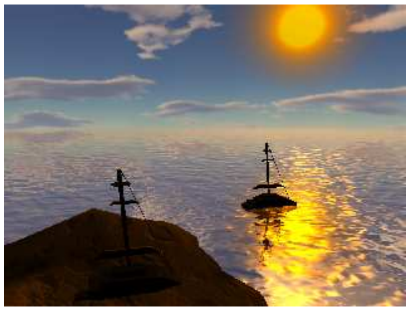
Another comparison of the 3 PSO algorithms was done, and the results are shown in Table 10. Here, the Pareto PSO was further outmatched by both the conventional PSO and sum of ranks PSO. Once again, the sum of ranks PSO statistically outperformed the other algorithms on more objectives simultaneously.

6.3 Table Conversation

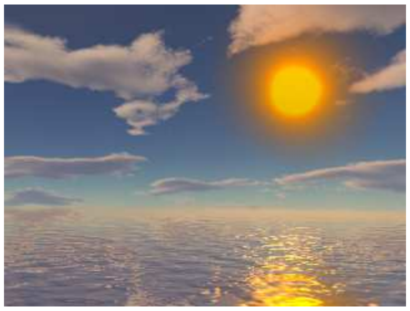
This example highlights the effect of colour matching as a problem constraint. Here, we are to find two specific people in a crowd of eight. The two people of interest will be identified via the colour similarity test, based on their clothing colour. Therefore, the PSO cannot rely on the OD algorithm to uniquely identify the pair of individuals, but instead must use the CS test to find them. The scene was set up so that the four male faces have the same model mask identifier, and



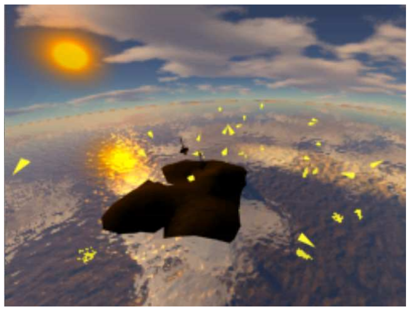
(a) Environment.



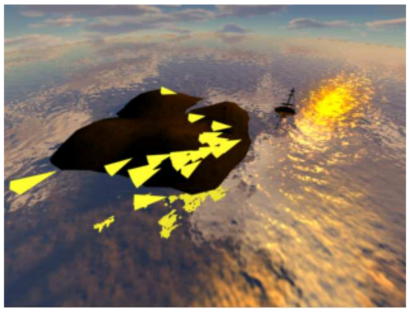
(b) Result 1.



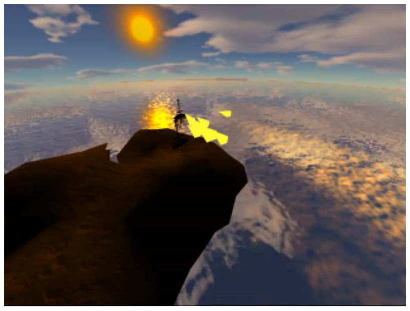
(c) Result 2. Ships missing, but Sun and horizon composed well.



(d) Swarm in initial scattered state.



(e) Swarm starts to converge.



(f) Swarm close to convergence.

Figure 1: Sunrise experiment.

Table 9: Objective scores for result images in Figure 2.

Objective	(a)	(b)	(c)
OD boat land (%)	18.4	16.2	18.1
OD boat water (%)	19.0	19.2	19.4
OD red moon (%)	9.9	10.0	10.0
OD blue moon (%)	9.9	10.0	10.0
ROT boat land	1.33	43.52	13.62
ROT boat water	79.40	18.11	1.94
ROT red moon	31.48	37.67	31.69
ROT blue moon	45.88	24.40	37.42
HZ	11	27	1
CS	0.023	0.028	0.005

Table 10: Frequency that algorithm beat others on Space scene objectives (95% confidence). (See Table 7.)

	B		
A	SRB	NB	PB
SRB	-	4	8
NB	0	-	9
PB	0	0	-

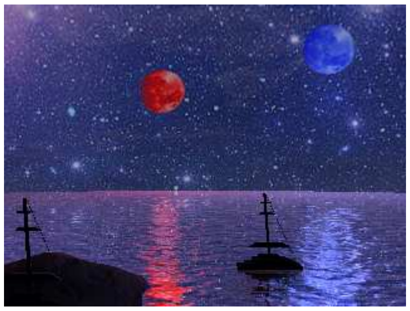
Table 11: Table Conversation objectives

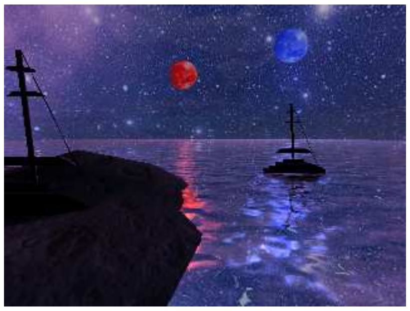
1.	Object detection: male face (10%)
2.	Object detection: female face (10%)
3.	Rule of thirds: male face
4.	Rule of thirds: female face
5.	Horizon line
6.	Colour similarity

similarly for the females. This means that all the people can satisfy the OD requirements, which forces the colour test to come into consideration. Table 11 lists the objectives for this simulation. The colour similarity image is also placed as a bitmap on the table (see Fig. 3, image (c)), which acts as a decoy to trick the swarm when evaluating colour similarity.

Three results are shown in Figure 3, and their objective scores are in Table 12. In image (a), the simulation found multiple faces for both female and male but managed to find the two people of interest (both partially obscured by man in green shirt) by using the colour similarity. Image (b) manages to find multiple males and females in the scene, but focuses on the female subject in the yellow shirt. Image (c) finds the people of interest, and composes them well with the rule of thirds. The "decoy" bitmap on the table, although visible in the image, did not distract the swarm from obtaining good OD scores.

The above images, and the scores in Table 12, show that solutions obtained in multi-objective problems are often compromises. As the scope and number of objectives increases, a balance must be found in satisfying objectives. For example, image (b) has the best object detection and rule of thirds scores for the male, but has poor OD and ROT scores for the female. These "democratic compromises" naturally arise with the sum of ranks scoring strategy.


(a) Result 1.


(b) Result 2.

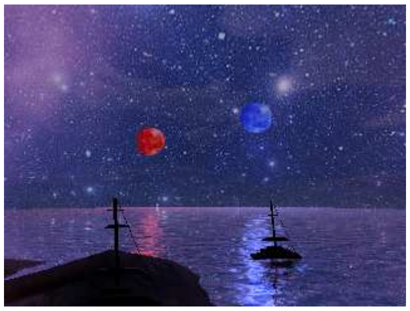
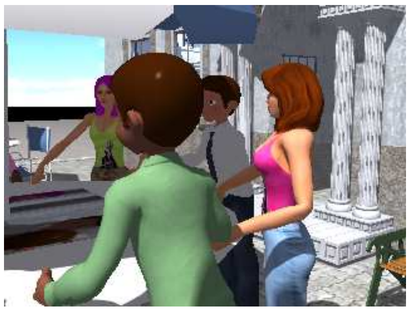
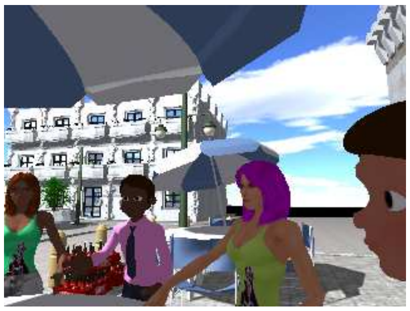

(c) Result 3.

Figure 2: Image results for the Space environment.


(a) Result 1: Top in CS, HZ.


(b) Result 2: Top in OD male & female,

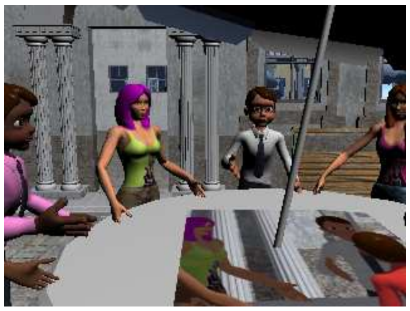

Result 3: Top in OD female.

Figure 3: Image results for the Table Conversation scene.

Table 12: Objective scores for result images in Figure 3.

Objective	(a)	(b)	(c)
OD male (%)	19.7	**18.1**	19.5
OD female (%)	19.9	**19.7**	19.8
ROT male	10.79	**7.45**	8.54
ROT female	7.01	21.94	**0.10**
HZ	**0**	**0**	3
CS	0.061	0.086	**0.110**

7. COMPARISON TO RELATED WORK

Gaspero *et al.* [9, 19] introduced the use of particle swarms for virtual photography. They use upwards of 10 rules, which are divided into *hard constraints*, which must be satisfied before an image can be accepted, and lower-priority *soft constraints*. Their conventional PSO requires the user to supply weights to specify the relative importance of objectives to each other. Compared to our system, their rules require the user to specify a much lower level of detail, for example, positions within the frame, object distances, and so on. We did not require weights for the SR-PSO (although they are optional). Our bootstrapping idea is analogous to Gaspero *et al.*'s hard constraints.

Abdullah *et al.* [1] use a PSO for virtual photography. The swarm produces several photos as a starting point from which a human director can use. They use rule of thirds, diagonal dominance, visual balance, and others. One major difference between their work and ours is that they use a conventional PSO which, again, requires user-supplied weights to balance objectives.

Other work in automatic composition and virtual photography is related to our research. Lino *et al.'s* virtual cinematography [21] applies various aesthetic rules to evaluate frames of animations in virtual environments. Liu *et al.* [22] apply a PSO to automatically improve the compositions of photographs. Bares and Kim [6] use composition rules such as the horizon line test to automatically control cameras in a virtual environment.

8. CONCLUSIONS

A new multi-objective swarm technique for automatic virtual photography has been proposed. The SR-PSO is capable of reconciling the high number of conflicting objectives found in virtual photography problems. Empirical comparisons of it against a conventional PSO and a Pareto-based PSO showed its effectiveness on the problems studied. Furthermore, problem-specific weights were not required. This contrasts to earlier work with conventional PSO's, which require user micromanagement of compositional specifications, for example, low-level measurement requirements of compositions, and weights for combining objectives in the PSO.

All experiments were done on a dual quad core Intel running at 2.9 GHz with 4G RAM. Runs took between 11 and 21 hours to process, largly due to the raytraced rendering required. Although execution time of the PSO system was not of primary concern in this research, future work should consider execution efficiency.

There are many other directions for future work. The introduction of more intelligent computer vision analyses is worth consideration, for example, automatic face recognition. More advanced virtual photography needs computer vision algorithms for object detection and recognition. These would be necessary if PSO photography were to be applied to non-virtual, real-world environments. More intelligent image analyses might also include knowledge-based approaches to virtual/real environment search. More intelligent search, perhaps looking for human subjects to photograph, would first look for instances of human habitation (towns, houses, roads, ...) within the virtual or real world, knowing that humans are likely to be found in such settings.

There has been much research in the field of multi-objective PSO [16]. The sum of ranks PSO should be empirically compared to other MOP PSO's on virtual photography problems, as well as other MOP problems in the literature. The issue of swarm diversity is also worth consideration. Adding diversity heuristics to a multi-objective PSO would result in a variety of alternative solution images from a single run.

There is also the opportunity to introduce more advanced rules of composition and aesthetics as found in [10, 17, 18, 20]. However, implementing many of these sophisticated principles of composition requires advanced computer vision and artificial intelligence.

9. REFERENCES

[1] R. Abdullah, M. Christie, G. Schofield, C. Lino, and P. Olivier. Advanced composition in virtual camera control. *Proc. 11th Intl. Conf. on Smart Graphics (SG 11)*, pages 13–24, 2011.

[2] Autodesk. 3ds max 2009. *http : //usa.autodesk.com/*, August 2012.

[3] S. Banerjee and B. Evans. Unsupervised automation of photographic composition rules in digital still cameras. In *in Proc. SPIE Conf. on Sensors, Color, Cameras, and Systems for Digital Photography VI*, pages 364–373, 2004.

[4] S. Banerjee and B. Evans. In-camera automation of photographic composition rules. *Trans. Img. Proc.*, 16(7):1807–1820, July 2007.

[5] W. Bares. A photographic composition assistant for intelligent virtual 3d camera systems. In A. B. et al., editor, *Proc. 6th Intl. Symp. on Smart Graphics, Vancouver, Canada, July 23-25*, volume 4073 of *LNCS*, pages 172–183. Springer, 2006.

[6] W. Bares and B. Kim. Generating virtual camera compositions. In *Proc. 6th Intl. Conf. on Intelligent User Interfaces*, IUI '01, pages 9–12, New York, NY, USA, 2001. ACM.

[7] P. Bentley and J. Wakefield. Finding acceptable solutions in the pareto-optimal range using multiobjective genetic algorithms. In P. C. et al., editor, *Soft Computing in Engineering Design and Manufacturing*, pages 231–240. Springer-Verlag, 1998.

[8] S. Bhattacharya, R. Sukthankar, and M. Shah. A Holistic Approach to Aesthetic Enhancement of Photographs. *ACM Trans. Multimedia Comp., Comm. and Apps.*, 7S(1):1–21, October 2012.

[9] P. Burelli, L. Gaspero, A. Ermetici, and R. Ranon. Virtual camera composition with particle swarm optimization. *Proc. 9th Intl. Symp. on Smart Graphics (SG 08)*, pages 130 – 141, 2008.

[10] P. Burian. *Mastering Digital Photography and Imaging*. Wiley, 2006.

[11] C. C. Coello, G. Lamont, and D. van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer-Verlag, New York, 2006.

[12] D. Corne and J. Knowles. Techniques for highly multiobjective optimisation: Some nondominated points are better than others. In *Proc. GECCO*, pages 773–780, New York, NY, 2007. ACM.

[13] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley, 2009.

[14] S. Desroche, V. Jolivet, and D. Plemenos. Towards plan-based automatic exploration of virtual worlds. *Proc. Int'l Conf. Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG).*, 1:25–32, 2007.

[15] R. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. *Proc. IEEE Intl. Conf. Neural Networks*, 4:1942 – 1948, 1995.

[16] J. Fieldsend. Multi-objective particle swarm optimisation methods. Technical report, TR419, Dept. Computer Science, U. Exeter, 2004.

[17] B. Fier. *Composition Photo Workshop*. Wiley, 2007.

[18] M. Freeman. *The Photographer's Eye: Composition & Design for Better Digital Photos*. Focal Press, 2007.

[19] L. D. Gaspero, A. Ermetici, and R. Ranon. Swarming in a virtual world: A pso approach to virtual camera composition. *ANTS 2008*, LNCS 5217:155–166, 2008.

[20] D. Graham and C. Redies. Statistical Regularities in Art: Relations with Visual Coding and Perception. *Vision Research*, 50:1503–1509, 2010.

[21] C. Lino, M. Christie, R. Ranon, and W. Bares. The directors lens: An intelligent assistant for virtual cinematography. In *Proc. 19th ACM Intl. Conf. on Multimedia*, pages 323–332, New York, NY, 2011.

[22] L. Liu, R. Chen, L. Wolf, and D. Cohen-Or. Optimizing photo composition. *Computer Graphics Forum*, 29:469–478, 2010.

[23] N. Michael, J. Fink, and V. Kumar. Cooperative manipulation and transportation with aerial robots. *Autonomous Robots*, 30(1):73–86, Jan 2011.

[24] S. Mostaghim and J. Teich. Strategies for finding good local guides in multi-objective particle swarm optimization (mopso). In *Proc. IEEE Swarm Intelligence Symp.*, 2003.

[25] M. Preuss, P. Burelli, and G. Yannnakakis. Diversified virtual camera composition. In *Proc. European Conf. on Applications of Evolutionary Computation*, pages 265–274, Malaga, Spain, 2012. Springer-Verlag.

[26] J. Smith and S. Chang. VisualSEEk: a fully automated content-based image query system. In *Proc. 4th ACM Intl. Conf. on Multimedia*, pages 87–98. ACM Press, 1996.

[27] F.-L. Zhang, M. Wang, and S.-M. Hu. Aesthetic Image Enhancement by Dependence-Aware Object Re-Composition. *IEEE Trans. on Multimedia*, 15, June 2013.