Automatic Blending of Multiple Perspective Views for Aesthetic Composition

Kairi Mashio[1], Kenichi Yoshida[1], Shigeo Takahashi[1], and Masato Okada[1,2]

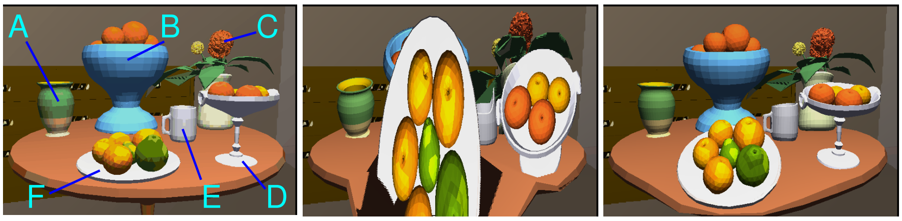
[1] The University of Tokyo, Japan, Chiba, 277-8561 Japan
[2] RIKEN, Japan, Saitama, 351-0198 Japan

Abstract. Hand-drawn pictures differ from ordinary perspective images in that the entire scene is composed of local feature regions each of which is projected individually as seen from its own vista point. This type of projection, called nonperspective projection, has served as one of the common media for our visual communication while its automatic generation process still needs more research. This paper presents an approach to automatically generating aesthetic nonperspective images by simulating the deformation principles seen in such hand-drawn pictures. The proposed approach first locates the optimal viewpoint for each feature region by maximizing the associated viewpoint entropy value. These optimal viewpoints are then incorporated into the 3D field of camera parameters, which is represented by regular grid samples in the 3D scene space. Finally, the camera parameters are smoothed out in order to eliminate any unexpected discontinuities between neighboring feature regions, by taking advantage of image restoration techniques. Several nonperspective images are generated to demonstrate the applicability of the proposed approach.

1 Introduction

In computer graphics, ordinary perspective projection is commonly used to transform 3D scenes onto the 2D screen space to simulate the effects of the pinhole camera model. On the other hand, hand-drawn pictures often differ from such ordinary perspective images in that the entire scene is composed of local feature areas each of which is projected as seen from its own optimal viewpoint. This type of projection, called *nonperspective projection*, has been synthesized by selecting such local viewpoints so that they retain the visual consistency with the overall scene composition. Furthermore, employing such local viewpoints allows us to emphasize/suppress specific objects and to avoid unexpected occlusions between objects, which is especially useful in drawing 3D map illustrations and artistic paintings.

Several models for nonperspective projection have been proposed by simulating the process of hand-drawn pictures and effects of magnification lenses. Agrawala et al. [1] proposed *artistic multiprojection rendering*, where they projected each individual object from a different viewpoint and merged the projected



(a) (b) (c)

Fig. 1. Nonperspective image of a 3D scene generated using our method. (a) Ordinary perspective image. (b) Nonperspective image with non-smooth blending of local viewpoints. (c) Nonperspective image with smooth blending of local viewpoints.

images pixel by pixel consistently. Kurzion et al. [2] simulated 3D object deformations by bending the sight rays together with hardware-assisted 3D texturing. Singh et al. [3] presented a generalized approach, called a *fresh perspective*, to generate smoothly deformed nonperspective images by arranging 3D local cameras and interpolating the corresponding camera parameters in the 3D scene space, and extended it to realize an interactive interface that directly manipulates 2D image deformations [4]. Deforming the target 3D objects allows us to simulate the effect of composing nonperspective images as seen from multiple viewpoints. This idea can be traced back to the concept of *view-dependent geometry* proposed by Rademacher [5]. Martin et al. [6] implemented *observer dependent deformations* by relating user-defined nonlinear transformations of 3D objects to the viewing distance and camera orientation. Takahashi et al. optimally deformed 3D terrain models for generating mountain guide-maps [7] and occlusion-free route navigation [8].

Basically, the process of designing nonperspective images consists of three steps: (1) segmenting the entire scene into local feature regions, (2) selecting the optimal viewpoint for each local region, and (3) generating a composite image from local regions each of which is projected as seen from its viewpoint. In practice, the above existing approaches successfully provide us with an interface for designing such nonperspective images, however, its design still needs a time-consuming trial and error process. This is because the design of nonperspective images inherently has excessive degrees of freedom in nature, and existing interfaces do not provide any guidelines for selecting appropriate viewpoints for local feature regions and aesthetically blending the corresponding camera parameters within the 3D scene space. Figure 1(b) shows such an example where excessive image distortions occur due to inappropriate selection and interpolation of local viewpoints assigned to the feature objects as indicated in Figure 1(a).

This paper presents an approach to generating an aesthetic nonperspective image from a set of segmented feature regions in the target 3D scene. Our approach automates the latter two steps in the above nonperspective design process, the selection of optimal viewpoints for feature regions and its visually plausible blending in the final composite image. The viewpoint selection is accomplished by employing adaptive Monte Carlo sampling of the viewpoint entropy function around the user-specified global viewpoint. The obtained local viewpoints will be interpolated over the 3D scene to compose a nonperspective image in a visually plausible manner, by taking advantage of image restoration techniques. The proposed formulation allows us to synthesize nonperspective images only by adjusting a single parameter that controls the smoothness of the viewpoint transition over the entire 3D scene. Several design examples including Figure 1(c) will be exhibited to demonstrate the feasibility of the proposed approach.

The remainder of this paper is organized as follows: Section 2 explains how we can calculate the optimal viewpoint for each feature region using the Monte Carlo sampling technique. Section 3 introduces a 3D field of camera parameters as the data structure to retain the calculated local viewpoints. Section 4 describes a method of interpolating such local viewpoints over the 3D scene using the image restoration techniques. After presenting several design examples in Section 5, Section 6 concludes this paper and refers to possible future extensions.

2 Selecting Optimal Viewpoints

Our design process starts with an ordinary perspective image projected from the user-specified global viewpoint, and then tries to find nonperspective image deformation by incorporating a locally optimal viewpoint calculated for each feature region. In our approach, users are requested to specify a set of representative objects so that the target 3D scene can be segmented into local feature regions. In this section, we explain how to select the optimal viewpoint for each feature region by the Monte Carlo sampling of the corresponding viewpoint entropy function.

2.1 Viewpoint Entropy

For finding the optimal viewpoints for representative objects, we employ the formulation of the *viewpoint entropy* proposed by Vázquez et al. [9]. In this formulation, the optimality of a viewpoint is evaluated by the following entropy function:

$$I = -\sum_{j=0}^{N} \frac{A_j}{S} \log_2 \frac{A_j}{S}, \tag{1}$$

where N represents the number of faces that compose a 3D polygonal model of the target object, and A_j is the projected area of the j-th polygonal face of that model on the 2D screen. Here, we assume that A_0 represents the area of the background region and thus the overall area of the screen S holds the following condition: $S = \sum_{j=0}^{N} A_j$. Eq. (1) implies that the definition of the viewpoint entropy is equivalent to that of the Shannon entropy, when we think of the relative ratio A_j/S as the probability of the corresponding face visibility. Thus, we can find the optimal viewpoint by exploring the best balance of the face visibilities based on the entropy formulation.

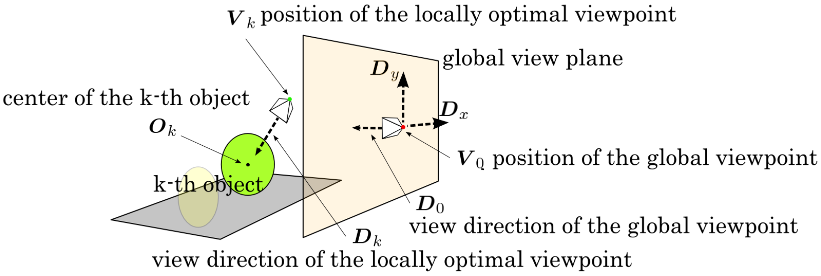

V_k position of the locally optimal viewpoint
global view plane
center of the k-th object
O_k
D_y
D_x
k-th object
V_0 position of the global viewpoint
D_0
view direction of the global viewpoint
D_k
view direction of the locally optimal viewpoint

Fig. 2. Global viewpoint and locally optimal viewpoints. Each viewpoint is represented by the 3D geometric position and view direction.

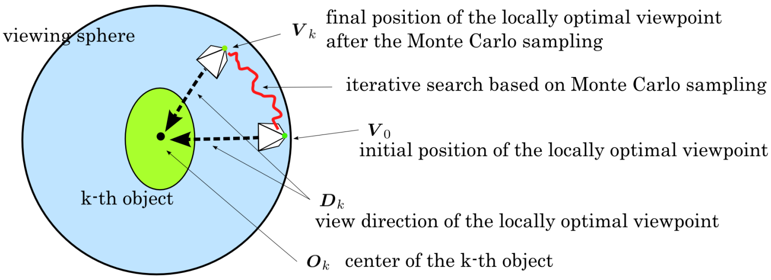

viewing sphere
V_k
final position of the locally optimal viewpoint
after the Monte Carlo sampling
iterative search based on Monte Carlo sampling
V_0
initial position of the locally optimal viewpoint
k-th object
D_k
view direction of the locally optimal viewpoint
O_k center of the k-th object

Fig. 3. Optimal viewpoint selection on the viewing sphere using Monte Carlo sampling.

This search for optimal viewpoints can be replaced with a more sophisticated version based on surface curvatures [10] if we put more weight on the salient features of the object shapes, or semantic-based approaches if the target scene has domain-specific context [11–13]. Readers can refer to an excellent overview [14] of this interesting subject.

2.2 Viewpoint Selection based on Monte Carlo Sampling

Before going into the details of the actual viewpoint calculation, we introduce our notation. In this paper, we represent each viewpoint by the 3D geometric position and view direction. Figure 2 illustrates this notation where the position and view direction for the k-th object are denoted by $\boldsymbol{V}_k$ and $\boldsymbol{D}_k$, respectively, while $\boldsymbol{V}_0$ and $\boldsymbol{D}_0$ correspond to those for the user-specified global viewpoint for the initial perspective projection. In addition, we refer to the plane perpendicular to the view direction $\boldsymbol{D}_0$ as the *global view plane*, and denote the horizontally and vertically aligned vectors that span the plane by $\boldsymbol{D}_x$ and $\boldsymbol{D}_y$.

In practice, we can compute the optimal viewpoint of the k-th object as follows: First, the viewpoint $\boldsymbol{V}_k$ for the k-th object is initialized to $\boldsymbol{V}_0$, and the view direction $\boldsymbol{D}_k$ is set to be a vector emanating from $\boldsymbol{V}_0$ to the center of the k-th object $\boldsymbol{O}_k$. For exploring the best viewpoint, as shown in Figure 3, we iteratively perform the Monte Carlo sampling in the vicinity of the current position $\boldsymbol{V}_k$ over the viewing sphere of the k-th object, and replace it with the new position if it has a higher value of the viewpoint entropy than the current

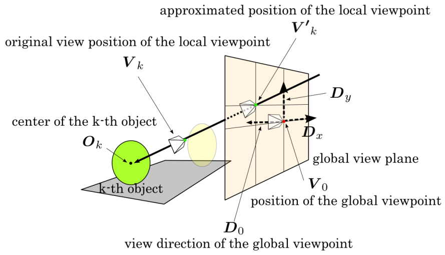

approximated position of the local viewpoint
$\boldsymbol{V'}_k$
original view position of the local viewpoint
$\boldsymbol{V}_k$
$\boldsymbol{D}_y$
center of the k-th object
$\boldsymbol{O}_k$
$\boldsymbol{D}_x$
global view plane
k-th object
$\boldsymbol{V}_0$
position of the global viewpoint
$\boldsymbol{D}_0$
view direction of the global viewpoint

Fig. 4. Projecting the local viewpoint onto the global view plane associated with the global viewpoint.

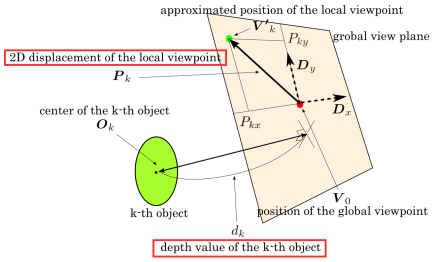

approximated position of the local viewpoint
$\boldsymbol{V'}_k$
P_{ky}
grobal view plane
2D displacement of the local viewpoint
$\boldsymbol{D}_y$
$\boldsymbol{P}_k$
center of the k-th object
$\boldsymbol{O}_k$
$\boldsymbol{D}_x$
P_{kx}
$\boldsymbol{V}_0$
k-th object
position of the global viewpoint
d_k
depth value of the k-th object

Fig. 5. 2D displacement vector and the depth value of the viewpoint to be stored in the 3D field of camera parameters.

value. We also set $\boldsymbol{D}_k$ to the vector from $\boldsymbol{V}_k$ to $\boldsymbol{O}_k$ at each updating step. This iterative computation will terminate when we reach a local maximum around the the position of initial global viewpoint, and employ the corresponding $\boldsymbol{V}_k$ and $\boldsymbol{D}_k$ as the optimal viewpoint for that object.

3 3D Field of Camera Parameters

Before synthesizing nonperspective images by interpolating the obtained local viewpoints, we introduce a 3D field of camera parameters on regular grid samples in the target 3D scene. This is because, from the 3D field of camera parameters, we can retrieve a locally optimal viewpoint at any 3D position in the target scene by trilinearly interpolating the eight nearest neighbor samples. Furthermore, the regular grid samples permit us to naturally incorporate image restoration techniques into this framework when smoothly blending the locally optimal viewpoints in a visually plausible manner. In what follows, we first introduce the representation of a locally optimal viewpoint for each object in our framework, then store the corresponding viewpoint information into the 3D field of camera parameters, and finally synthesize the corresponding nonperspective image with reference to that 3D field.

3.1 Approximating the Locally Optimal Viewpoints

In the 3D field of camera parameters, we first store the 2D displacements of the local viewpoints from their original position for generating the initial perspective image, while having fixed other parameters such as the focal length, view direction, up vector, size of the screen and its orientation. This setting effectively allows us to control the associated nonperspective projection without worrying about the excessive degrees of freedom inherent in the design of such nonperspective images. However, we cannot describe a locally optimal viewpoint assigned to each representative object only with the 2D displacement, and thus have to approximate the position of the local viewpoint by projecting it to the

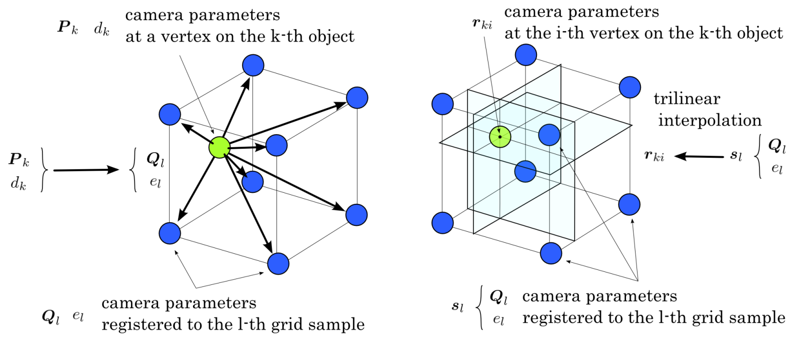

P_k d_k camera parameters
at a vertex on the k-th object
P_k
d_k
Q_l
e_l
Q_l e_l camera parameters
registered to the l-th grid sample
r_{ki} camera parameters
at the i-th vertex on the k-th object
trilinear
interpolation
r_{ki}
s_l
Q_l
e_l
s_l Q_l e_l camera parameters
registered to the l-th grid sample

Fig. 6. Registering camera parameters to the 3D field: (Left) Storing the camera parameters to the nearest eight grid samples. (Right) Retrieving the camera parameters by trilinearly interpolating those at the nearest eight grid samples.

global view plane along the corresponding view direction. Figure 4 shows such an viewpoint approximation where the original viewpoint for the k-th object $\boldsymbol{V}_k$ is projected onto the new position $\boldsymbol{V}'_k$ as:

$$\boldsymbol{V}'_k = \frac{(\boldsymbol{O}_k - \boldsymbol{V}_0) \cdot \boldsymbol{D}_0}{(\boldsymbol{O}_k - \boldsymbol{V}_k) \cdot \boldsymbol{D}_0} (\boldsymbol{V}_k - \boldsymbol{O}_k) + \boldsymbol{O}_k. \tag{2}$$

This approximation effectively allows us to store the information about the local viewpoints into the 3D field of camera parameters.

3.2 Storing Camera Parameters in the 3D Field

As described earlier, the 2D displacement of each locally optimal viewpoint can be obtained by calculating the difference in position between the new viewpoint and original global viewpoint along the global view plane, as $\boldsymbol{V}'_k - \boldsymbol{V}_0$. Thus, as shown in Figure 5, the 2D displacement vector $\boldsymbol{P}_k = (P_{kx}, P_{ky})$ to be stored in the 3D field can be given by

$$P_{kx} = (\boldsymbol{V}'_k - \boldsymbol{V}_0) \cdot \boldsymbol{D}_x \quad \text{and} \quad P_{ky} = (\boldsymbol{V}'_k - \boldsymbol{V}_0) \cdot \boldsymbol{D}_y. \tag{3}$$

In addition to the 2D viewpoint displacement, we also associate the distance between the global view plane to the target representative object, as its depth value. Figure 5 again indicates the depth value of the k-th object, which can be given by

$$d_k = (\boldsymbol{O}_k - \boldsymbol{V}_0) \cdot \boldsymbol{D}_0. \tag{4}$$

These 2D displacement vectors and depth values for local viewpoints are recorded as the camera parameters at the grid samples in the 3D field.

3.3 Synthesizing Nonperspectives with the Camera Parameters

Suppose that the target scene has been composed of a set of 3D meshes. For synthesizing the nonperspective image of the target scene, we have to retrieve the camera parameters of each vertex on the meshes from the 3D field of camera parameters. However, since the mesh vertices do not necessarily coincide with the grid samples of the 3D field geometrically, we have to register the aforementioned 2D displacement vectors and depth values of the mesh vertices appropriately to the 3D field. In our implementation, we propagate the camera parameters of each mesh vertex to the eight nearest grid samples as shown on the left of Figure 6. Here, in this figure, the 2D displacement of the viewpoint $\boldsymbol{P}_k$ and depth value d_k for the k-th object are distributed to the l-th grid sample as $\boldsymbol{Q}_l$ and e_l, respectively. On the other hand, when synthesizing nonperspective images by referring to the 3D field of camera parameters, we compute the camera parameters at the 3D position of a mesh vertex by trilinearly interpolating those at the eight nearest grid samples as shown on the right of Figure 6.

In practice, we simulate the effects of nonperspective projection by projecting each mesh vertex with reference to the corresponding 2D displacement of viewpoint and depth value. Figure 7 shows such an example where the change in the viewpoint position results in the deformation of the target object in the final nonperspective image. Note that, when synthesizing the nonperspective image in this scheme, we fix the center of each representative object in the 3D scene space. This implies that, as shown in Figure 8, we have to translate the vertex $\boldsymbol{x}$ along the global view plane by:

$$\boldsymbol{r} = \big(\frac{d - d_k}{d_k}\big)\boldsymbol{P}_k \quad \text{and} \quad d = (\boldsymbol{x} - \boldsymbol{V}_0) \cdot \boldsymbol{D}_0, \tag{5}$$

where d represents the depth value of the vertex $\boldsymbol{x}$ in the global view coordinates. This formulation successfully allows us to synthesize nonperspective images if an appropriate 3D field of camera parameters is provided. Note that the registered locally optimal camera parameters are not used directly; they will be smoothed out when synthesizing aesthetic nonperspective images as described in the next section.

4 Blending Local Viewpoints

In generating final nonperspective images, we have to fully smooth out the 3D field of camera parameters. Otherwise, we may incur unexpected visual inconsistency such as folds and breaks over the projected images around possible sudden changes in the camera parameters. In this section, we introduce a method of smoothly blending the locally optimal viewpoints that correspond to the representative objects in the 3D scene, by taking advantage of image restoration techniques [15].

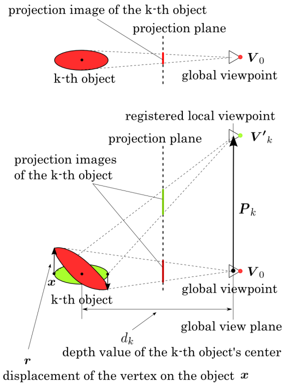


Fig. 7. Simulating nonperspective projection by deforming the target object.

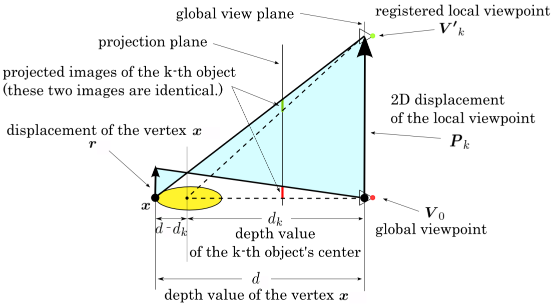

global view plane
registered local viewpoint
V'_k
projection plane
projected images of the k-th object
(these two images are identical.)
2D displacement
of the local viewpoint
P_k
displacement of the vertex x
r
x
V_0
global viewpoint
d - d_k
d_k
depth value
of the k-th object's center
d
depth value of the vertex x

Fig. 8. Displacement of a vertex according to the change in the position of the corresponding viewpoint.

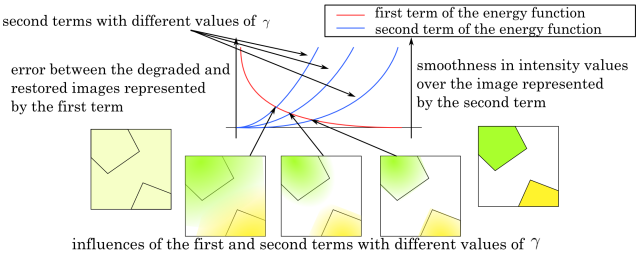

second terms with different values of γ
first term of the energy function
second term of the energy function
error between the degraded and
restored images represented
by the first term
smoothness in intensity values
over the image represented
by the second term
influences of the first and second terms with different values of γ

Fig. 9. Image restoration process by minimizing an energy function. The parameter γ controls the smoothness of the restored image.

4.1 Energy Function for Blending Viewpoints

Basically, image restoration techniques restore a degraded noisy image by seeking smooth spatial change in intensity value, while minimizing the difference between the degraded image and newly restored image. For example, we can transform the degraded image $\boldsymbol{g}$ into the restored image $\boldsymbol{f}$, by minimizing the following function F (Figure 9):

$$F = \frac{1}{2}\sum_{i}^{L}(g_i - f_i)^2 + \frac{\gamma}{2}\sum_{(j,k)}(f_j - f_k)^2, \tag{6}$$

where f_i and g_i represent the intensity values at the i-th pixel in $\boldsymbol{f}$ and $\boldsymbol{g}$, respectively, and L corresponds to the number of pixels in the images. In addition, $\sum_{(j,k)}$ indicates that we calculate the summation of the corresponding terms for all the pairs of neighboring pixels in the image. (The j-th and k-th pixels are immediate neighbors in this case.) On the right side of Eq. (6), the first term represents the error between the degraded image $\boldsymbol{g}$ and restored image $\boldsymbol{f}$, and the second term evaluates the smoothness of the intensity values over the restored image $\boldsymbol{f}$, while γ denotes the relative ratio of the reliability of $\boldsymbol{f}$ with respect to $\boldsymbol{g}$. The best restored image can be obtained by minimizing the energy function F.

A similar strategy can be used to achieve the smooth interpolation between local viewpoints, by smoothing out the transition of camera parameters in the 3D space while maximally respecting the calculated locally optimal viewpoints. In practice, this problem can be formulated by introducing an energy function that evaluates the quality of the 3D field of camera parameters, with reference to the definition of the energy function in Eq. (6). Let s_l denote the displacement of the position at the l-th grid sample in the 3D field of camera parameters, as shown on the left of Figure 6. We define the energy function that evaluates the quality of the 3D field of camera parameters as

$$E = \frac{1}{2}\sum_{i=1}^{N}(s_i^n - s_i^o)^2 + \frac{\gamma}{2}\sum_{(j,k)}(s_j^n - s_k^n)^2, \tag{7}$$

where s_i^o and s_i^n represent the displacement before and after the optimization, and $\sum_{(j,k)}$ indicates that we calculate the summation of the corresponding terms for all the pairs of neighboring grid samples in the 3D field. In the same way as in Eq. (6), the first term on the right side of Eq. (7) represents the difference between the 3D field of camera parameters obtained by calculated local viewpoints and its updated version, while the second term represents the smoothness of the updated 3D field. Here, the parameter value γ controls the smoothness of the 3D field of camera parameters and can be adjusted by users for tweaking aesthetic continuity of the interpolated local viewpoints in the nonperspective scene.

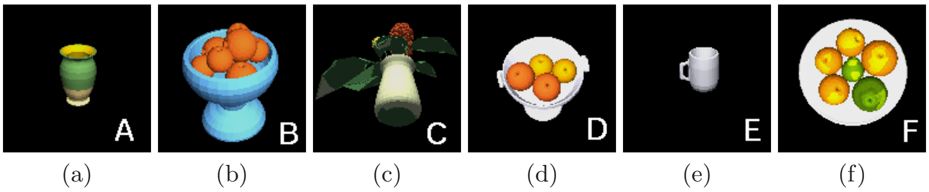
4.2 Iteratively Updating Camera Parameters

Given an appropriate parameter value γ, we explore the 3D field of camera parameters that minimizes the energy function in Eq. (7), which provides us with the optimal interpolation of local viewpoints assigned to the representative objects. In our approach, this is achieved by iteratively updating the camera parameters using the steepest descend method to seek the minimal value of the energy function. For each update, we replace the old displacement of the l-th grid sample s_l^o with the new one s_l^n, by zeroing the derivative of Eq. (7) with respect to s_l^n, as

$$\frac{\partial E}{\partial s_l^n} = (s_l^n - s_l^o) + \gamma \sum_{m\in\text{nearest}} (s_l^n - s_m^o) = 0. \tag{8}$$

Here, $\sum_{m\in\text{nearest}}$ represents the summation of the corresponding terms for all the nearest grid samples of the l-th sample, while the number of the nearest samples is 6 in this 3D case because the l-th sample has two neighbors along each of the three coordinate axes. This means that we can rewrite Eq. (8) as

$$s_l^n = \frac{s_l^o + \gamma \sum_{m\in\text{nearest}} s_m^o}{1 + 6\gamma} \tag{9}$$



(a) (b) (c) (d) (e) (f)

Fig. 10. Representative objects for synthesizing the nonperspective image in Figure 1, where each object is projected as seen from its own optimal viewpoint.

By applying the above updates to all the grid samples, we can iteratively optimize the 3D field of camera parameters. If the difference in the energy function between the current field and updated field becomes less than some specific threshold, we consider that the 3D field has been converged to the optimal one and terminate this iterative process. Finally, we employ the final 3D field of camera parameters to synthesize aesthetic compositions of nonperspective images.

5 Results

This section provides several design examples to demonstrate the capability of the proposed method. First, Figure 10 shows the set of representative objects specified by users for synthesizing the nonperspective scene in Figure 1, where each of which is projected as seen from its own optimal viewpoint. Note that Figure 1(a) shows how the representative objects look like in the ordinary perspective projection where they are projected with the given global viewpoint. We cannot inevitably avoid undesirable breaks and folds in the projected image as seen in Figure 1(b) by just storing the locally optimal viewpoints for the representative objects to the 3D field of camera parameters, because we have several stepwise discontinuities in the 3D field without any smoothing operations. After having smoothed out the 3D field using our approach, we can successfully interpolate the locally optimal viewpoints over the scene to synthesize a visually plausible nonperspective image as shown in Figure 1(c). Note that we use $\gamma = 10.0$ when minimizing the energy function in Eq. (7) in this case.

This design example nonetheless suggests that we cannot necessarily retain the locally optimal view of each representative object, for example, as seen through the comparison between optimal local views in Figure 10(a), (b), and (c) and the final synthesized image in Figure 1(c). Thus, we like to observe how the nonperspective image deformation changes according to the parameter value of γ. Figure 11 presents such an example, where we use another set of representative objects as our target scene, and generate nonperspective images with different values of γ. Figure 11(a) shows the corresponding ordinary perspective image, while Figures 11(b)-(f) present how the different values of γ will influence on the overall interpolation of local viewpoints. The associated results reveal that the small value of γ will not maintain the smoothness of the 3D field of camera parameters, as shown in Figure 11(b). On the other hand, the large value of γ

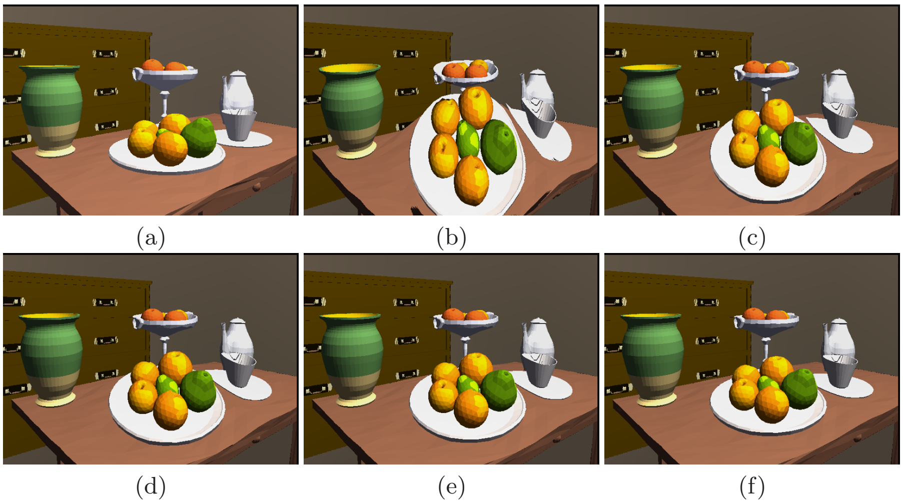


Fig. 11. Results with different values of γ. (a) Ordinary perspective image. Nonperspective images with smooth fields obtained by optimizing the energy function with (b) $\gamma = 3.0$, (c) $\gamma = 7.0$, (d) $\gamma = 10.0$, (e) $\gamma = 12.0$, and (f) $\gamma = 20.0$.

will excessively decrease the flexibility of the nonperspective image deformation, and thus make the projected image close to the original perspective image, as shown in Figure 11(f). This implies that we can control the visual plausibility of the nonperspective image deformation by controlling only one parameter γ, which is still manually adjusted by users in our implementation. Adaptively controlling the value of γ in the 3D target scene would help us improve the aesthetic interpolation of local viewpoints.

6 Conclusion

This paper has presented an approach for automatically synthesizing aesthetic nonperspective images by interpolating locally optimal viewpoints assigned to the given set of representative objects in the scene. The plausible interpolation of local viewpoints can be accomplished by adjusting only single control parameter so that the optimal view of each representative object can be sufficiently reflected in the final nonperspective image. Currently, our scheme cannot necessarily preserve all the locally optimal viewpoints in the synthesized nonperspective image, for example, when the corresponding viewpoints are quite different in its direction from the global viewpoint, or when different viewpoints are assigned to multiple representative objects within a relatively small region. Adaptively adjusting the control parameter in 3D scene space, together with non-regular samples of camera parameters, will allow us to respect the spatial configuration of representative objects in the final nonperspective image. Applying the present approach to nonperspective animations will demand more systematic control of camera parameters, and thus will an interesting theme for future research.

Acknowledgements

This work has been partially supported by Japan Society of the Promotion of Science under Grants-in-Aid for Scientific Research (A) No. 20240020, Scientific Research (B) No. 20300033, and Young Scientists (B) No. 17700092, and the Casio Science Promotion Foundation.

References

1. Agrawala, M., Zorin, D., Munzner, T.: Artistic multiprojection rendering. In: Eurographics Rendering Workshop 2000. (2000) 125–136
2. Kurzion, Y., Yagel, R.: Interactive space deformation with hardware-assisted rendering. IEEE Computer Graphics and Applications **17**(5) (1997) 66–77
3. Singh, K.: A fresh perspective. In: Proceedings of Graphics Interface 2002. (2002) 17–24
4. Coleman, P., Singh, K., Barrett, L., Sudarsanam, N., Grimm, C.: 3D scene-space widgets for non-linear projection. In: Proceedings of GRAPHITE 2005. (2005) 221–228
5. Rademacher, P.: View-dependent geometry. In: Proceedings of SIGGRAPH '99. (1999) 439–446
6. Martín, D., García, S., Torres, J.C.: Observer dependent deformations in illustration. In: Proceedings of the 1st International. Symposium on Non-Photorealistic Animation and Rendering (NPAR2000). (2000) 75–82
7. Takahashi, S., Ohta, N., Nakamura, H., Takeshima, Y., Fujishiro, I.: Modeling surperspective projection of landscapes for geographical guide-map generation. Computer Graphics Forum **21**(3) (2002) 259–268
8. Takahashi, S., Yoshida, K., Shimada, K., Nishita, T.: Occlusion-free animation of driving routes for car navigation systems. IEEE Transactions on Visualization and Computer Graphics **12**(5) (2006) 1141–1148
9. Vázquez, P.P., Feixas, M., M.Sbert, Heidrich, W.: Viewpoint selection using view entropy. In: Proceedings of Vision Modeling and Visualization Conference (VMV2001). (2001) 273–280
10. Lee, C.H., Varshney, A., Jacobs, D.W.: Mesh saliency. ACM Transactions on Graphics **24**(3) (2005) 659–666
11. Vázquez, P.P., Feixas, M., Sbert, M., Llobet, A.: Realtime automatic selection of good molecular views. Computers and Graphics **30**(1) (2006) 98–110
12. Sokolov, D., Plemenos, D., Tamine, K.: Viewpoint quality and global scene exploration strategies. In: Proceedings on International Conference on Computer Graphics Theory and Applications (GRAPP2006). (2006) 184–191
13. Mühler, K., Neugebauer, M., Tietjen, C., Preim, B.: Viewpoint selection for intervention planning. In: Proceedings of Eurographics/IEEE-VGTC Symposium on Visualization. (2007) 267–274
14. Elmqvist, N., Tsigas, P.: A taxonomy of 3d occlusion management for visualization. IEEE Transactions on Visualization and Computer Graphics **14**(5) (2008) 1095–1109
15. Geman, D.: Random fields and inverse problems in imaging. In: Lecture Notes in Mathematics. Volume 1427., Springer 113–193