C^3 Assignment: Camera Cubemap Color Assignment for Creative Interior Design

Juncong Lin, Pintong Xiao, Yinan Fu, Yubin Shi, Hongran Wang, Shihui Guo, *Senior Member, IEEE*, Ying He, and Tong-Yee Lee, *Senior Member, IEEE*

Abstract—Color design for 3D indoor scenes is a challenging problem due to many factors that need to be balanced. Although learning from images is a commonly adopted strategy, this strategy may be more suitable for natural scenes in which objects tend to have relatively fixed colors. For interior scenes consisting mostly of man-made objects, creative yet reasonable color assignments are expected. We propose C^3 *Assignment*, a system providing diverse suggestions for interior color design while satisfying general global and local rules including color compatibility, color mood, contrast, and user preference. We extend these constraints from the image domain to $\mathbb{R}^3$, and formulate 3D interior color design as an optimization problem. The design is accomplished in an omnidirectional manner to ensure a comfortable experience when the inhabitant observes the interior scene from possible positions and directions. We design a surrogate-assisted evolutionary algorithm to efficiently solve the highly nonlinear optimization problem for interactive applications, and investigate the system performance concerning problem complexity, solver convergence, and suggestion diversity. Preliminary user studies have been conducted to validate the rule extension from 2D to 3D and to verify system usability.

Index Terms—Creative interior colorization, harmony, mood

1 Introduction

Object/scene colorization is a fundamental problem in many application scenarios, especially in interior design. Usually, the user not only specifies appropriate object-wise colors, but also ensures the global color compatibility between objects. A well-designed color theme can make the scene visually pleasing and evoke certain moods. However, due to the lack of a unified color theme representation that can span various domains [1], the user has to specify the desired colors with scribbles [2], [3], [4]. This process is tedious and time-consuming, especially for 3D scenes with complex objects. As the results greatly depend on users' skill and experience, this approach is limited only to professional users. Another choice is to provide a reference image to facilitate the color design process [5], [6], [7]; however, this usually leads to a chicken-or-egg situation. Also, the quality of colorization relies heavily on the reference image.

In the deep learning (DL) era, a popular way of solving the colorization problem is to formulate it as a regression problem and solve it using deep neural networks [8], [9], [10]. Some DL approaches increase the controllability with scribbles [10], [11] or by using reference images [12]. Similar ideas have also been adopted in 3D indoor scenes [13], [14], [15], where colorization is learned using a huge reference dataset. These data-driven methods infer and assign the most commonly-used color to an object by learning from a huge set of samples. This is suitable for application scenarios with natural objects (which usually have few specific colors with small variations), but not applicable to man-made objects (of which the colors can be arbitrary). This poses further challenges in creative design tasks as global color compatibility must be considered.

Existing works in 3D scene colorization mainly focused on the interaction between light sources and object materials (including textures), and proposed various methods for manual [16] or automatic [13], [14], [15], [17] color assignment. However, a major difference when extending from 2D to 3D is that users may observe from different positions and directions in the 3D scene. The viewpoint variation may critically affect user perception and is under-explored in the corpus of existing works.

We present a creative color design framework for 3D interior scenes, called C^3 *Assignment* that automatically assigns color to objects (including light sources) in a 3D scene and provides interfaces for global setting adjustment (Fig. 1). Our method distinguishes itself from previous works by providing diverse and appropriate color suggestions for the whole scene and by dealing with indoor colorization in a 3D omnidirectional setting. We make the following technical contributions in this paper:

- A minimum-count-maximum-covering (MCMC) method to determine the camera placement. The selected viewpoints ensure the maximal observation

- *Juncong Lin, Pintong Xiao, Yinan Fu, Yubin Shi, Hongran Wang, and Shihui Guo are with the School of Informatics, Xiamen University, Xiamen, Fujian 361005, China. E-mail: {jclin, guoshihui}@xmu.edu.cn, xpt_solferino@qq.com, yinan.fu1202@gmail.com, tens444@163.com, 24320162202914@stu.xmu.edu.cn.*
- *Ying He is with the School of Computer Engineering, Nanyang Technological University, Singapore 639798, Singapore. E-mail: yhe@ntu.edu.sg.*
- *Tong-Yee Lee is with the Department of Computer Science and Information Engineering, National Cheng-Kung University, Tainan City 701, Taiwan. E-mail: tonylee@mail.ncku.edu.tw.*

Manuscript received 11 Dec. 2019; revised 18 Nov. 2020; accepted 23 Nov. 2020. Date of publication 1 Dec. 2020; date of current version 1 July 2022.
(Corresponding author: Shihui Guo.)
Recommended for acceptance by T. Ritschel.
Digital Object Identifier no. 10.1109/TVCG.2020.3041728

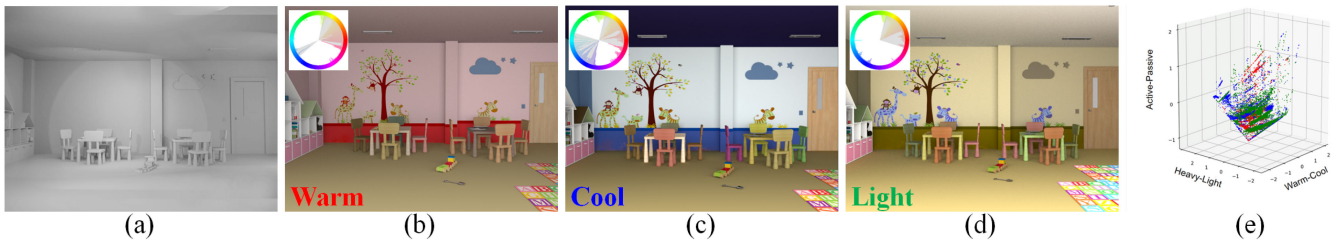


Fig. 1. Creative interior color design: (a) a 3D indoor scene; (b-d) harmonious (see hue template on the left top corner) colorization with different moods (indicated by the text on the left bottom corner); (e) pixel distribution of (b-d) in the mood space with colors corresponding to the text with the same color.

of the interior scene and minimal redundancy across all viewpoints. This reduces the computation load for latter procedures.

- A novel formulation of creative 3D interior color assignment that provides diverse suggestions by following certain guidelines relating to harmony, mood, contrast, etc. The formulation considers omnidirectional constraints and varied lighting conditions while ensuring comfortability of the proposed color configurations.
- An efficient evolutionary algorithm for the nonlinear and highly complex interior color optimization problem faced by interactive applications. A kriging model is used to approximate the time-consuming color harmony evaluation while maintaining the flexibility to adjust the objective function.

2 Related Work

2.1 3D Scene/Object Colorization

Leifman and Tal [16] proposed a scribble-based mesh colorization algorithm that can reasonably apply desired colors from scribbles to the whole mesh. There are also some works on material suggestion for 3D objects. Jain *et al.* [17] modeled the relationship between shape and material using a database of 3D objects with materials, while Nguyen *et al.* [18] transferred material style from a guiding source to a target 3D scene by formulating it as a combinatorial optimization of discrete materials assigned to discrete objects in the target. For indoor scenes, Chen *et al.* [13] introduced the magic decorator which can automatically generate material suggestions for 3D indoor scenes. Some other works [14], [15] automatically assigned harmonious colors to rooms and furniture according to models encoding the relationship between decorating styles and furniture colors. We also address the problem of indoor scene colorization with an emphasis on providing appropriate color configurations for creative interior design, and differ from existing works in not being limited by the most commonly used object colors [13], [14], [15]. In contrast, we adopt global constraints (harmony and mood) with more general local constraints (contrast and user preference) to provide more diverse yet appropriate suggestions, while maintaining effective computation.

2.2 Color Compatibility and Color Emotion

Among the large collection of theories on color compatibility, the idea of hue templates is probably the most popular one and has been widely used in art and design [19], [20], [21]. In particular, Matsuda's color harmony model [21] has been intensively used in the graphics and vision communities [22], [23], [24]. The theory of *hue templates* generalizes Goethe's theory by describing compatible colors as fixed rotations around the color wheel. This theory's main weakness is that the colors are defined independently of the underlying hue wheel [25]. Nemcsics [26] declaimed that colors were considered harmonious if one dimension of the space contrasts while the others remain fixed or they lie along lines. Although there are many theories on color compatibility, little consensus exists among them. Our work extends these theories to 3D by taking the Mastuda's model as an example and explores novel approaches to reduce the time cost of such an extension.

Another popular topic is the strong association between colors and emotions, which has been confirmed by psychological studies. Quantitative color mood models explicitly formulated mood scales from color appearance attributes like luminance, hue, and chroma [27], [28], [29]. Ou *et al.* [28] proposed a 3D color mood space which is consistent with findings in [27]. In a follow-up work [29], they identified three similar factors by principle component analysis (PCA) for color pairs, revealing a coherent color mood space for either single color or two-color combination. Wang *et al.* [30] adopted the theory for image color theme enhancement. We extend the theory into 3D with two or more colors and validate the effectiveness with experiments and user studies.

2.3 Camera Control in Virtual Environments

As a basic problem in computer graphics, 3D camera control has been addressed by a wide range of techniques in various contexts [31]. Automatic computation of single viewpoint was first addressed by Blinn [32], who presented an efficient technique to compute the position and orientation of a camera according to the specification of on-screen properties. It was then further generalized to consider various visual properties of the image space as constraints on the degrees of freedom of the camera [33]. Some researchers enforced elements of the filmic language onto the control of virtual cameras [34], while others used different camera representations, such as the Toric space, to simplify the expression and solving of the problem [35]. New challenges (such as collision, visibility and smoothness) emerge when computing sequences of viewpoints [36], [37]. Due to the popularity of quadrotors as new photography devices, the study of automatic camera control has increased in response to the need to create the desired sequences in possibly evolving environments [38], [39]. While most existing works focused on either determining camera paths in virtual/real scenes or taking high quality photos that follow certain

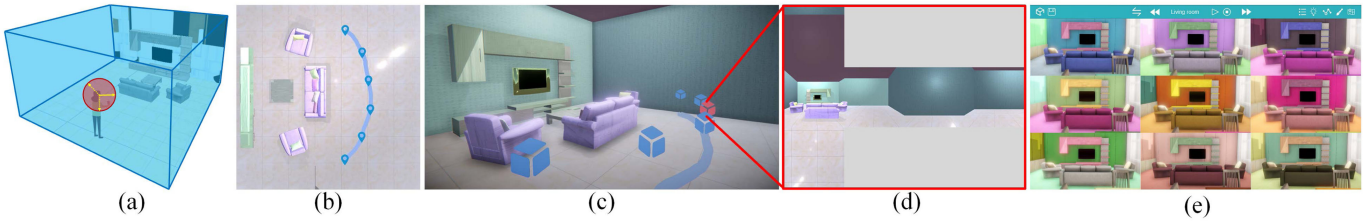


Fig. 2. Overview: (a) the integration domain for camera position (blue) and direction (red); (b) the generation of optimal viewpoints; (c-d) the generation of cubemap on each viewpoint; (e) the exploratory interface for colorization suggestions.

aesthetical rules, we have a different and more specific purpose. We aim to find the minimum number of representative viewpoints, with the maximum variations in the observed scenes, in the areas most frequently visited by inhabitants.

2.4 Surrogate-Assisted Evolutionary Algorithm

Evolutionary algorithms have been used in the computer graphics community but remain unpopular. A representative study by Sims [40] applied evolutionary algorithms with simple mutation and random crossover operations to generate images, textures, 3D plant structures and animations. Their work was then extended by themselves [41] and Pilat *et al.* [42] to generate creatures with various motions. Some other researchers investigated the usage of evolutionary algorithm on parameter optimization for procedural modeling [43]. Finally, evolutionary algorithm has also been used to evolve a shape collection and thus to extend the diversity of design gallery [44].

A major obstacle preventing the popularity of evolutionary algorithm could be the extremely high time cost. Optimization problem in the graphics community is commonly nonlinear and high dimensional, leading to intensive computation workload. Many researchers tried to conquer the obstacle by using approximate fitness evaluations [45]. However, the use of surrogate models may suffer from the so-called "curse of uncertainty" [46], i.e., the inaccuracies of the surrogates cause evolutionary algorithms to stall or converge to a false optimum.

In this paper, we design an efficient surrogate-assisted evolution framework that solves the expensive interior color optimization problem while satisfying the needs of interactive applications and avoiding the curse.

3 Overview

Our objective is to find color configurations for objects (including light sources) in a 3D indoor scene to satisfy certain indoor design constraints (harmony, mood, etc.), when inhabitants view from all possible positions and directions. Mathematically speaking, we describe the objective using a double integral

$$\arg\min_{\mathbf{v}} \int_{\Omega} \int_{\Theta} f(\theta, \mathbf{p}, \mathbf{v}) d\theta d\mathbf{p}, \tag{1}$$

where $\mathbf{v}$ is the color configuration of the objects in the scene, $\mathbf{p} \in \Omega$ is the camera position, $\theta \in \Theta$ is the viewing direction, and the evaluation function $f(\theta, \mathbf{p}, \mathbf{v})$ measures how well the color configuration meets those constraints under the current camera setting. The inner integral integrates over all possible viewing directions given the current camera position (the red sphere around the person in Fig. 2a), while the outer integral covers all possible camera positions in the indoor space (the blue-shadowed space in Fig. 2a).

As the evaluation function $f(\ldots)$ is usually rather complicated and non-integrable, directly solving for optimal color configurations through discretization of Ω and Θ is time-consuming if not impossible. We design a novel framework to cast it into a discrete optimization problem by first identifying representative camera positions in the scene (Fig. 2b) and then generating a cubemap of the current scene from each camera position (Fig. 2c). Finally, we sum up all the objective $f(\ldots)$ of each cubemap face in a weighted manner. Suggestions are generated through optimization of the objective function in an interactive evolution framework and presented to the user for exploration.

4 MCMC Camera Placement

To find the optimal camera placement, we first find the camera trajectories that the inhabitant would mostly follow, and then determine camera positions on these trajectories to capture the maximal variation of the scene view. Each position is also assigned a default viewing direction, which is set as the tangent of the trajectory.

4.1 Camera Trajectory Generation

We design a semi-automatic trajectory generation scheme. First, we generate an action map (Fig. 3a) of the current indoor scene with the method in [47]. We then convert the map into a binary image (Fig. 3b) and apply a thinning operation to get a skeleton (Fig. 3c) which is treated as the camera trajectory after pruning. The user can also manually specify the camera trajectory with sketches in the top view of the entire scene (Fig. 2b).

4.2 Optimal Camera Placement

The next step is to identify a set of optimal camera positions P_o on a trajectory C_i. P_o initially contains two ending points $\mathbf{p}_s$ and $\mathbf{p}_e$ of a trajectory by default. To better cover all possible viewpoints, we choose additional points for camera placement. Starting from either ending point, we move along the trajectory toward the other one. A candidate position $\mathbf{p}_c$ is added to P_o if the projected area (S_c^k) of each object (indexed by k) on the current camera cubemap varies significantly from the existing camera positions $\mathbf{p}_i$ in P_o. This is mathematically formulated as

$$m_c = \frac{\sum_k [S_c^k - S_i^k]}{S} > \epsilon, \forall \mathbf{p}_i \in P_o. \tag{2}$$

Fig. 3. Automatic generation of camera trajectory: (a) action map; (b) binarized action map; (c) skeleton of action map.

Here, S is the area of the whole cubemap, and ϵ is the threshold for the total variation of the scene view between the two camera positions. ϵ can be specified directly by the user or estimated from the division of m_c between the two ending viewpoints of the trajectory and an expected viewpoint number n_v specified by the user. This strategy minimizes the viewpoint redundancy when constructing P_o. To calculate the projected area of an object, we turn the light in the scene off and set the materials of all objects as white and the material of the specified object as red. The projected area is the sum of the red pixels.

Fig. 4 illustrates the trajectories and viewpoints' cubemap images of MCMC and uniform placement for a qualitative comparison. The corresponding quantitative comparison between the two methods is summarized in Table 1. The statistics show that the metric m_c between all the neighboring viewpoint pairs on the trajectory keeps steady in the MCMC placement while there are significant fluctuations in the uniform method. A large m_c indicates that important objects may be missed from the scene, while a small m_c indicates unnecessary viewpoints.

TABLE 1
Quantitative Comparison of m_c

	(b-c)*	(c-d)	(d-e)	(e-f)
MCMC	3.87%	3.94%	4.01%	3.90%
Uniform	3.60%	3.50%	0.82%	3.13%

** 'b' and 'c' refer to (b) and (c) in Fig. 4, similar for others.*

5 Omnidirectional 3D Interior Color Design

5.1 Objective Function

Our purpose is to find appropriate color configurations for objects and light sources in the 3D scene to achieve harmony under various constraints. The color configuration of an object O_k is represented by a vector $\mathbf{v}_k$ containing tuples of RGB: $\mathbf{v}_k = (\mathbf{v}_{k1}, \mathbf{v}_{k2}, \ldots, \mathbf{v}_{kj})$. Each object's multiple RGB tuples correspond to different object components and textures (indexed by j) assigned to it. Also, different objects may share the same RGB tuples due to various constraints.

We pack all RGB tuples $\mathbf{v}_k$s into a long vector $\mathbf{v} = (\mathbf{v}_1, \mathbf{v}_2, \ldots \mathbf{v}_k)$ for the whole scene. The objective function is defined as the weighted sum of four terms (the harmony energy E_h, the mood energy E_m, the contrast energy E_c and the anchor energy E_a) to enforce plausible spatial arrangements of colors and overall color compatibility

$$E(\mathbf{v}) = w_h E_h(\mathbf{v}) + w_m E_m(\mathbf{v}) + w_c E_c(\mathbf{v}) + w_a E_a(\mathbf{v}). \tag{3}$$

For coefficients in the equation, we set: $w_h = 100$, a large number to enforce harmony; $w_a = 100$, also a large number to ensure the consistency with user preference; and $w_m = 5$, $w_c = 1$ by default so the user can fine tune for their specific requirements.

The definitions of E_h, E_m and E_c are given in Sections 5.2, 5.3 and 5.4 respectively. The anchor energy E_a is used to ensure that a given object O_k is assigned a color similar to the one $\mathbf{v}_k^a$ prescribed by the user and is simply defined in the following form:

$$E_a = \sum_k \| \mathbf{v}_k - \mathbf{v}_k^a \|^2 / 3. \tag{4}$$

The anchor energy can also be treated as a hard constraint by removing the color component of the specified object from $\mathbf{v}$ directly.

5.2 Color Harmony Evaluation

As mentioned in Section 2.2, there is no unified model of color harmony or color compatibility. The existing models are mainly for 2D space and have little consensus. We have no specific preference of one model over the others as the goal of our work is to explore 3D color harmony, which has so far been studied by very few people. To be specific, we explore the effectiveness of these schemes in 3D. We extend the idea of hue templates [21], [23] to 3D by evaluating the

Fig. 4. MCMC (top) versus uniform (bottom) camera placements: the different viewpoint distributions on the given trajectory are given in (a); projected scene images on the main faces (the side toward TV in the scene) of each viewpoint's cubemaps are shown in (c-e), (b) and (f) are shared by the two placements.

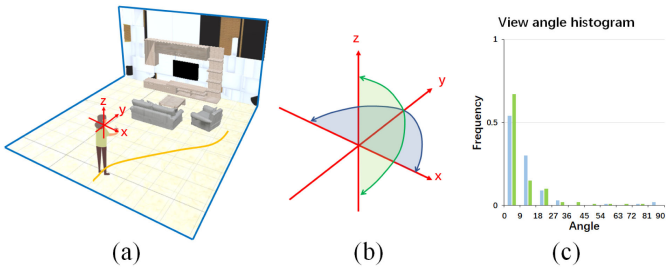


Fig. 5. Pilot study of user's navigation behavior: (a) the reference coordinate is defined by the up (z-axis), side (x-axis) and front (y-axis) directions. The y axis is the tangent vector of trajectory; (b) the viewpoint deviations from the front direction in the x-y plane (blue) and y-z plane (green) are observed; (c) histograms of deviation angles in the two planes.

color harmony on each face of the cubemap and summing them in a weighted manner

$$E_h(\mathbf{v}) = \sum_i w_i \sum_{f=1}^{6} w_f E_{ht}(f_{i,f}(\mathbf{v})), \tag{5}$$

where w_i is the weight for each camera cubemap M_i and w_f is the weight for each face $f_{i,f}(\mathbf{v})$. $w_i = 1$ by default when the camera trajectory is manually sketched, and it is set as the possibility of inhabitants observing from the camera position when the trajectory is automatically extracted from the action map. w_f is set to 0.1 for the top and bottom faces of the cubemap and 0.2 for the side faces. A pilot study was conducted to illustrate the suitability of the weight values. Using a VR helmet, we observed the head movements of participants as they navigated the scene and recorded how they rotated around the up and side directions (Fig. 5a). The histograms of the rotating angles around the two directions are plotted in Fig. 5c. The result shows that participants tended to keep straight ahead, indicating the importance of the front face. We also provide another mode to automatically ignore the cubemap faces with minor importance according to the pilot study. Computation redundancy in the optimization process is reduced by about 30 percent when ignoring the top, bottom, and back faces.

The harmony energy for the face is defined similarly to [22] but with normalization

$$E_{ht}(f_{i,f}(\mathbf{v})) = \arg\min_{m,\alpha} \sum_p \parallel H(p) - E_{T_m(\alpha)}(p) \parallel \cdot \frac{S(p)}{180},$$

where H and S denote the hue and saturation channels respectively, p is a pixel on the face image $f_{i,f}(\mathbf{v})$; the hue distance $\parallel \cdot \parallel$ refers to the arc-length distance on the hue wheel (measured in radians). T_m represents the mth template in Matsuda's scheme [21], and $E_{T_m(\alpha)}(p)$ is the sector border hue of T_m with orientation α that is closest to the hue of the pixel p [22]. Hues that reside inside the sectors of T_m are considered to have zero distance from the template.

5.3 Color Mood Evaluation

Colors and their combinations can evoke various emotional feelings. For example, the combination of pink and purple colors is used to convey "gracefulness", while the combination of blue and grey colors can make people feel "sad". Such effects are especially important in interior color design.

We use Ou *et al.*'s model [28], [29] to quantitatively associate color with emotion. They applied principle component analysis and constructed a 3D color mood space of activity (active-passive), weight (heavy-light), and heat (warm-cool). They used the following functions to transfer colors from the CIELab space $\mathbf{\Phi} \subset \mathbb{R}^3$ to the mood space $\mathbf{\Psi} \subset \mathbb{R}^3$

$$\begin{aligned}
\boldsymbol{\psi}_0 &= \alpha_1 + \alpha_2 \left[(\boldsymbol{\phi}_0 - \alpha_3)^2 + (\boldsymbol{\phi}_1 - \alpha_4)^2 + \left(\frac{\boldsymbol{\phi}_2 - \alpha_5}{\alpha_6} \right)^2 \right]^{1/2}, \\
\boldsymbol{\psi}_1 &= \omega_1 + \omega_2(\omega_3 - \boldsymbol{\phi}_0) + \omega_4 \cos(h - \omega_5), \\
\boldsymbol{\psi}_2 &= \tau_1 + \tau_2 (C)^{\tau_3} \cos(h - \tau_4),
\end{aligned}$$

where $\boldsymbol{\phi} \in \mathbf{\Phi}$ with its three components are the CIELab lightness coordinates; C is the CIELab chroma; h is the CIELab hue angle; and $\boldsymbol{\psi} \in \mathbf{\Psi}$ with its three components represents activity, weight, and heat, respectively. Table 2 lists the specific values of the coefficients (referenced from [28]).

The additive relationship of mood evocation was validated for two-color combinations in [29] and further generalized to more than two colors in [30], but without a formal validation. We use a similar formulation to evaluate the mood $\boldsymbol{\psi}_M$ of a cubemap M (or each of its face) by averaging the mood of each pixel. The mood energy E_m for the cubemap (or one of its faces) is defined by summing the energy of the three components E_m^i

$$E_m = \sum_{i=0}^{2} E_m^i, \tag{6}$$

where E_m^i is defined according to the user preference as below:

$$E_m^i = \begin{cases} (2 - \boldsymbol{\psi}_i)^2/16, & \text{for } \textit{active, heavy and warm} \\ (\boldsymbol{\psi}_i + 2)^2/16, & \text{for } \textit{passive, light and cool.} \end{cases}$$

5.4 Color Contrast Evaluation

While the harmony and mood terms can help enforce global consistency between colors, they cannot provide good scene colorization. Contrast is an important factor in visual perception [48], and areas with high contrast are more likely to attract human attention and help distinguish scene components [49]. There are many different metrics for defining

TABLE 2
Coefficients of the Mood Space

Mood dimension	activity						weight					heat			
Parameter	α_1	α_2	α_3	α_4	α_5	α_6	ω_1	ω_2	ω_3	ω_4	ω_5	τ_1	τ_2	τ_3	τ_4
Value	-2.1	0.06	50	3	17	1.4	-1.8	0.04	100	0.45	100	-0.5	0.02	1.07	50

Fig. 6. Texture color assignment.

contrast (see [50]). Here, we use global luminance and define the contrast energy as

$$E_c = \sum_{m,n} w_{mn}(1.0 - (\parallel L_m - L_n \parallel /100)^2), \tag{7}$$

where L_m and L_n are the L channels in the CIE Lab color space of two scene objects O_m and O_n. w_{mn} is the contact area of the two neighboring objects, O_m and O_n.

5.5 Rendering Issues

We only consider the diffusion components of scene objects in the optimization process as they have a dominant effect on scene colorization. Texture colors are also considered and can be smoothly altered with existing photo recoloring methods [51] (see Fig. 6 for an illustration). Applying these methods in each evaluation of the objective function can lead to an extremely time-consuming optimization process (See Section 6 for details). Therefore, we use a simplified method during the optimization process and apply recoloring methods at the end. To be specific, we first extract a palette with size k ($k = 5$ by default but can be specified by the user) using the method in [51] and then assign each pixel to the nearby palette entry. During optimization, we alter the palette entries' colors, and the pixels will also be changed to the same color as the entry they belong to.

The choice of renderer greatly affects the efficiency and effectiveness of the optimization process. Highly realistic renderers are usually very complex and time-consuming, making it difficult to achieve interactive rate in the design process, while simple renderers may lead to deviations from the real case in the rendering result, making the optimal material configuration not leading to harmony design in the final result. In this paper, we make a careful balance by using a simple OpenGL shader with illumination mapping. Experiments have shown the effectiveness of this strategy in general cases (see Section 7 and Fig. 11).

6 Surrogate-Assisted Genetic Algorithm for Color Optimization

Considering the non-linearity and high complexity of the objective function (Equation (3)), stochastic optimization algorithms like genetic algorithm or simulated annealing are more suitable for the problem. Although extensive efforts have been made to reduce the time cost for evaluation of the objective function (GPU acceleration, simplification of texture color transfer and renderer), the optimization process is still time consuming. The main reason is the highly frequent evaluation of the cost function. To accelerate this process, we propose a surrogate-assisted genetic algorithm for color optimization.

Algorithm 1. Surrogate-Assisted Genetic Algorithm

Input: population size N_p, sampling space size N_s of surrogate model, elitism size $N_e = 20\%N_p$
Output: optimal color configuration vector of the scene $\mathbf{x}_{opt}$

1 bool isSurrogateAvailable=false;
2 $P_0 = \emptyset$;
3 $P_1 = \emptyset$;
4 **for** $i \in \{1 \dots N_p\}$ **do**
5 append(P_0,newIndividual());
6 float **f**[N];
7 calcFitnessWithObjectiveFunction(P_0, **f**);
8 append(S,P_0, **f**);
9 **if** ($|S| \geq N_s$) **then**
10 buildSurrogateModel(S); isSurrogateAvailable=true;
11 **while** *termination criterion is not met* **do**
12 sort(P_0, **f**);
13 **for** $i \in \{1 \dots N_e\}$ **do**
14 append(P_1, $P_0[i]$, **f**[i]);
15 eliteFiltering(P_1);
16 **for** $i \in \{1 \dots (N_p - N_e/4)/2\}$ **do**
17 $\mathbf{x}_1$=select(P_0, **f**);
18 $\mathbf{x}_2$=select(P_0, **f**);
19 **if** $rand() \leq mutationProbability$ **then**
20 $\mathbf{x}_1^c$=mutate($\mathbf{x}_1$);
21 $\mathbf{x}_2^c$=mutate($\mathbf{x}_2$);
22 **else**
23 crossover($\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_1^c$, $\mathbf{x}_2^c$);
24 float f_1, f_2;
25 **if** *isSurrogateAvailable* **then**
26 f_1=calcFitnessWithSurrogateModel($\mathbf{x}_1^c$);
27 f_2=calcFitnessWithSurrogateModel($\mathbf{x}_2^c$);
28 **else**
29 f_1=calcFitnessWithObjectiveFunction($\mathbf{x}_1^c$);
30 f_2=calcFitnessWithObjectiveFunction($\mathbf{x}_2^c$);
31 append(S, $\mathbf{x}_1^c$, f_1);
32 append(S, $\mathbf{x}_2^c$, f_2);
33 **if** ($|S| \geq N_s$) **then**
34 buildSurrogateModel(S); isSurrogateAvailable=true;
35 append(P_1, $\mathbf{x}_1^c$, f_1);
36 append(P_1, $\mathbf{x}_2^c$, f_2);
37 $P_0 = P_1$;
38 $P_1 = \emptyset$;

6.1 Framework

A brief outline of the surrogate-assisted genetic algorithm is presented in Algorithm 6, and the differences between the proposed algorithm and a traditional genetic algorithm are highlighted in bold. The algorithm begins by initializing a population of design points. During the database building phase, the algorithm operates like a traditional genetic algorithm and accurately evaluates the standard fitness function until enough sample points are collected to build the surrogate model (Algorithm 6 line 7-8). Subsequently, the algorithm proceeds to build the surrogate model and uses it in the following evolution.

Population Initialization. We use the spoke-dart-based high-dimensional blue-noise sampling method [52] to generate an initial population, which covers the search space to the fullest extent.

Elitism. During each iteration, we select the top 20 percent of the whole population as elites according to their fitness. These elites will be further filtered out in the steps below.

Curse of Uncertainty. Selected individuals are resorted according to the uncertainty estimation (see the end of Section 6.2) in an ascending order, and the top 50 percent are reserved.

Diversity. To maintain the diversity of evolution, we select 50 percent of the reserved individuals with the maximum dissimilarity from each other. The dissimilarity of two individuals (corresponding to two color assignments of the scene) is defined as the area-weighted sum of the L*a*b* distances between the same object's two different color assignments

$$\sum_i A_i \|\mathbf{c}_i - \tilde{\mathbf{c}}_i\|,$$

where A_i is the summed area of all triangles on an object, $\mathbf{c}_i$ and $\tilde{\mathbf{c}}_i$ are the object's colors in the L*a*b* space.

6.2 Surrogate Model

We choose to use the kriging model [53] as the surrogate model. A kriging model predicts the response value at an un-sampled point $\mathbf{x}$ as the sum of a global trend function $\mathbf{f}^T(\mathbf{x})\boldsymbol{\beta}$ and Gaussian process $G(\mathbf{x})$

$$y(\mathbf{x}) = \mathbf{f}^T(\mathbf{x})\boldsymbol{\beta} + G(\mathbf{x}), \mathbf{x} \in R^m, \tag{8}$$

where $\mathbf{f}(\mathbf{x}) = [f_0(\mathbf{x}), \ldots, f_{p-1}(\mathbf{x})]^T \in R^p$ contains a set of the regression basis functions and $\boldsymbol{\beta} = [\beta_0, \ldots, \beta_{p-1}] \in R^p$ refers to the corresponding coefficients. $\mathbf{f}^T(\mathbf{x})\boldsymbol{\beta}$ is generally defined as either a constant or low-order polynomials. The random process $G(\mathbf{x})$ is assumed to be a Gaussian process with zero mean, variance σ^2 and nonzero covariance

$$Cov[G(\mathbf{x}), G(\mathbf{x}')] = \sigma^2 R(\mathbf{x}, \mathbf{x}').$$

The correlation function $R(\mathbf{x}, \mathbf{x}')$ depends on the euclidean distance between any two $\mathbf{x}$ and $\mathbf{x}'$ only. In this paper, a Gaussian exponential correlation function is adopted with the following form:

$$R(\mathbf{x}, \mathbf{x}') = exp\left[-\sum_{k=1}^{m} \theta_k |x_k - x'k|^{p_k}\right], 1 < p_k \leq 2,$$

where $\boldsymbol{\theta} = [\theta_1, \theta_2, \ldots, \theta_m]^T$ and $\mathbf{p} = [p_1, p_2, \ldots p_m]^T$ denote the vectors of the unknown model parameters (hyper parameters) to be tuned.

According to the derivation by Sacks *et al.* [54], the kriging predictor $\hat{y}(\mathbf{x})$ for a set of N sample points $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ and their corresponding responses $\mathbf{y}_s = \{y_1, \ldots, y_N\}$ can be written as

$$\hat{y}(\mathbf{x}) = \mathbf{f}^T(\mathbf{x})\boldsymbol{\beta}^* + \mathbf{r}^T(\mathbf{x})R^{-1}(\mathbf{y}_s - \mathbf{F}\boldsymbol{\beta}^*),$$

where $\mathbf{r} = \{R(\mathbf{x}, \mathbf{x}_1), \ldots, R(\mathbf{x}, \mathbf{x}_N)\}^T$ denotes the correlation vector, and the generalized least square estimation of $\boldsymbol{\beta}^*$ is

$$\boldsymbol{\beta}^* = (\mathbf{F}^T\mathbf{R}^{-1}\mathbf{F})^{-1}\mathbf{F}^T\mathbf{R}^{-1}\mathbf{y}_s,$$

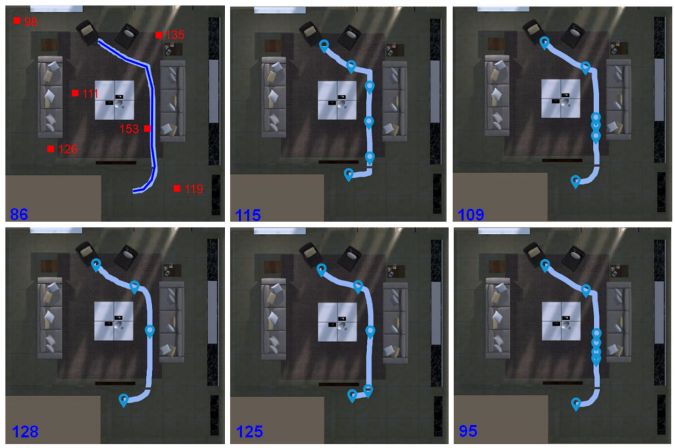


Fig. 7. Effect of camera placement on scene color optimization: the top-left image shows objective function values (red) on several camera positions (red square) and the averaged value (blue) on the blue trajectory by dense sampling; the top-middle image shows the averaged objective function value on the 6 uniformly sampled camera position, while the other images show the averaged objective function values on 4, 5, 6, 7 camera positions sampled with the MCMC method.

where $\mathbf{F}$ is an $N \times p$ matrix with $F_{ij} = f_j(\mathbf{x}_i)$, $\mathbf{R}$ is an $N \times N$ correlation matrix with $R_{ij} = R(\mathbf{x}_i, \mathbf{x}_j)$.

A major benefit of the kriging model is the uncertainty estimation for prediction, which uses the following form of mean square error

$$\sigma^2(\mathbf{x}) = \sigma^2\{1 - \mathbf{r}(\mathbf{x})\mathbf{R}^{-1}\mathbf{r}(\mathbf{x}) + [\mathbf{F}^T\mathbf{R}^{-1}\mathbf{r}(\mathbf{x}) - \mathbf{f}(\mathbf{x})]^T(\mathbf{F}^T)[\mathbf{F}^T\mathbf{R}^{-1}\mathbf{r}(\mathbf{x}) - \mathbf{f}(\mathbf{x})]\}.$$

The uncertainty estimation provides an important cue to address the "curse of uncertainty".

7 Experimental Results

We have developed a prototype system in Unity with C#, using the DACE toolbox [55] for the kriging model. All experiments run on a workstation with an AMD Ryzen Threadriper 1920× 12 Core CPU, 64G RAM and a Geforce 1080Ti GPU.

We investigated how different camera placements can affect the color optimization process by observing their different objective function values under the same color assignment. As we can see from Fig. 7, the objective function value (109) of the MCMC method is closer to that of a densely sampled trajectory (to approximate line integral on the trajectory) in the top-left image (86) when compared to that of uniform sampling (115). Fewer camera placements are needed to achieve the same accuracy as uniform placement according to the results shown in the bottom row of the figure, indicating reduced computation redundancy. We can also notice the approximation to dense sampling with the number of camera samples increased in the MCMC method.

The accurate fitness evaluation and the kriging-approximated fitness evaluation both lead to consistent convergence result of the evolution process, as shown in Fig. 8a. However, the time costs for these two methods differ significantly: 15 minutes for the accurate evaluation case, and 2-3 seconds for the approximation case. Fig. 8b demonstrates the reliability of the uncertainty measure provided by the

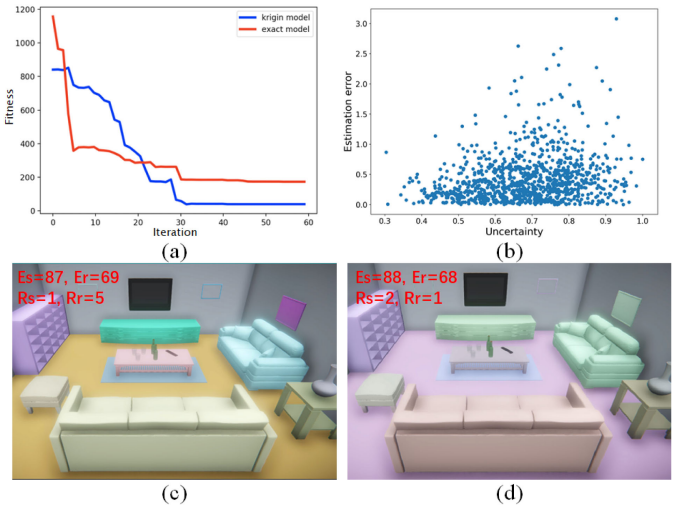


Fig. 8. Effectiveness of kriging approximation: (a) convergence of evolutionary algorithm with approximate and accurate fitness evaluations; (b) estimation error versus uncertainty measure for estimated individuals; (c) optimal colorization with the surrogate fitness rank R_s=1 corresponds while the real fitness rank R_r=5; (d) optimal colorization with the real fitness rank R_r=1 corresponds while the surrogate fitness rank R_s=2. All ranks are in the ascending order.

TABLE 3
Performance Statistics

Scene	# objs	Pre-computation (s)	Evolution (s)	# iters
Fig. 1	59	55.1	1.76	29
Fig. 11	72	44	1.5	35
Fig. 12 (row 2)	54	38	2.1	30
Fig. 12 (row 3)	492	49	2.7	55
Fig. 15	94	52	2.3	52
Fig. 16	29	50	1.24	45
Fig. 23	703	55	2.3	45

kirging model. The elitism filtering towards uncertainty in Section 6.1 reserves individuals with good estimated fitness and thus guarantees the plausibility of the surrogate-assisted evolution framework. Finally, Figs. 8c and 8d show a high consistency for colorization between the approximate and accurate modes as the rankings of colorizations in the counterpart are close.

Table 3 reports the timing statistics for scene colorization referred in the paper. It generally costs 2-3 seconds to provide suggestions for a scene. The duration of the optimization process is independent of the scene complexity. To validate this, we constructed five interior scenes with the number of objects ranging from 10 to 50, and then colorized the scene with our method. We can see from Fig. 9 that the time cost for fitness evaluation is close to stable.

We further investigated the effect using different cubemap resolutions. Our current implementation runs on GPU, therefore the process duration is almost linearly proportional to the cubemap resolution (please refer to the top left image in Fig. 10).

We also investigated the effect of different renderers on the calculation of the objective function. As we can see from Fig. 11, the color histogram on hue wheel looks highly similar and the distributions of the image pixels in the mood space are also highly overlapped. Therefore, the type of

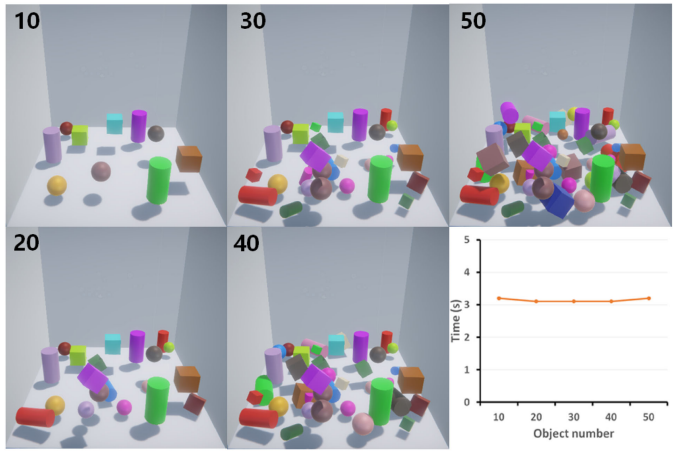


Fig. 9. Performance statistics under different levels of scene complexity.

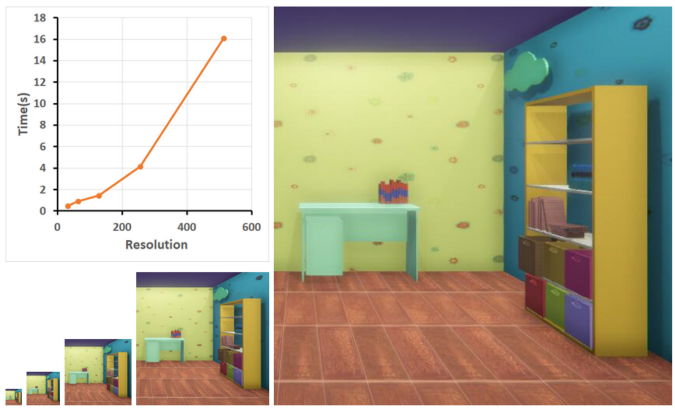


Fig. 10. Performance statistics under different cubemap resolutions.

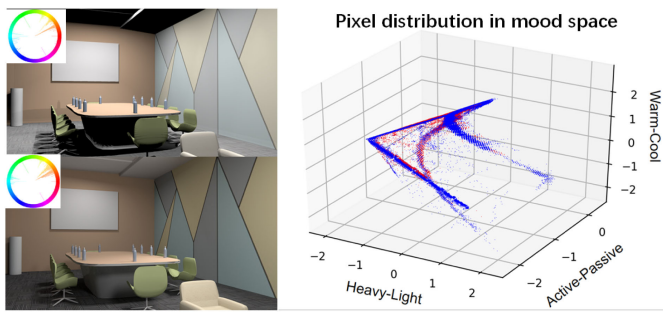


Fig. 11. Influence of using different renderers. The left side shows the rendered results with OpenGL in Maya (top) and Arnold in Maya (bottom) respectively with the corresponding hue histogram, while the right side shows pixel distributions in the mood space.

renderer has limited influence on the evaluation of harmony and mood. This implies the distinct use of the fast OpenGL renderer in the optimization process and the high-quality Arnold renderer to output the final results.

Fig. 12 shows a comparison with the related works on diversity. As we can see from the first row, the colorizations of [13], [15] are highly similar in appearance across different scenes (dark red cabinets, brown floor, white quilts, etc.) as they are learned from sample images of daily scenarios, whereas our system generates diverse yet promising colorizations (the second row of Fig. 12). Even when a specific theme is given, various colorizations can be generated, as

Fig. 12. Comparison with existing works: blue squared images are from [13], red squared images are from [15] while green squared images are generated by our system.

Fig. 13. Subjective comparison: the second row shows suggestions presented by our system while the third row shows those presented by MagicDecorator.

shown in the third row of Fig. 12 where the results are generated under a theme similar to [15]. In summary, our method performs better in terms of colorization diversity.

We also conducted a user study to compare our system more accurately with MagicDecorator [13].[1] A set of suggestions generated by both systems were presented to the participant, who was then asked to rate them according to three different aspects: reasonability, inspiration, and overall preference. According to the results presented in Fig. 13, our system scored better than MagicDecorator for inspiration and overall preference but slightly worse for reasonability. The system can also optimize lighting properties to generate optimal colorization, as shown in Fig. 14, which has seldom been explored in previous works.

The effects of grouping constraints and energy terms in Equation (3) are presented in Fig. 15. As we can see from Fig. 15a, sofas were assigned with the same color with the grouping constraint (top), otherwise different colors (bottom). To better investigate the effect of harmony energy, we turned off the grouping constraint for sofas (the grouping constraint for cabinets was kept throughout the whole process). The

1. Results are provided by the authors

Fig. 14. Colorization with different lighting conditions.

removal of the harmony energy term leads to a more diverse coloring of the sofas (bottom) and a more diverse histogram distribution (see the right side of the images in Fig. 15b). To better investigate the effect of the last three energy terms, we temporarily set the weight to 100 for the corresponding term, with the others set as 1 (except the harmony term, which is always set as 100). Fig. 15c shows the comparison between the two opposing moods: heavy (top) and light (bottom). From the corresponding histogram on the right side, we can observe a clear distribution of pixels towards the corresponding poles. We can also see a sharper color contrast between sofa and floor in Fig. 15d when contrast is enhanced (top). In Fig. 15e, it can be seen that the sofa is colorized similar to the specified color (inset in the top left of each image) when user preference is enforced.

8 User Study

8.1 Preparation

We recruited 17 participants (male: 12, female: 5) aged 19 to 30 years. We generated 200 different color assignment results for a typical living room and selected 5 with uniformly distributed metrics in each dimension (harmony-disharmony, warm-cool, positive-negative, heavy-light). The other metrics were kept the same to reduce their influence. Fig. 16 shows the chosen schemes. We described the task at each stage and gave the participants time to familiarize themselves with it.

A set of example images[2] were prepared to show participants what is harmony and color mood (See Fig. 17).

We provided each participant with a VR headset so that they could immerse themselves and navigate in the indoor scene (Figs. 18a and 18b). Their traces during navigation were recorded and visualized as a heat map (Fig. 18c). This map was then used to train the action map generation model [47] to infer optimal camera positions in a different yet similar indoor scene.

8.2 Validation of 3D Harmony

In this stage, we first informed each participant about harmony and disharmony by introducing existing color harmony models and showing them some opposing pairs of images (Fig. 17a).

Two colorization schemes for the testing scene were randomly chosen from those in the first row of Fig. 16 for each

2. Collected from [22] and the following websites:
http://www.verydesigner.cn
https://articulo.mercadolibre.com.ar
https://zhuanlan.zhihu.com/p/61712002
https://freshome.com and https://freshmedia.bg/

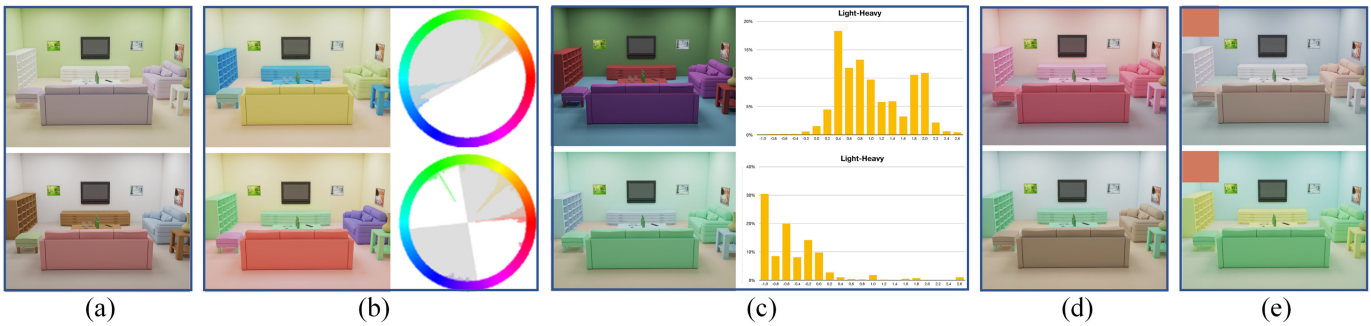


Fig. 15. Effects of grouping constraint and energy terms: (a) grouping; (b) harmony; (c) mood; (d) contrast; (e) preference.

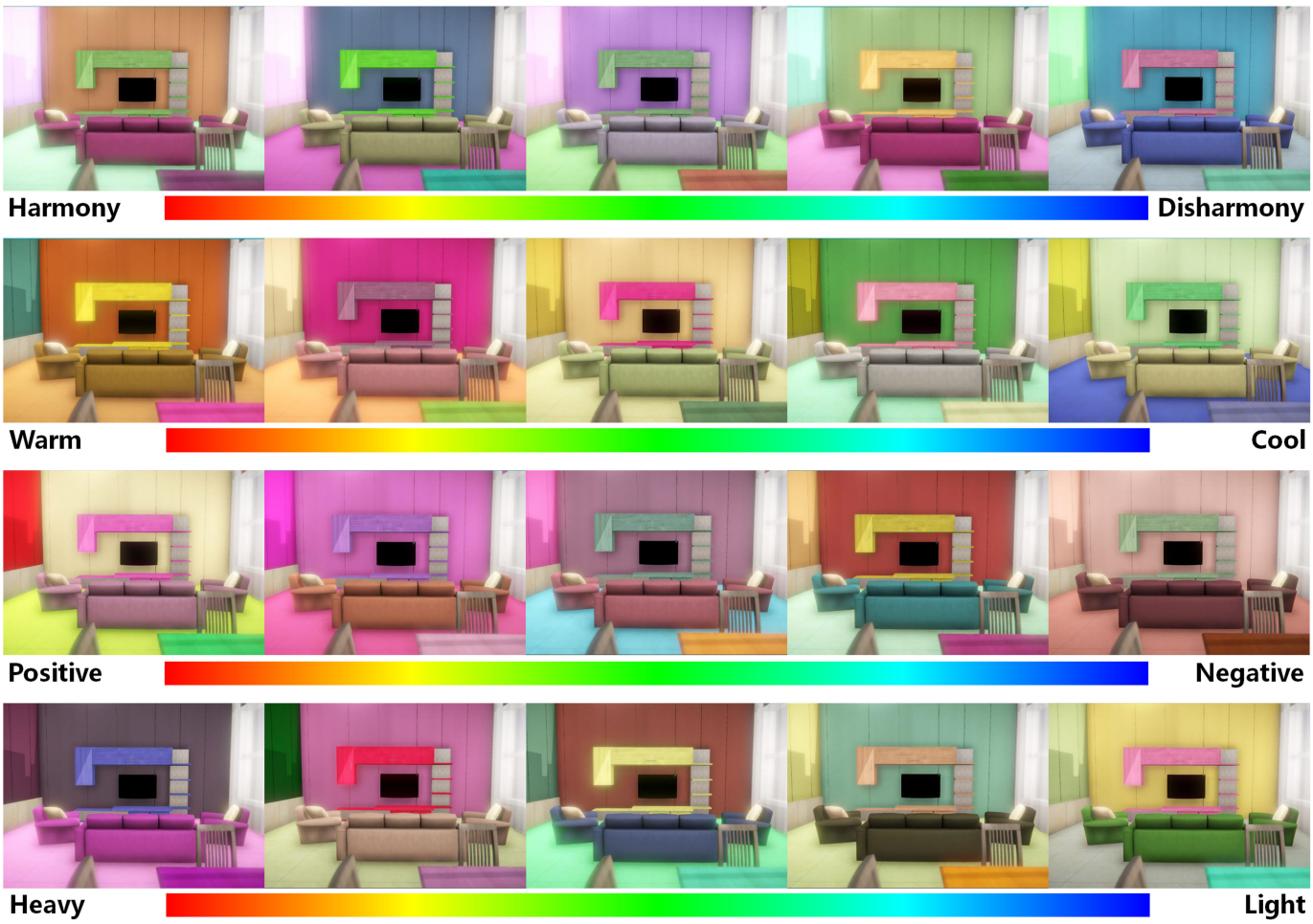


Fig. 16. Scoring scenes. Each row shows five colorization results with uniformly distributed attribute values in the corresponding dimension while other attributes are the same.

Fig. 17. Opposing image pairs shown to the participant: (a) harmony (left) versus disharmony (right); (b) warm (left) versus cool (right); (c) positive (left) versus negative (right); (d) heavy (left) versus light (right).

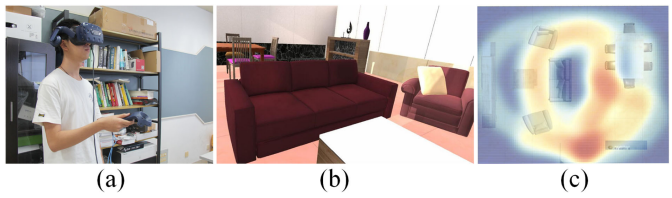


Fig. 18. User study setup: a participant wore a VR headset (a) and navigated in the colorized scene (b) with his action map recorded (c).

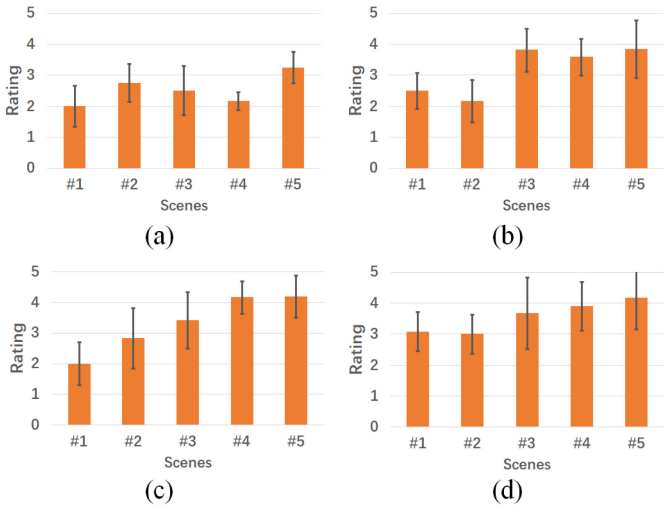


Fig. 19. Scene rating: (a) harmony versus disharmony; (b) warm versus cool; (c) positive versus negative; (d) heavy-light.

participant. The participants were then asked to put on the VR headset and were given 20 seconds to navigate the indoor scene per colorization scheme. They were then asked to point out which scheme had more harmony and score the harmony of each scheme using a five-point Likert scale, with 5 indicating harmony and 1 indicating disharmony. Fig. 19a shows the rating statistics.

8.3 Validation of 3D Mood

Similar to the previous stage, we first informed each participant about color mood by introducing them to the existing color mood models and showing them some opposing pairs of images on the three dimensions of the mood model (warm-cool, positive-negative and heavy-light in Figs. 17b, 17c, and 17d). Two colorization schemes for each dimension were randomly chosen from those in Fig. 19 for each participant. The participants were then told to put on the VR headset and given 20 seconds to navigate the indoor scene per colorization scheme. They were then asked to score the warmth (positiveness/heaviness) of each scheme using a five-point Likert scale, with 5 indicating warm (positive/heavy) and 1 indicating cool (negative/light). Figs. 19b, 19c, and 19d show the rating statistics.

8.4 Validation of System Usability

The purpose of this stage is to investigate how the system could help people with creative interior color design. The participants were asked to colorize a given interior scene with our system and with the standard Unity system. The colorization interfaces of two systems are shown in Fig. 20. After introduction of the two interfaces, each participant

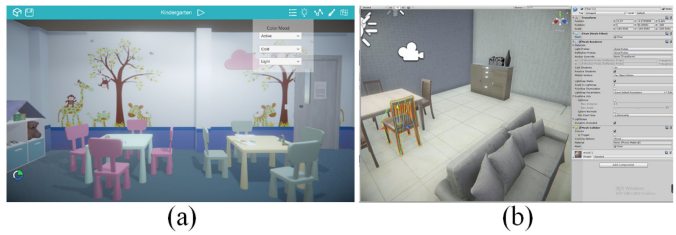


Fig. 20. Interfaces for interior colorization: (a) our system; (b) the standard Unity.

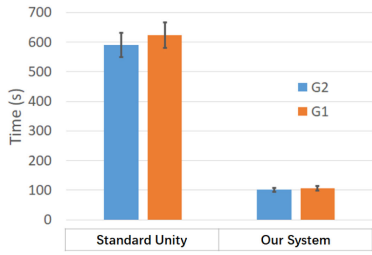


Fig. 21. Usability timing statistics. The vertical axis shows the average time to complete each experiment.

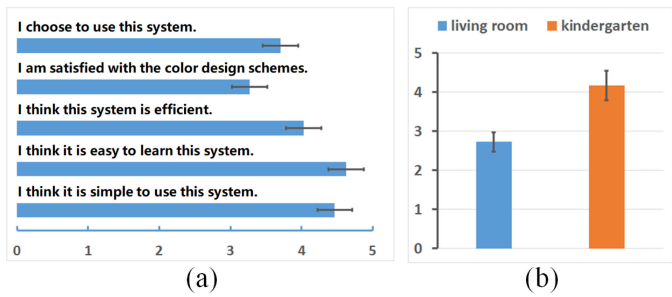


Fig. 22. Questionnaire. (a) Overall rating. (b) Separate ratings for different scenes on whether the participant was satisfied with the results.

was given 5 minutes to familiarize with the interfaces before performing the task.

The duration of the colorization process was recorded (Fig. 21). Participants were divided into two groups: G1 and G2. G1 used our system first and then the standard Unity system while G2 followed the reverse order. The results show that participants were 6 times faster when using our system compared to the alternative choice.

After finishing the tasks, each participant completed a questionnaire about their preference towards our interface by rating its different aspects using a 5 point Likert scale, with 5 indicating that they strongly agree and 1 indicating that they strongly disagree. Fig. 22 presents the rating statistics.

Some participants pointed out that some of the colorization schemes generated by our system were too colorful and more suitable for public interior scenes (such as bars or shopping malls) rather than home interiors. Therefore, we added a scene and recruited new participants. We separated the ratings for the two scenes on Q5 for a comparison, which is presented in Fig. 22b. The score for "kindergarten" increased to about 4.2 while the score for "living room" dropped to around 2.8, which made sense as kindergartens tend to be colorful. In general, our method is more suitable for places like kindergartens, bars, shopping malls, and other commercial establishments where a rich variety of

Fig. 23. An unappealing colorization with well optimized objectives.

colors is welcomed. Some colorization results (Fig. 23) may not look so appealing especially in heavy mood, although the objective function is well optimized. This may partially explain the fluctuations in Fig. 19a. The issue was partially solved by presenting multiple suggestions to the participants after evolution. Potential solutions include adding constraints into evolution process and allowing the participants to interactively adjust the angles of the harmony hue templates to avoid unappealing color combinations.

9 Conclusion

This paper proposes a novel design method to provide diverse color suggestions for a 3D interior scene when observed from all possible positions and directions. Our system starts with the (semi-)automatic specification of multiple viewpoints in the scene and then looks for optimal color assignments for the objects using a surrogate-assisted algorithm. The viewpoints are selected to be on the trajectory that an inhabitant most possibly passes by and to cover a maximum variation of the scene view. General global and local rules, including harmony, mood, contrast and user preference, are considered when finding the optimal color assignments. Thanks to the introduction of the kriging model as a surrogate, the optimization problem can be solved fast enough to satisfy the needs of interactive applications. The diversity of the suggestions is preserved and the curse of uncertainty is alleviated by the carefully designed evolution process. Experiments and user studies have been conducted to validate various aspects of the system. Compared with existing works, our method provides more diverse colorization results, making it more suitable for creative interior design.

In the future, we would like to explore the following areas to further improve the system:

- Global user preference. Although we allow the user to specify his preferred colors for certain objects (local preference), the overall colorization style (global preference) is commonly expected. However, such global preference is quite abstract and the user usually only has a vague demand at the beginning, making it difficult to clearly specify the global reference. Our initial solution is to fit a preference model based on the residence time of eye ray (captured through eye tracking) for each color suggestion (Fig. 2e). The preference model will then be integrated and refined by using an interactive evolutionary computing framework [56] to incorporate the global preference.
- Multi-objective optimization. Although most parameters are automatically determined according to the statistical analysis of inhabitant behavior and generally fixed in the experiments, we still need to fine-tune different weights corresponding to the optimization energies. This work uses a sequential strategy to fix the weight for harmony as a large value due to its importance and the large solution space. The weights for mood and anchor constraints are tuned according to preference, and the contrast term is considered finally. Our future work may consider solving this problem with multi-objective optimization and avoid the challenge of parameter tuning.
- Computational efficiency. The current implementation uses the MCMC camera placement together with orientation filtering to reduce the redundant computation in the double integral of Equation (1). We would like to investigate more efficient strategies for defining an indicator on each object surface reflecting how the surface could attract user attention. This indicator could further assist in finding optimal viewpoints.

Acknowledgments

The authors would like to thank the anonymous reviewers for their constructive comments. This work was supported in part by the National Natural Science Foundation of China under Grants 62077039, 62072383, 61702433, the Fundamental Research Funds for the Central Universities China under Grant 20720190006, Ministry of Science and Technology 108-2221-E-006-038-MY3 Taiwan and Singapore Ministry of Education Tier 1 Grant RG20/20.

References

[1] M. Shugrina, A. Kar, S. Fidler, and K. Singh, "Nonlinear color triads for approximation, learning and direct manipulation of color distributions," *ACM Trans. Graph.*, vol. 39, no. 4, 2020. [Online]. Available: https://doi.org/10.1145/3386569.3392461

[2] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 689–694, 2004.

[3] Q. Luan, F. Wen, D. Cohen-Or, L. Liang, Y.-Q. Xu, and H.-Y. Shum, "Natural image colorization," in *Proc. 18th Eurographics Conf. Rendering Techn.*, pp. 309–320.

[4] Y. Qu, T.-T. Wong, and P.-A. Heng, "Manga colorization," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 1214–1220, 2006.

[5] X. Liu *et al.*, "Intrinsic colorization," *ACM Trans. Graph.*, vol. 27, no. 5, 2008, Art. no. 152.

[6] A. Y.-S. Chia *et al.*, "Semantic colorization with internet images," *ACM Trans. Graph.*, vol. 30, no. 6, 2011, Art. no. 156.

[7] A. Bugeau, V.-T. Ta, and N. Papadakis, "Variational exemplar-based image colorization," *IEEE Trans. Image Process.*, vol. 23, no. 1, pp. 298–307, Jan. 2014.

[8] Z. Cheng, Q. Yang, and B. Sheng, "Deep colorization," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 415–423.

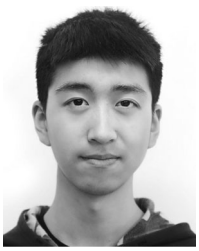
[9] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Let there be color!: Joint end-to-end learning of global and local image priors for automatic image colorization," *ACM Trans. Graph.*, vol. 35, no. 4, 2016, Art. no. 110.

[10] R. Zhang *et al.*, "Real-time user-guided image colorization with learned deep priors," *ACM Trans. Graph.*, vol. 36, no. 4, 2017, Art. no. 119.
[11] P. Sangkloy, J. Lu, C. Fang, F. Yu, and J. Hays, "Scribbler: Controlling deep image synthesis with sketch and color," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, vol. 1, pp. 6836–6845.
[12] M. He, D. Chen, J. Liao, P. V. Sander, and L. Yuan, "Deep exemplar-based colorization," *ACM Trans. Graph.*, vol. 37, no. 4, 2018, Art. no. 47.
[13] K. Chen, K. Xu, Y. Yu, T.-Y. Wang, and S.-M. Hu, "Magic decorator: Automatic material suggestion for indoor digital scenes," *ACM Trans. Graph.*, vol. 34, no. 6, 2015, Art. no. 232.
[14] G. Chen, G. Li, Y. Nie, C. Xian, and A. Mao, "Stylistic indoor colour design via Bayesian network," *Comput. Graph.*, vol. 60, pp. 34–45, 2016.
[15] J. Zhu, Y. Guo, and H. Ma, "A data-driven approach for furniture and indoor scene colorization," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 9, pp. 2473–2486, Sep. 2018.
[16] G. Leifman and A. Tal, "Mesh colorization," *Comput. Graph. Forum*, vol. 31, no. 2, pp. 421–430, 2012.
[17] A. Jain, T. Thormählen, T. Ritschel, and H.-P. Seidel, "Material Memex: Automatic material suggestions for 3D objects," *ACM Trans. Graph.*, vol. 31, no. 6, 2012, Art. no. 143.
[18] C. H. Nguyen, T. Ritschel, K. Myszkowski, E. Eisemann, and H. P. Seidel, "3D material style transfer," *Comput. Graph. Forum*, vol. 31, no. 2, pp. 431–438, 2012.
[19] J. Itten, *The Art of Color: The Subjective Experience and Objective Rationale of Color*, 3rd ed. New York, NY, USA: Van Nostrand, 1966.
[20] J. Krause, *Color Index. HOW Books*, New York, USA, 2010.
[21] Y. Matsuda, *Color Design*. Tokyo, Japan: Asakura Shoten, 1995.
[22] D. Cohen-Or, O. Sorkine, R. Gal, T. Leyvand, and Y.-Q. Xu, "Color harmonization," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 624–630, 2006.
[23] M. Tokumaru, N. Muranaka, and S. lmanishi, "Color design support system considering color harmony," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, pp. 378–383.
[24] C. Li and T. Chen, "Aesthetic visual quality assessment of paintings," *IEEE J. Sel. Topics Signal Process.*, vol. 3, no. 2, pp. 236–252, Apr. 2009.
[25] P. O'Donovan, A. Agarwala, and A. Hertzmann, "Color compatibility from large datasets," *ACM Trans. Graph.*, vol. 30, no. 4, 2011, Art. no. 63.
[26] A. Nemcsics, "The coloroid color system," *Color Res. Appl.*, vol. 5, no. 2, pp. 113–120, 1980. [Online]. Available: https://www.onlinelibrary.wiley.com/doi/abs/10.1002/col.5080050214
[27] T. Sato, K. Kajiwara, H. Hoshino, and T. Nakamura, "Quantitative evaluation and categorising of human emotion induced by colour," *Advances Colour Sci. Technol.*, vol. 3, pp. 53–59, 2000.
[28] L.-C. Ou, M. R. Luo, A. E. Woodcock, and A. Wright, "A study of colour emotion and colour preference. Part I: Colour emotions for single colours," *Color Res. Appl.*, vol. 29, no. 3, pp. 232–240, 2004.
[29] L.-C. Ou, M. R. Luo, A. E. Woodcock, and A. Wright, "A study of colour emotion and colour preference. Part II: Colour emotions for two-colour combinations," *Color Res. Appl.*, vol. 29, no. 4, pp. 292–298, 2004.
[30] B. Wang, Y. Yu, T.-T. Wong, C. Chen, and Y.-Q. Xu, "Data-driven image color theme enhancement," *ACM Trans. Graph.*, vol. 29, no. 6, 2010, Art. no. 146.
[31] P. Olivier and M. Christie, "Camera control in computer graphics: Models, techniques and applications," in *Proc. ACM SIGGRAPH ASIA Courses*, 2009, Art. no. 3.
[32] J. Blinn, "Where am I? What am i looking at?," *IEEE Comput. Graphics Appl.*, vol. 8, no. 4, pp. 76–81, Jul. 1988.
[33] R. Ranon and T. Urli, "Improving the efficiency of viewpoint composition," *IEEE Trans. Vis. Comput. Graphics*, vol. 20, no. 5, pp. 795–807, May 2014.
[34] L.-W. He, M. F. Cohen, and D. H. Salesin, "The virtual cinematographer: A paradigm for automatic real-time camera control and directing," in *Proc. 23rd Annu. Conf. Comput. Graph. Interactive Techn.*, 1996, pp. 217–224.
[35] C. Lino and M. Christie, "Intuitive and efficient camera control with the toric space," *ACM Trans. Graph.*, vol. 34, no. 4, 2015, Art. no. 82.
[36] T. Oskam, R. W. Sumner, N. Thuerey, and M. Gross, "Visibility transition planning for dynamic camera control," in *Proc. ACM SIGGRAPH/Eurographics Symp. Comput. Animation*, 2009, pp. 55–65.
[37] C. Lino, M. Christie, F. Lamarche, G. Schofield, and P. Olivier, "A real-time cinematography system for interactive 3D environments," in *Proc. ACM SIGGRAPH/Eurographics Symp. Comput. Animation*, 2010, pp. 139–148.
[38] N. Joubert, M. Roberts, A. Truong, F. Berthouzoz, and P. Hanrahan, "An interactive tool for designing quadrotor camera shots," *ACM Trans. Graph.*, vol. 34, no. 6, 2015, Art. no. 238.
[39] K. Xie *et al.*, "Creating and chaining camera moves for quadrotor videography," *ACM Trans. Graph.*, vol. 37, no. 4, 2018, Art. no. 88.
[40] K. Sims, "Artificial evolution for computer graphics," *ACM SIGGRAPH Comput. Graph.*, vol. 25, pp. 319–238, 1991.
[41] K. Sims, "Evolving virtual creatures," in *Proc. 21st Annu. Conf. Comput. Graph. Interactive Techn.*, 1994, pp. 15–22.
[42] M. L. Pilat and C. Jacob, "Creature academy: A system for virtual creature evolution," in *Proc. IEEE Congr. Evol. Comput.*, 2008, pp. 3289–3297.
[43] K. Haubenwallner, H.-P. Seidel, and M. Steinberger, "ShapeGenetics: Using genetic algorithms for procedural modeling," *Comput. Graph. Forum*, vol. 36, no. 2, pp. 213–223, 2017.
[44] K. Xu, H. Zhang, D. Cohen-Or, and B. Chen, "Fit and diverse: Set evolution for inspiring 3D shape galleries," *ACM Trans. Graph.*, vol. 31, no. 4, 2012, Art. no. 57.
[45] Y. Jin, "Surrogate-assisted evolutionary computation: Recent advances and future challenges," *Swarm Evol. Comput.*, vol. 1, pp. 61–70, 2011.
[46] Y.-S. Ong, Z. Zhou, and D. Lim, "Curse and blessing of uncertainty in evolutionary algorithm using approximation," in *Proc. IEEE Int. Conf. Evol. Comput.*, 2006, pp. 2928–2935.
[47] M. Savva, A. X. Chang, P. Hanrahan, M. Fisher, and M. Nießner, "SceneGrok: Inferring action maps in 3D environments," *ACM Trans. Graph.*, vol. 33, no. 6, 2014, Art. no. 212.
[48] D. Lamming, *Contrast Sensitivity*, vol. 5. London, U.K.: Macmillan Press, 1991.
[49] Y.-F. Ma and H.-J. Zhang, "Contrast-based image attention analysis by using fuzzy growing," in *Proc. 11th ACM Int. Conf. Multimedia*, 2003, pp. 374–381.
[50] S. Lin, D. Ritchie, M. Fisher, and P. Hanrahan, "Probabilistic color-by-numbers: Suggesting pattern colorizations using factor graphs," *ACM Trans. Graph.*, vol. 32, no. 4, 2013, Art. no. 37.
[51] H. Chang, O. Fried, Y. Liu, S. DiVerdi, and A. Finkelstein, "Palette-based photo recoloring," *ACM Trans. Graph.*, vol. 34, no. 4, 2015, Art. no. 139.
[52] S. A. Mitchell *et al.*, "Spoke darts for efficient high dimensional blue noise sampling," *ACM Trans. Graph.*, vol. 37, no. 2, 2018, Art. no. 22.
[53] D. G. Krige, "A statistical approach to some basic mine valuations problems on the Witwatersrand," *J. Chem. Metallurgical Mining Eng. Soc. South Africa*, vol. 52, no. 6, pp. 119–139, 1951.
[54] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, "Design and analysis of computer experiments," *Statist. Sci.*, vol. 4, pp. 409–423, 1989.
[55] S. N. Lophaven, H. B. Nielsen, and J. Søndergaard, "DACE: A Matlab kriging toolbox," Tech. Univ. Denmark, Kgs. Lyngby, Denmark, Rep. IMM-TR-2002–12, 2002.
[56] H. Takagi, "Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation," *Proc. IEEE*, vol. 89, no. 9, pp. 1275–1296, Sep. 2001.

Juncong Lin received the BS and PhD degrees from Zhejiang Univesity, Hangzhou, China, in 2003 and 2008, respectively. He is currently a full professor with the School of Informatics, Xiamen University, leading the Graphics and Virtual Reality Laboratory. Before joining Xiamen University, he has worked with CUHK, NTU as a postdoc researcher. His research interests include shape modeling with creativity support, sketch based shape modeling, and comic content processing.

Pintong Xiao is currently working toward the undergraduate degree with the School of Informatics, Xiamen University, Xiamen, China and supervised by professor Juncong Lin and associate professor Shihui Guo. Her research interests include computer graphics and virtual reality.

Yinan Fu is currently working toward the undergraduate degree with the School of Informatics, Xiamen University, Xiamen, China and supervised by professor Juncong Lin and associate professor Shihui Guo. Her research interests include computer graphics and virtual reality.



Yubin Shi is currently working toward the undergraduate degree with the School of Informatics, Xiamen University, Xiamen, China and supervised by professor Juncong Lin and associate professor Shihui Guo. His research interests include computer graphics and virtual reality.

Hongran Wang received the bachelor's degree from the School of Informatics, Xiamen University, Xiamen, China, in 2020. His research interests include computer graphics and virtual reality.

Shihui Guo (Senior Member, IEEE) received the BS degree in electrical engineering from Peking University, Beijing, China, and the PhD degree in computer animation from the National Centre for Computer Animation, Bournemouth University, Poole, U.K. He is currently an associate professor with the School of Informatics, Xiamen University. His research interests include character animation and 3D modeling.

Ying He received the bachelor's and master's degrees in electrical engineering from Tsinghua University, Beijing, China, in 1997 and 2000, respectively, and the PhD degree in computer science from Stony Brook University, Stony Brook, New York, in 2006. He is currently an associate professor with the School of Computer Engineering, Nanyang Technological University, Singapore. He is interested in the problems that require geometric computing and analysis.

Tong-Yee Lee (Senior Member, IEEE) received the PhD degree in computer engineering from Washington State University, Pullman, Washington, in 1995. He is currently a chair professor with the Department of Computer Science and Information Engineering, National ChengKung University, Tainan, Taiwan, ROC. He leads the Computer Graphics Group, Visual System Laboratory, National Cheng-Kung University. His research interests include computer graphics, nonphotorealistic rendering, medical visualization, and VR. He is a senior member of the IEEE Computer Society.