

METHOD AND APPARATUS FOR GENERATING VIRTUAL OBJECT, DEVICE, STORAGE MEDIUM, AND PROGRAM PRODUCT

TECHNICAL FIELD

- [0001] This application relates, but is not limited to the technical field of computer vision, and in particular, to a method and apparatus for generating a virtual object, a device, a storage medium, and a program product.

BACKGROUND

- [0002] Reconstructing a three-dimensional virtual object from an image is a very challenging task. In the related art, in a case that there are no three-dimensional baseline information or multi-view data in a training set, certain cycle consistency is used to aid in training autocoders, so as to obtain a reconstructed three-dimensional virtual object. However, since methods for reconstructing the three-dimensional virtual object in the related art often pay more attentions to the visible portion of the object, the three-dimensional virtual object under a new angle of view looks unreal, and they can hardly be generalized for scenarios where the object in the image is uncommon and the object in the image is occluded.

SUMMARY

- [0003] In view of this, embodiments of this application provide at least a method and apparatus for generating a virtual object, a device, a storage medium, and a program product.
- [0004] The technical solutions of the embodiments of this application are implemented as follows.
- [0005] According to an aspect, an embodiment of this application provides a method for generating a virtual object. The method includes the following operations.
- [0006] A first image including a first object is acquired.
- [0007] Based on a first potential feature of a generative adversarial network, a first three-dimensional virtual object is generated by using the generative adversarial network. The generative adversarial network is trained by using a sample image set. Each sample image in the sample image set includes a second object that belongs to a same object category as the first object.
- [0008] The first three-dimensional virtual object is rendered as a second image according to a first camera pose of the first image.
- [0009] A first loss value is determined based on the first image and the second image.
- [0010] Based on the first loss value, the first potential feature and the first camera pose are updated to obtain a second potential feature and a second three-dimensional virtual object corresponding to the second potential feature. The second three-

dimensional virtual object is generated based on the second potential feature by using the generative adversarial network.

- [0011] According to another aspect, an embodiment of this application provides an apparatus for generating a virtual object. The apparatus includes a first acquisition module, a first generation module, a first rendering module, a first determination module, and a first updating module.
- [0012] The first acquisition module is configured to acquire a first image including a first object.
- [0013] The first generation module is configured to generate, based on a first potential feature of a generative adversarial network, a first three-dimensional virtual object by using the generative adversarial network. The generative adversarial network is trained by using a sample image set. Each sample image in the sample image set includes a second object that belongs to a same object category as the first object.
- [0014] The first rendering module is configured to render the first three-dimensional virtual object to a second image according to a first camera pose of the first image.
- [0015] The first determination module is configured to determine a first loss value based on the first image and the second image.
- [0016] The first updating module is configured to update, based on the first loss value, the first potential feature and the first camera pose to obtain a second potential feature and a second three-dimensional virtual object corresponding to the second potential feature. The second three-dimensional virtual object is generated based on the second potential feature by using the generative adversarial network.
- [0017] In still another aspect, an embodiment of this application provides a computer device. The computer device includes a memory and a processor. The memory stores a computer program executable on the processor. The processor, when executing the computer program, implements part of all of steps in the above method.
- [0018] In yet another aspect, an embodiment of this application provides a computer-readable storage medium. The computer-readable storage medium stores a computer program. Part or all of steps in the above method are implemented when the computer program is executed by a processor.
- [0019] In yet another aspect, an embodiment of this application provides a computer program. The computer program includes a computer-readable code. When the computer-readable code is operated in a computer device, a processor in the computer device executes and implements part or all of steps in the above method.
- [0020] In yet another aspect, an embodiment of this application provides a computer program product. The computer program product includes a non-transitory computer-readable storage medium storing a computer program. When the computer program is read and executed by a computer, part or all of steps in the above method are implemented.
- [0021] In the embodiments of this application, the first image including a first object is first acquired. The first three-dimensional virtual object is generated based on the first potential feature of the generative adversarial network by using the trained generative adversarial network. In this way, since the generative adversarial network is trained by

using the sample image set, each sample image in the sample image set including a second object that belongs to the same object category as the first object, the generative adversarial network can learn priori knowledge (such as three-dimensional semantics, object geometry and texture details) of the object category, so that the generative adversarial network can be used to implicitly standardize the generated first three-dimensional virtual object within a boundary of the object category, and effective guidance is provided for the generation of an object portion that is not observed in the first image. Therefore, the authenticity and fidelity of the generated first three-dimensional virtual object are standardized, the authenticity of the first three-dimensional virtual object at a new angle of view is improved, and the generalization for scenarios where the first object is uncommon and scenarios where the first object in the first image is occluded can be realized. Next, the first three-dimensional virtual object is rendered as the second image according to the first camera pose of the first image. The first loss value is determined based on the first image and the second image. Based on the first loss value, the first potential feature and the first camera pose are updated to obtain the second potential feature and the second three-dimensional virtual object corresponding to the second potential feature. In this way, on one hand, by rendering the first three-dimensional virtual object to the second image and updating the first potential feature based on the first loss value between the first image and the second image to obtain the second potential feature, the first three-dimensional virtual object generated based on the second potential feature in a two-dimensional space can be consistent with the first image, so that the determined second three-dimensional virtual object can faithfully reflect the features of the first object in the first image; on the other hand, by performing fine tuning on a camera pose used for rendering the second image during the updating of the first potential feature, the alignment of pixels between the rendered second image and the first image can be improved, so that the accuracy of the first loss value can be enhanced, and then the accuracy of the determined second potential feature and the second three-dimensional virtual object can be further enhanced.

[0022] It should be understood that, the above general description and the following detailed description are merely exemplary and explanatory, and do not limit the technical solutions of this application.

BRIEF DESCRIPTION OF THE DRAWINGS

[0023] Drawings are incorporated into the specification and constitute a part of the specification. The drawings illustrate embodiments in accordance with this application and serve to describe the technical solutions of this application together with the specification.

[0024] FIG. 1A is a schematic implementation flowchart of a method for generating a virtual object according to an embodiment of this application.

[0025] FIG. 1B is a schematic effect diagram of a method for generating a virtual object according to an embodiment of this application.

[0026] FIG. 2 is a schematic implementation flowchart of a method for generating a virtual object according to an embodiment of this application.

[0027] FIG. 3 is a schematic implementation flowchart of a method for generating a

virtual object according to an embodiment of this application.

[0028] FIG. 4 is a schematic implementation flowchart of a method for generating a virtual object according to an embodiment of this application.

[0029] FIG. 5 is a schematic diagram of a composition structure of an apparatus for generating a virtual object according to an embodiment of this application.

[0030] FIG. 6 is a schematic diagram of a hardware entity of a computer device according to an embodiment of this application.

DETAILED DESCRIPTION

[0031] For making the purposes, technical solutions and advantages of this application clearer, the technical solutions of this application will further be described below in combination with the drawings and embodiments in detail. The described embodiments should not be considered as limits to this application. All other embodiments obtained by those of ordinary skill in the art without creative work shall fall within the scope of protection of this application.

[0032] "Some embodiments" involved in the following descriptions describes a subset of all possible embodiments. However, it can be understood that "some embodiments" may be the same subset or different subsets of all the possible embodiments, and may be combined without conflicts.

[0033] Term "first/second/third" involved is only for distinguishing similar objects and does not represent a specific sequence of the objects. It can be understood that "first/second/third" may be interchanged to specific sequences or orders if allowed to implement the embodiments of this application described herein in sequences except the illustrated or described ones.

[0034] Unless otherwise defined, all technical and scientific terms in the specification have the same meaning as those skilled in the art, belonging to this application, usually understand. Terms used in the specification are only used for describing the purpose of this application, but not intended to limit this application.

[0035] An embodiment of this application provides a method for generating a virtual object. The method may be performed by a computer device. The computer device may include, but is not limited to, various types of terminals such as a notebook computer, a tablet computer, a desktop computer, a set-top box, or a mobile device (for example, a mobile phone, a portable music player, a personal digital assistant, a special message device, and a portable game device), or may be implemented as a server. The server may be an independent physical server, may be a server cluster formed by a plurality of physical servers or a distributed system, or may be a cloud server configured to provide basic cloud computing services such as a cloud service, a cloud database, cloud computing, a cloud function, cloud storage, a network service, cloud communication, a middleware service, a domain name service, a security service, a Content Delivery Network (CDN), and a big data and artificial intelligence platform.

[0036] The technical solutions in the embodiments of this application are clearly and completely described below with reference to the drawings in the embodiments of this application.

- [0037] FIG. 1A is a schematic implementation flowchart of a method for generating a virtual object according to an embodiment of this application. As shown in FIG. 1A, the method includes S101 to S105.
- [0038] At S101, a first image including a first object is acquired.
- [0039] Herein, the first object may be any proper object, and includes, but is not limited to, a human body, a bird, a vehicle, a cat, a dog, a tree, a house, and the like, and this embodiment of this application is not limited thereto.
- [0040] The first image may be any proper image including the first object. The first object included in the first image frame may have any proper postures. For example, in a case that the first object is the human body, the human body included in the first image may be at least one of a sideways human body, a tiptoe human body, a jumping human body, or a bowed human body. For another example, in a case that the first object is the bird, the bird included in the first image may be at least one of a standing bird or a bird with outstretched wings.
- [0041] During implementation, the first image may be acquired in real time by an image acquisition module of the computer device, or may be sent to the computer device after acquired by an image acquisition device other than the computer device, which is not limited in this embodiment of this application. In some implementations, the first image may be a monocular image acquired by a monocular camera.
- [0042] At S102, based on a first potential feature of a generative adversarial network, a first three-dimensional virtual object is generated by using the generative adversarial network. The generative adversarial network is trained by using a sample image set. Each sample image in the sample image set includes a second object that belongs to a same object category as the first object.
- [0043] Herein, the generative adversarial network may be trained in advance by using at least one sample image. Each sample image includes the second object that belongs to the same object category as the first object. For example, in a case that the object category of the first object is the bird, the second object may be any proper bird, and the generative adversarial network may be trained by using at least one sample image included any proper bird. During implementation, the generative adversarial network may be trained by using the sample image under the supervision of a two-dimensional space. Those skilled in the art may train the generative adversarial network by any proper training method according to actual situations, and this embodiment of this application is not limited thereto.
- [0044] The first potential feature may be any proper potential feature that is determined from a manifold of the trained generative adversarial network. The manifold of the generative adversarial network is a high-dimensional space locally having Euclidean space properties. Each three-dimensional virtual object may be represented as a vector (that is, the potential feature) in the high-dimensional space. The three-dimensional virtual object, that is, the first three-dimensional virtual object, corresponding to the first potential feature may be generated based on the first potential feature by using the trained generative adversarial network. During implementation, those skilled in the art may, according to the actual situations, determine the first potential feature from the manifold of the generative adversarial network by using proper manners, which is not limited herein. For example, the first potential feature may be

randomly determined from the manifold of the trained generative adversarial network, or may be extracted from the first image by using a pre-trained autocoder. In some implementations, the first three-dimensional virtual object may be generated based on the first potential feature by using a generative network in the trained generative adversarial network.

[0045] The first three-dimensional virtual object may be represented by means of any proper three-dimensional object model, and the three-dimensional object model includes, but is not limited to, a grid model, a point cloud model, and the like. During implementation, those skilled in the art may represent the first three-dimensional virtual object by using proper manners according to actual situations, and select the proper generative adversarial network according to the representation of the first three-dimensional virtual object. This embodiment of this application is not limited thereto.

[0046] In some implementations, the first three-dimensional virtual object may be represented by using the grid model (such as a triangular mesh and quadrilateral mesh). The grid model may include a geometric map for representing a three-dimensional structure of the first three-dimensional virtual object and a texture map for representing texture details of the first three-dimensional virtual object. In a case that the first three-dimensional virtual object is represented by using the grid model, the generative network in the generative adversarial network may include an image generation network configured to generate the geometric map and the texture map of the three-dimensional virtual object. By using the image generation network, the geometric map and the texture map of the first three-dimensional virtual object may be generated based on the first potential feature, so that the first three-dimensional virtual object is obtained. During implementation, those skilled in the art may select a proper image generation network for the generative adversarial network according to the actual situations, such as a convolutional grid (ConvMesh) network and a dual convolutional grid (DualConvMesh) network, which is not limited herein.

[0047] In some implementations, the first three-dimensional virtual object may be represented by using the point cloud model. The point cloud model may include a point cloud shape for representing the first three-dimensional virtual object. In a case that the first three-dimensional virtual object is represented by using the point cloud model, the generative network in the generative adversarial network may include a point cloud generation network configured to generate the point cloud shape of the three-dimensional virtual object. By using the point cloud generation network, the point cloud shape of the first three-dimensional virtual object may be generated based on the first potential feature, so that the first three-dimensional virtual object is obtained. During implementation, those skilled in the art may select a proper point cloud generation network for the generative adversarial network according to the actual situations, which is not limited herein.

[0048] In some implementations, in a case that the first three-dimensional virtual object is the grid model and the generative network is the image generation network, the generative adversarial network may include the autocoder, the generative network, and a determination network. The autocoder may be configured to extract a potential feature from the sample image. The generative network may be configured to generate the three-dimensional virtual object based on the potential feature. The determination network may be configured to respectively distinguish the geometric map and the texture map of the

three-dimensional virtual object with the baseline information of the sample image. During implementation, the determination network in the generative adversarial network may include, but is not limited to, a first determination network that is configured to distinguish the geometric map and the texture map of the three-dimensional virtual object generated by the generative network in a UV texture space and a second determination network that is configured to distinguish the geometric map and the texture map of the three-dimensional virtual object generated by the generative network in an image space. U and V in the UV texture space are texture map coordinates. U and V define position information of each point on the texture map. These points are related to the grid model of the three-dimensional virtual object. U and V are coordinates that accurately correspond each point on the texture map to a surface of the three-dimensional virtual object. In this way, the three-dimensional virtual object of the generative network may be determined in both UV texture space and the image space, so that the realism of the three-dimensional virtual object generated by the trained generative adversarial network may be further improved.

[0049] In some implementations, in a case that the determination network in the generative adversarial network includes the a first determination network corresponding to the UV texture space and the second determination network corresponding to the image space, during the training of the generative adversarial network, a learning process of the generative network may be supervised by means of a result that the first determination network determines the three-dimensional virtual object generated by the generative network in the UV texture space and a result that the second determination network determines the three-dimensional virtual object generated by the generative network in the image space. A learning process of the first determination network may be supervised by means of a result that the first determination network determines pseudo baseline information of the UV texture space and a result that the first determination network determines the three-dimensional virtual object generated by the generative network in the UV texture space. A learning process of the second determination network may be supervised by means of a result that the second determination network determines real baseline information of the image space and a result that the second determination network determines the three-dimensional virtual object generated by the generative network in the image space. For example, learning processes of the generative network $G(\square)$, the first determination network $D_{uv}(\square)$, and the second determination network $D_I(\square)$ may be supervised by using a least squares loss function shown in a formula 1 during the training of the generative network.

$$\begin{cases} L_G = \lambda_{uv} E_{z \sim P_z(z)} [(D_{uv}(G(z)) - 1)^2] + \lambda_I E_{z \sim P_z(z)} [(D_I(R(G(z), \pi)) - 1)^2] \\ L_{D_{uv}} = E_{S, T \sim P_{pseudo}} [(D_{uv}(S, T) - 1)^2] + E_{z \sim P_z(z)} [(D_{uv}(G(z)))^2] \\ L_{D_I} = E_{I \sim P_{data}} [(D_I(I) - 1)^2] + E_{z \sim P_z(z)} [(D_I(R(G(z), \pi)))^2] \end{cases} \quad (1)$$

where, L_G is a loss value of the generative network $G(\square)$; $P_z(z)$ is probability distribution of a potential vector z ; π is a camera pose of a sample image; $R(\square)$ is a renderer; λ_{uv} is the weight of the first determination network, λ_I is the weight of the

second determination network, and λ_{uv} and λ_I may be set according to an actual application scenario during implementation; $L_{D_{uv}}$ is a loss value of the first determination network $D_{uv}(\square)$; P_{pseudo} is the probability distribution of a pseudo baseline geometric map S and a pseudo baseline texture map T ; L_{D_I} is a loss value of the first determination network $D_I(\square)$; and P_{data} is the probability distribution of a baseline image I .

[0050] At S103, the first three-dimensional virtual object is rendered as a second image according to a first camera pose of the first image.

[0051] Herein, the first camera pose of the first image may be preset, may be obtained by predicting the camera pose of the first image by using a pre-trained first neural network model, or may be obtained by estimating the camera pose of the first image by using a camera pose estimator, which is not limited herein. The first camera pose may characterize a position and posture of a camera of the first image in a three-dimensional space, and may include a coordinate and a rotation angle of the camera in a coordinate system of the three-dimensional space.

[0052] The first three-dimensional virtual object may be rendered as the two-dimensional second image according to the first camera pose of the first image. During implementation, any proper renderer may be used to perform image rendering on the first three-dimensional virtual object to obtain the second image, and this embodiment of this application is not limited thereto. For example, an Interpolation-based Differentiable Renderer (DIB-R), a Soft Rasterizer (SoftRas), and the like may be used.

[0053] At S104, a first loss value is determined based on the first image and the second image.

[0054] Herein, the first image and the second image may be constrained by using a proper constraint manner according to an actual situation. A loss value between the first image and the second image is determined, that is, the first loss value, and this embodiment of this application is not limited thereto.

[0055] In some implementations, a distance between the first image and the second image may be determined as the first loss value. During implementation, the distance between the first image and the second image may include, but is not limited to, a cosine distance, a Euclidean distance, a chamfer distance, and the like.

[0056] In some implementations, feature extraction may be respectively performed on the first image and the second image to obtain a first feature map and a second feature map, and a distance between the first feature map and the second feature map is determined as the first loss value.

[0057] In some implementations, foreground extraction may be respectively performed on the first image and the second image to obtain a first pixel point set and a second pixel point set, and a distance between the first pixel point set and the second pixel point set is determined as the first loss value.

[0058] At S105, based on the first loss value, the first potential feature and the first camera pose are updated to obtain a second potential feature and a second three-dimensional virtual object corresponding to the second potential feature. The second

three-dimensional virtual object is generated based on the second potential feature by using the generative adversarial network.

[0059] Herein, a manner for updating the first potential feature and the first camera pose may be determined according to actual situations, and may include, but not limited to, at least one of a gradient descent method or a Newton momentum method, which is not limited herein.

[0060] During implementation, the first potential feature and the first camera pose may be updated at least once. After the first potential feature and the first camera pose are updated each time, the updated first three-dimensional virtual object may be generated based on the updated first potential feature by using the generative adversarial network. Based on the updated first camera pose and the updated first three-dimensional virtual object, the updated second image is determined. The updated first loss value is determined based on the first image and the updated second image. Based on the updated first loss value, whether next updating is continuously performed on the first potential feature and the first camera pose may be determined.

[0061] In some implementations, in a case that the first loss value satisfies a set loss condition, or the number of updates performed on the first potential feature and the first camera pose is less than a set number threshold, the next updating performed on the first potential feature and the first camera pose may be stopped. Then, the updated first potential feature is determined as a second potential feature, and the updated first three-dimensional virtual object is determined as a second three-dimensional virtual object. Herein, the preset condition may include, but is not limited to, the first loss value being less than a preset loss threshold, change convergence between the first loss values updated each time, and the like, which is not limited herein.

[0062] In some implementations, in a case that the loss value does not satisfy the preset condition, the next updating performed on the first potential feature and the first camera pose may be continued.

[0063] In some implementations, the second potential feature is a potential feature that the corresponding three-dimensional virtual object in the manifold of the trained generative adversarial network is the most similar to the first object in the first image, that is, the first potential feature that causes the first loss value between the first image and the second image to be minimum. During implementation, a target for updating the first potential feature and the first camera pose may be shown in the following formula 2:

$$z^*, \pi^* = \arg \min_{z, \pi} L_{inv}(R(G(z), \pi), I_{in}) \quad (2)$$

where, $G(z)$ is the first three-dimensional virtual object that is generated based on the first potential feature z by using the generative adversarial network; $R(G(z), \pi)$ is the second image that is obtained by rendering the first three-dimensional virtual object $G(z)$ according to the first camera pose π ; $L_{inv}(R(G(z), \pi), I_{in})$ is the first loss value that is determined based on the first image I_{in} and the second image $R(G(z), \pi)$; z^* and π^* are the first potential feature (that is, the second potential feature) and the first camera pose that cause the first loss value $L_{inv}(R(G(z), \pi), I_{in})$ to be minimum; and the

second three-dimensional virtual object corresponding to the second potential feature may be $G(z^*)$.

[0064] In the embodiments of this application, the first image including a first object is first acquired. The first three-dimensional virtual object is generated based on the first potential feature of the generative adversarial network by using the trained generative adversarial network. In this way, since the generative adversarial network is trained by using at least one sample image, each sample image includes the second object that belongs to the same object category as the first object. Therefore, the generative adversarial network may learn prior knowledge (such as three-dimensional semantics, object geometry and texture details) of the object category, so that the generative adversarial network may be used to implicitly standardize the generated first three-dimensional virtual object within a boundary of the object category, and effective guidance is provided for the generation of an object portion that is not observed in the first image. Therefore, the authenticity and fidelity of the generated first three-dimensional virtual object are standardized, the authenticity of the first three-dimensional virtual object at a new angle of view is improved, and scenarios where the first object is uncommon and scenarios where the first object in the first image is occluded can be generalized. Next, the first three-dimensional virtual object is rendered as the second image according to the first camera pose of the first image. The first loss value is determined based on the first image and the second image. Based on the first loss value, the first potential feature and the first camera pose are updated to obtain the second potential feature and the second three-dimensional virtual object corresponding to the second potential feature. In this way, in an aspect, by rendering the first three-dimensional virtual object to the second image and updating the first potential feature based on the first loss value between the first image and the second image to obtain the second potential feature, the first three-dimensional virtual object generated based on the second potential feature in a two-dimensional space may be consistent with the first image, so that the determined second three-dimensional virtual object may faithfully reflect the features of the first object in the first image. In another aspect, by performing fine tuning on a camera pose used for rendering the second image during the updating of the first potential feature, the alignment of pixels between the rendered second image and the first image may be improved, so that the accuracy of the first loss value may be enhanced. Therefore, the accuracy of the determined second potential feature and the second three-dimensional virtual object may be further enhanced.

[0065] In some embodiments, the second three-dimensional virtual object has a second geometric map and a second texture map. The above method for generating a virtual object may further include S111 to S112.

[0066] At S111, a second grid is determined based on the second geometric map and a set second grid template.

[0067] Herein, the second grid template may be a preset template which is configured to define a fixed connection surface at a vertex of a grid model. During implementation, those skilled in the art may preset the proper second grid template according to the actual situations, which is not limited herein. In some implementations, the second grid template may be a dipolar sphere template. In some implementations, the second grid template may be a grid template that is obtained by deforming the dipolar sphere template.

- [0068] The second three-dimensional virtual object may have the second geometric map and the second texture map. The second geometric map may characterize the deformation of the second grid template corresponding to the second three-dimensional virtual object. The second texture map may characterize the texture of a surface of the second three-dimensional virtual object.
- [0069] During implementation, corresponding deformation may be performed on the second grid template according to the second geometric map, to obtain a deformed second grid template, that is, the second grid.
- [0070] In some implementations, the second geometric map may include an offset to a position of each vertex in the second grid template. According to the offset to the position of each vertex in the second grid template, the second grid may be obtained by offsetting the position of the corresponding vertex in the second grid template.
- [0071] At S112, image rendering is performed based on the second texture map and the second grid according to a set second camera pose, to obtain a rendered third image.
- [0072] Herein, the second camera pose may be a camera pose that is preset and corresponds to a specific observation angle of view, may characterize a position and posture of a camera for observing the second three-dimensional virtual object in the three-dimensional space, and may include a coordinate and a rotation angle of the camera in the coordinate system of the three-dimensional space. Those skilled in the art may set the proper second camera pose according to actual requirements, which is not limited in this embodiment of this application.
- [0073] The second grid may characterize a three-dimensional structure of the second three-dimensional virtual object. The second texture map may characterize the texture details of the second three-dimensional virtual object. The second camera pose may characterize the position and posture of the camera for observing the second three-dimensional virtual object in the three-dimensional space. Therefore, according to the second camera pose, the second three-dimensional virtual object may be rendered as a two-dimensional space based on the second texture map and the second grid, to obtain a rendered third image.
- [0074] During implementation, by using any proper renderer, image rendering may be performed based on the second texture map and the second grid according to the set second camera pose, to obtain the rendered third image, which is not limited herein. For example, renderers such as DIB-R and SoftRas may be used.
- [0075] In this embodiment of this application, the second grid characterizing the three-dimensional structure of the second three-dimensional virtual object is determined based on the second geometric map of the second three-dimensional virtual object and the set second grid template. According to the set second camera pose, image rendering is performed based on the second texture map of the second three-dimensional virtual object and the second grid, to obtain the rendered third image. In this way, the rendering of the second three-dimensional virtual object at the free angle of view may be supported, to obtain the third image of the second three-dimensional virtual object at an angle of view corresponding to the set second camera pose. Since the second geometric map and the second texture map of the second three-dimensional virtual object are generated by using the generative adversarial network that has learned prior knowledge of the object category to which the first object belongs, the prior knowledge can provide effect

guidance for the generation of an object portion that is not observed in the first image. Therefore, the authenticity of the second three-dimensional virtual object at a new angle of view may be improved, that is to say, the authenticity of the third image may be improved.

- [0076] In some embodiments, the above method for generating a virtual object may further include S121 to S126.
- [0077] At S121, a fourth image including a third object is acquired, where the third object and the first object belong to a same object category.
- [0078] At S122, based on a third potential feature, a third three-dimensional virtual object is generated by using the generative adversarial network.
- [0079] At S123, the third three-dimensional virtual object is rendered as a fifth image according to a third camera pose of the fourth image.
- [0080] At S124, a second loss value is determined based on the fourth image and the fifth image.
- [0081] At S125, based on the second loss value, the third potential feature and the third camera pose are updated to obtain a fourth potential feature and a fourth three-dimensional virtual object corresponding to the fourth potential feature, where the fourth three-dimensional virtual object is generated based on the fourth potential feature by using the generative adversarial network.
- [0082] Herein, the third object may be any proper object that belongs to a same object category as the first object, and this embodiment of this application is not limited thereto.
- [0083] S121 to S125 respectively correspond to S101 to S105, which may refer to specific implementations of S101 to S105 during implementation.
- [0084] At S126, texture migration is performed between the second three-dimensional virtual object and the fourth three-dimensional virtual object, to obtain a fifth three-dimensional virtual object and a sixth three-dimensional virtual object.
- [0085] Herein, the second three-dimensional virtual object may have the second geometric map and the second texture map. The fourth three-dimensional virtual object may have a fourth geometric map and a fourth texture map.
- [0086] The performing of texture migration between the second three-dimensional virtual object and the fourth three-dimensional virtual object refers to exchanging between the texture maps of the second three-dimensional virtual object and the fourth three-dimensional virtual object. That is to say, the second texture map of the second three-dimensional virtual object is replaced with the fourth texture map, and the fourth texture map of the fourth three-dimensional virtual object is replaced with the second texture map. In this way, the obtained fifth three-dimensional virtual object may have the second geometric map and the fourth texture map. The sixth three-dimensional virtual object may have the fourth geometric map and the second texture map.
- [0087] FIG. 1B is a schematic effect diagram of a method for generating a virtual object according to an embodiment of this application. As shown in FIG. 1B, the first image 10 includes the first object 11, and the second image 20 includes the second object 21. By means of the method for generating a virtual object provided in this embodiment of this application, the second three-dimensional virtual object 12 corresponding to the

first object 11, and the fourth three-dimensional virtual object 22 corresponding to the second object 21 may be obtained. Texture migration is performed between the second three-dimensional virtual object 12 and the fourth three-dimensional virtual object 22, to obtain the fifth three-dimensional virtual object 13 and the sixth three-dimensional virtual object 23.

[0088] In the above embodiment, since the geometric maps and the texture maps of the plurality of three-dimensional virtual objects generated by using the same generative adversarial network are topologically and semantically aligned, both the fifth three-dimensional virtual object and the sixth three-dimensional virtual object that are obtained after texture migration is performed between the second three-dimensional virtual object and the fourth three-dimensional virtual object generated by using the generative adversarial network can have desirable authenticity and fidelity.

[0089] FIG. 2 is a schematic implementation flowchart of a method for generating a virtual object according to an embodiment of this application. As shown in FIG. 2, the method includes S201 to S206.

[0090] At S201, a first image including a first object is acquired.

[0091] At S202, based on a first potential feature of a generative adversarial network, a first three-dimensional virtual object is generated by using the generative adversarial network. The generative adversarial network is trained by using a sample image set. Each sample image in the sample image set includes a second object that belongs to a same object category as the first object.

[0092] At S203, the first three-dimensional virtual object is rendered as a second image according to a first camera pose of the first image.

[0093] At S204, a first loss value is determined based on the first image and the second image.

[0094] Herein, S201 to S204 respectively correspond to S101 to S104, which may refer to specific implementations of S101 to S104 during implementation.

[0095] At S205, in a case that the first loss value does not satisfy the set loss condition, the first potential feature and the first camera pose are respectively updated, to obtain the updated first potential feature and the updated first camera pose.

[0096] Herein, the loss condition may include, but is not limited to, at least one of the first loss value being less than a preset loss threshold, change convergence between the first loss values updated each time, or the like. During implementation, those skilled in the art may set the appropriate loss condition according to actual situations, which is not limited herein.

[0097] At S206, based on the updated first potential feature and the updated first camera pose, the second potential feature and the second three-dimensional virtual object corresponding to the second potential feature are determined. The second three-dimensional virtual object is generated based on the second potential feature by using the generative adversarial network.

[0098] Herein, the updated first three-dimensional virtual object may be generated based on the updated first potential feature by using the generative adversarial network. Based on the updated first camera pose and the updated first three-dimensional virtual

object, the updated second image is determined. The updated first loss value is determined based on the first image and the updated second image. Based on the updated first loss value, whether next updating is continuously performed on the first potential feature and the first camera pose may be determined.

[0099] In some implementations, in a case that the updated first loss value satisfies a set loss condition, or the number of updates performed on the first potential feature and the first camera pose is less than a set number threshold, the next updating performed on the first potential feature and the first camera pose may be stopped. Then, the updated first potential feature is determined as a second potential feature, and the updated first three-dimensional virtual object is determined as a second three-dimensional virtual object.

[00100] In some embodiments, S206 may include S211 to S214.

[00101] At S211, based on the updated first potential feature, an updated first three-dimensional virtual object is generated by using the generative adversarial network.

[00102] At S212, an updated second image is determined based on the updated first camera pose and the updated first three-dimensional virtual object.

[00103] At S213, the updated first loss value is determined based on the first image and the updated second image.

[00104] Herein, S211 to S214 respectively correspond to S202 to S204, which may refer to specific implementations of S202 to S204 during implementation.

[00105] At S214, in a case that the updated first loss value satisfies the set loss condition, the updated first potential feature is determined as the second potential feature, and the updated first three-dimensional virtual object is determined as the second three-dimensional virtual object.

[00106] In this embodiment of this application, in a case that the first loss value does not satisfy the set loss condition, the first potential feature and the first camera pose are respectively updated, to obtain the updated first potential feature and the updated first camera pose. Based on the updated first potential feature and the updated first camera pose, the second potential feature and the second three-dimensional virtual object corresponding to the second potential feature are determined. Herein, by continuously updating the first potential feature and the first camera pose, the accuracy of the second potential feature may be further enhanced, so that the authenticity and fidelity of the second three-dimensional virtual object corresponding to the second potential feature may be improved.

[00107] FIG. 3 is a schematic implementation flowchart of a method for generating a virtual object according to an embodiment of this application. As shown in FIG. 3, the method includes S301 to S306.

[00108] At S301, a first image including a first object is acquired.

[00109] At S302, based on a first potential feature of a generative adversarial network, a first three-dimensional virtual object is generated by using the generative adversarial network. The generative adversarial network is trained by using a sample image set. Each sample image in the sample image set includes a second object that belongs to a same object category as the first object.

- [00110] Herein, S301 to S302 respectively correspond to S101 to S102, which may refer to specific implementations of S101 to S102 during implementation.
- [00111] At S303, a first grid is determined based on the first geometric map of the first three-dimensional virtual object and a set first grid template.
- [00112] Herein, the first grid template may be a preset template which is configured to define a fixed connection surface at a vertex of a grid model. During implementation, those skilled in the art may preset the proper first grid template according to the actual situations, which is not limited herein. In some implementations, the first grid template may be a dipolar sphere template. In some implementations, the first grid template may be a grid template that is obtained by deforming the dipolar sphere template.
- [00113] The first three-dimensional virtual object may have the first geometric map and the first texture map. The first geometric map may characterize the deformation of the first grid template corresponding to the first three-dimensional virtual object. The first texture map may characterize the texture of a surface of the first three-dimensional virtual object.
- [00114] During implementation, corresponding deformation may be performed on the first grid template according to the first geometric map, to obtain a deformed first grid template, that is, the first grid.
- [00115] In some implementations, the first geometric map may include an offset to a position of each vertex in the first grid template. According to the offset to the position of each vertex in the first grid template, the first grid may be obtained by offsetting the position of the corresponding vertex in the first grid template.
- [00116] At S304, image rendering is performed based on the first texture map of the first three-dimensional virtual object and the first grid according to the first camera pose, to obtain a rendered second image.
- [00117] Herein, the first grid may characterize a three-dimensional structure of the first three-dimensional virtual object. The first texture map may characterize the texture details of the first three-dimensional virtual object. The first camera pose may characterize the position and posture of the camera for collecting the first three-dimensional virtual object in the three-dimensional space. Therefore, according to the first camera pose, the first three-dimensional virtual object may be rendered as a two-dimensional space based on the first texture map and the first grid, to obtain a rendered second image.
- [00118] During implementation, by using any proper renderer, image rendering may be performed based on the first texture map and the first grid according to the set first camera pose, to obtain the rendered second image, which is not limited herein. For example, renderers such as DIB-R and SoftRas may be used.
- [00119] At S305, the first loss value is determined based on the first image and the second image.
- [00120] At S306, based on the first loss value, the first potential feature and the first camera pose are updated to obtain the second potential feature and the second three-dimensional virtual object corresponding to the second potential feature. The second three-dimensional virtual object is generated based on the second potential feature by

using the generative adversarial network.

[00121] Herein, S305 to S306 respectively correspond to S104 to step S105, which may refer to specific implementations of S104 to S105 during implementation.

[00122] In this embodiment of this application, the first grid is determined based on the first geometric map of the first three-dimensional virtual object and the set first grid template. According to the first camera pose of the first image, image rendering is performed based on the first texture map of the first three-dimensional virtual object and the first grid, to obtain the rendered second image. In this way, the second image corresponding to a same camera pose as the first image may be obtained, so that the more accurate first loss value may be obtained. Therefore, the more accurate second potential feature and the more realistic second three-dimensional virtual object may be determined.

[00123] In some embodiments, the first loss value may include a texture loss value. S305 may include S311.

[00124] At S311, the texture loss value is determined based on the first image and the second image.

[00125] Herein, the texture loss value may characterize a loss between a texture of the first object in the first image and a texture of the first three-dimensional virtual object in the second image. During implementation, those skilled in the art may adopt a proper manner to determine the texture loss value based on the first image and the second image according to actual situations, and this embodiment of this application is not limited thereto.

[00126] In some implementations, the first object in the first image may be segmented to obtain a first sub-image including the first object. The texture loss value is determined based on the first sub-image and the second image. For example, a distance between the first sub-image and the second image may be determined as the first loss value, or a mean square error between a pixel value of each pixel point in the first sub-image and a pixel value of each pixel point in the second image may be determined as the texture loss value. Herein, the distance between the first sub-image and the second image may include, but is not limited to, a cosine distance, a Euclidean distance, and the like.

[00127] In some implementations, at least one first key point of the first object may be extracted from the first image. At least one second key point of the first three-dimensional virtual object may be extracted from the second image. The texture loss value is determined based on the at least one first key point and the at least one second key point. For example, the sum or mean value of differences between a pixel value of each first key point and a pixel value of the second key point corresponding to the first key point may be determined as the texture loss value. Alternatively, the sum or mean value between a coordinate value of each first key point in the first image and a coordinate value of the second key point corresponding to the first key point in the second image may be determined as the texture loss value.

[00128] In some implementations, feature extraction may further be performed on the first image and the second image, to obtain a first feature map of the first image and a second feature map of the second image. The at least one first key point of the first object is extracted from the first image. The at least one second key point of the first three-dimensional virtual object is extracted from the second image. The texture loss value is

determined based on a feature value corresponding to the at least one first key point in the first feature map and a feature value corresponding to the at least one second key point in the second feature map. For example, the sum or mean value of differences between the feature value corresponding to each first key point in the first feature map and the feature value corresponding to the second key point corresponding to the first key point in the second feature map may be determined as the texture loss value.

- [00129] In some embodiments, the texture loss value includes a first texture loss value. S311 may include S321 to S322.
- [00130] At S321, foreground extraction is performed on the first image and the second image to obtain a first pixel point set and a second pixel point set.
- [00131] Herein, foreground extraction may be performed on the first image and the second image by using any proper foreground extraction algorithm according to the actual situations, and this embodiment of this application is not limited thereto. For example, a Poisson foreground extraction algorithm, a grabcut algorithm, and the like may be used.
- [00132] The first pixel point set may be obtained by performing foreground extraction on the first image. The first pixel point set may include at least one foreground pixel point in the first image. Since the first image includes the first object, the first object is a foreground of the first image, so that the at least one foreground pixel point obtained by performing foreground extraction on the first image is a pixel point corresponding to the first object.
- [00133] The second pixel point set may be obtained by performing foreground extraction on the second image. The second pixel point set may include at least one foreground pixel point in the second image. Since the second image includes the first three-dimensional virtual object, the first three-dimensional virtual object is a foreground of the second image, so that the at least one foreground pixel point obtained by performing foreground extraction on the second image is a pixel point corresponding to the first three-dimensional virtual object.
- [00134] At S322, the first texture loss value is determined based on the first pixel point set and the second pixel point set.
- [00135] Herein, the first texture loss value may characterize a loss between each pixel point in the first pixel point set and each pixel point in the second pixel point set, that is, the texture loss between the first object in the first image and the first three-dimensional virtual object in the second image. During implementation, those skilled in the art may use a proper constraint manner to constrain the first pixel point set and the second pixel point set according to the actual situations, so as to determine the loss between each pixel point in the first pixel point set and each pixel point in the second pixel point set, that is, the first texture loss value. This embodiment of this application is not limited thereto.
- [00136] In some implementations, the first texture loss value may include a first pixel point distance. A distance between a pixel value of each pixel point in the first pixel point set and a pixel value of each pixel point in the second pixel point set may be determined as the first pixel point distance. During implementation, the distance between a pixel value of each pixel point in the first pixel point set and a pixel value of each pixel point in the second pixel point set may include, but is not limited to, a cosine distance, a

Euclidean distance, a chamfer distance, and the like.

- [00137] In some implementations, the first texture loss value may include a second pixel point distance. A distance between a coordinate value of each pixel point in the first pixel point set and a coordinate value of each pixel point in the second pixel point set may be determined as the second pixel point distance. During implementation, the distance between a coordinate value of each pixel point in the first pixel point set and a coordinate value of each pixel point in the second pixel point set may include, but is not limited to, a cosine distance, a Euclidean distance, a chamfer distance, and the like.
- [00138] In the above embodiment, by performing foreground extraction on the first image, the first pixel point set including at least one pixel point of the first object in the first image may be obtained. By performing foreground extraction on the second image, the second pixel point set including at least one pixel point of the first three-dimensional virtual object in the second image. In this way, the first texture loss value determined based on the first pixel point set and the second pixel point set may better characterize the texture loss between the first object in the first image and the first three-dimensional virtual object in the second image. Therefore, by supervising the first texture loss value, the more accurate second potential feature and the more realistic second three-dimensional virtual object may be determined.
- [00139] In some embodiments, S322 may include S331 to S333.
- [00140] At S331, for each first pixel point in the first pixel point set, a first distance between the first pixel point and each second pixel point in the second pixel point set is determined, and the minimum first distance is determined as a first pixel loss value of the first pixel point.
- [00141] Herein, the first distance between the first pixel point and the second pixel point may include, but is not limited to, at least a distance between the first pixel point and the second pixel point in space, a distance in appearance, or the like. The distance between the first pixel point and the second pixel point in space may include a distance between a coordinate value of the first pixel point and a coordinate value of the second pixel point. The distance between the first pixel point and the second pixel point in appearance may include a distance between a pixel value of the first pixel point and a pixel value of the second pixel point. During implementation, the distance between the coordinate value of the first pixel point and the coordinate value of the second pixel point may include, but is not limited to, a cosine distance, a Euclidean distance, and the like. The distance between the pixel value of the first pixel point and the pixel value of the second pixel point may include, but is not limited to, a cosine distance, a Euclidean distance, and the like.
- [00142] At S332, for each second pixel point in the second pixel point set, a second distance between the second pixel point and each first pixel point in the first pixel point set is determined, and the minimum second distance is determined as a second pixel loss value of the second pixel point.
- [00143] Herein, the second distance between the second pixel point and the first pixel point may include, but is not limited to, at least a distance between the second pixel point and the first pixel point in space, a distance in appearance, or the like. The distance between the second pixel point and the first pixel point in space may include a distance between a coordinate value of the second pixel point and a coordinate value of the first

pixel point. The distance between the second pixel point and the first pixel point in appearance may include a distance between a pixel value of the second pixel point and a pixel value of the first pixel point.

[00144] At S333, the first texture loss value is determined based on each first pixel loss value and each second pixel loss value.

[00145] Herein, a proper manner may be used to determine the first texture loss value based on each first pixel loss value and each second pixel loss value according to the actual situations, and this embodiment of this application is not limited thereto.

[00146] In some implementations, the sum of each first pixel loss value and each second pixel loss value may be determined as the first texture loss value.

[00147] In some implementations, the mean value of each first pixel loss value and each second pixel loss value may be determined as the first texture loss value.

[00148] In some implementations, the sum of the mean value of each first pixel loss value and the mean value of each second pixel loss value may be determined as the first texture loss value. For example, the first texture loss value may be calculated by using a manner shown in the following formula 3:

$$L_{CD}(S_1, S_2) = \frac{1}{|S_1|} \sum_{x \in S_1} \min_{y \in S_2} D_{xy} + \frac{1}{|S_2|} \sum_{y \in S_2} \min_{x \in S_1} D_{yx} \quad (3)$$

where, S_1 is the first pixel point set; x is the first pixel point in S_1 ; S_2 is the second pixel point set; y is the second pixel point in S_2 ; D_{xy} is the first distance between the first pixel point x and the second pixel point y ; D_{yx} is the second distance between the second pixel point y and the first pixel point x ; $\min_{y \in S_2} D_{xy}$ is the first pixel loss value of the first pixel point x ; $\min_{x \in S_1} D_{yx}$ is the second pixel loss value of the second pixel point y ;

and by calculating the sum of the mean value $\frac{1}{|S_1|} \sum_{x \in S_1} \min_{y \in S_2} D_{xy}$ of each first pixel loss value and the mean value $\frac{1}{|S_2|} \sum_{y \in S_2} \min_{x \in S_1} D_{yx}$ of each second pixel loss value, the first texture loss value $L_{CD}(S_1, S_2)$ may be obtained.

[00149] In the above embodiment, for each first pixel point in the first pixel point set, the first distance between the first pixel point and each second pixel point in the second pixel point set is determined, and the minimum first distance is determined as the first pixel loss value of the first pixel point. For each second pixel point in the second pixel point set, the second distance between the second pixel point and each first pixel point in the first pixel point set is determined, and the minimum second distance is determined as the second pixel loss value of the second pixel point. The first texture loss value is determined based on each first pixel loss value and each second pixel loss value. In this way, the loss between the first pixel point set and the second pixel point set may be considered from an overall angle. That is to say, the texture loss may be calculated by

searching matched pixel points in local areas respectively corresponding to the first pixel point set and the second pixel point set, so that texture dislocation due to the non-alignment of the pixel points in the first image and the second image is reduced. Therefore, the more accurate second potential feature and the more realistic second three-dimensional virtual object may be determined.

[00150] In some embodiments, S331 of determining the first distance between the first pixel point and each second pixel point in the second pixel point set may include S341.

[00151] At S341, for each second pixel point in the second pixel point set, the first distance between the first pixel point and the second pixel point is determined based on the distance between the pixel value of the first pixel point and the pixel value of the second pixel point and the distance between the coordinate value of the first pixel point and the coordinate value of the second pixel point.

[00152] Herein, the pixel value of the pixel point may be an RGB value, or may be a grayscale value, which is not limited herein. The distance between the pixel value of the first pixel point and the pixel value of the second pixel point may include, but is not limited to, at least one of a cosine distance, a Euclidean distance, or the like.

[00153] The coordinate value of the first pixel point is a coordinate value of the first pixel point in the first image, and the coordinate value of the second pixel point is a coordinate value of the second pixel point in the second image. The distance between the coordinate value of the first pixel point and the coordinate value of the second pixel point may include, but is not limited to, at least one of a cosine distance, a Euclidean distance, or the like.

[00154] During implementation, the first distance between the first pixel point and the second pixel point may be determined based on the distance between the pixel value of the first pixel point and the pixel value of the second pixel point and the distance between the coordinate value of the first pixel point and the coordinate value of the second pixel point by using any proper manner, and this embodiment of this application is not limited thereto.

[00155] In some implementations, the first distance between the first pixel point and the second pixel point may be obtained by summing the distance between the pixel value of the first pixel point and the pixel value of the second pixel point and the distance between the coordinate value of the first pixel point and the coordinate value of the second pixel point.

[00156] In some implementations, the first distance between the first pixel point and the second pixel point may be obtained by performing quadrature on the distance between the pixel value of the first pixel point and the pixel value of the second pixel point and the distance between the coordinate value of the first pixel point and the coordinate value of the second pixel point. For example, the first distance between the first pixel point and the second pixel point may be determined in a manner shown in the following formula 4:

$$D_1 = \max((D_1^s + \dot{o}_{s1})^{\alpha_1}, 1) \times (D_1^a + \dot{o}_{a1}) \quad (4)$$

where, the first distance D_1 between the first pixel point and the second pixel point may

be obtained by multiplying a first factor $\max((D_1^s + \dot{o}_{s1})^{\alpha 1}, 1)$ and a second factor $(D_1^a + \dot{o}_{a1})$; D_1^s is the distance between the coordinate value of the first pixel point and the coordinate value of the second pixel point; D_1^a is the distance between the pixel value of the first pixel point and the pixel value of the second pixel point; $\alpha 1$ is a set scale factor; $\dot{o}_{s1}$ and $\dot{o}_{a1}$ are preset residual terms greater than 0; $\dot{o}_{s1}$ and $\dot{o}_{a1}$ may ensure that the first distance D_1 between the first pixel point and the second pixel point is not 0; the first factor $\max((D_1^s + \dot{o}_{s1})^{\alpha 1}, 1)$ may be used as the weight of the second factor $(D_1^a + \dot{o}_{a1})$, and may set $\dot{o}_{s1}$ to be less than 1, to cause $(D_1^s + \dot{o}_{s1})^{\alpha 1}$ to be less than 1 in a case that the degree of the non-alignment of the first pixel point and the second pixel point in space is relatively small, so that the first factor $\max((D_1^s + \dot{o}_{s1})^{\alpha 1}, 1)$ may be maintained at 1. In this case, the offset of the first pixel point and the second pixel point in space is not required to be considered, so that local dislocation between the first three-dimensional virtual object in the generated second image and the first object in the first image is tolerable. Therefore, the second three-dimensional virtual object that is more semantic may be obtained.

[00157] The above S332 in which the second distance between the second pixel point and each first pixel point in the first pixel point set is determined may include S342.

[00158] At S342, for each first pixel point in the first pixel point set, the second distance between the second pixel point and the first pixel point is determined based on the distance between the pixel value of the second pixel point and the pixel value of the first pixel point and the distance between the coordinate value of the second pixel point and the coordinate value of the first pixel point.

[00159] Herein, a process of determining the second distance between the second pixel point and the first pixel point in S342 corresponds to a process of determining the first distance between the first pixel point and the second pixel point in S341, which may refer to a specific implementation in S341 during implementation.

[00160] In the above embodiment, since the distance between the pixel value of the first pixel point and the pixel value of the second pixel point may reflect the loss of the first pixel point and the second pixel point in appearance, and the distance between the coordinate value of the first pixel point and the coordinate value of the second pixel point may reflect the loss of the first pixel point and the second pixel point in space, the first distance between the first pixel point and the second pixel point is determined based on the distance between the pixel value of the first pixel point and the pixel value of the second pixel point and the distance between the coordinate value of the first pixel point and the coordinate value of the second pixel point; and the second distance between the second pixel point and the first pixel point is determined based on the distance between the pixel value of the second pixel point and the pixel value of the first pixel point and the distance between the coordinate value of the second pixel point and the coordinate value of the first pixel point. In this way, the loss of the first pixel point and the second pixel point in appearance and in space, so that the robustness of the first distance between the first pixel point and the second pixel point and the second distance between the second pixel point and the first pixel point may be improved. Therefore, the accuracy of

the first texture loss value may be enhanced, so as to obtain the more accurate second potential feature and the more realistic second three-dimensional virtual object.

[00161] In some embodiments, the texture loss value further includes a second texture loss value. S311 may further include S351 to S352.

[00162] At S351, feature extraction is performed on the first image and the second image to obtain a first feature map and a second feature map.

[00163] Herein, the first feature map of the first image may be obtained by performing feature extraction on the first image. The pixel points in the first feature map are in one-to-one correspondence with the pixel points in the first image. Each pixel point in the first feature map has a corresponding feature value.

[00164] The second feature map of the second image may be obtained by performing feature extraction on the second image. The pixel points in the second feature map are in one-to-one correspondence with the pixel points in the second image. Each pixel point in the second feature map has a corresponding feature value.

[00165] During implementation, those skilled in the art may respectively perform feature extraction on the first image and the second image by using any proper feature extraction algorithm according to the actual situations, which is not limited herein. For example, feature extraction may be respectively performed on the first image and the second image by using a pre-trained second neural network model, to obtain the first feature map corresponding to the first image and the second feature map corresponding to the second image. Alternatively, each pixel point in the first image may be described by using a set feature descriptor, to obtain the feature value of each pixel point in the first image, so that the first feature map is obtained. Each pixel point in the second image is described by using the feature descriptor to obtain the feature value of each pixel point in the second image, so that the second feature map is obtained.

[00166] At S352, the second texture loss value is determined based on the first feature map and the second feature map.

[00167] Herein, the first feature map and the second feature map may be constrained by using a proper constraint manner according to the actual situation. The loss between the first feature map and the second feature map is determined, that is, the second texture loss value, and this embodiment of this application is not limited thereto.

[00168] In some implementations, at least one first key point of the first object may be extracted from the first image. At least one second key point of the first three-dimensional virtual object may be extracted from the second image. The second texture loss value is determined based on the feature value corresponding to the at least one first key point in the first feature map and the feature value corresponding to the at least one second key point in the second feature map. For example, the sum or mean value between the feature value corresponding to each first key point in the first feature map and the feature value corresponding to the second key point corresponding to the first key point in the second feature map may be determined as the second texture loss value.

[00169] In some implementations, first feature pixel points of set number may be randomly selected from the first feature map. Second feature pixel points of set number may be randomly selected from the second feature map. The second texture loss value is determined based on the feature value corresponding to the at least one first feature pixel

point in the first feature map and the feature value corresponding to the at least one second feature pixel point in the second feature map. For example, the sum or mean value of differences between the feature value corresponding to each first feature pixel point in the first feature map and the feature value corresponding to the second feature pixel point corresponding to the first feature pixel point in the second feature map may be determined as the second texture loss value.

[00170] In the above embodiment, the texture loss value includes a first texture loss value and a second texture loss value. Feature extraction is performed on the first image and the second image to obtain the first feature map and the second feature map, and the second texture loss value is determined based on the first feature map and the second feature map. In this way, since the second texture loss value may characterize the texture loss of the first image and the second image at a feature level and the first texture loss value may characterize the texture loss of the first image and the second image at a pixel level, by comprehensively supervising the first texture loss value and the second texture loss value, the second three-dimensional virtual object generated based on the second potential feature can well recover the texture of the first object at both the feature level and the pixel level. Therefore, the fidelity of the generated second three-dimensional virtual object may be further improved.

[00171] In some embodiments, S352 may include S361 to S363.

[00172] At S361, for each first pixel point in the first pixel point set, a third distance between the first pixel point and each second pixel point in the second pixel point set is determined based on the first feature map and the second feature map, and the minimum third distance is determined as a third pixel loss value of the first pixel point.

[00173] Herein, the third distance between the first pixel point and the second pixel point may include, but is not limited to, at least a distance between the first pixel point and the second pixel point in space, a distance in appearance, or the like. The distance between the first pixel point and the second pixel point in space may include a distance between a coordinate value of the first pixel point and a coordinate value of the second pixel point. The distance between the first pixel point and the second pixel point in appearance may include the distance between the feature value corresponding to the first pixel point in the first feature map and the feature value corresponding to the second pixel point in the second feature map. During implementation, the distance between the coordinate value of the first pixel point and the coordinate value of the second pixel point may include, but is not limited to, a cosine distance, a Euclidean distance, and the like. The distance between the feature value corresponding to the first pixel point in the first feature map and the feature value corresponding to the second pixel point in the second feature map may also include, but is not limited to, a cosine distance, a Euclidean distance, and the like.

[00174] At S362, for each second pixel point in the second pixel point set, a fourth distance between the second pixel point and each first pixel point in the first pixel point set is determined based on the first feature map and the second feature map, and the minimum fourth distance is determined as a fourth pixel loss value of the second pixel point.

[00175] Herein, the fourth distance between the second pixel point and the first pixel point may include, but is not limited to, at least a distance between the second pixel point

and the first pixel point in space, a distance in appearance, or the like. The distance between the second pixel point and the first pixel point in space may include a distance between a coordinate value of the second pixel point and a coordinate value of the first pixel point. The distance between the second pixel point and the first pixel point in appearance may include the distance between the feature value corresponding to the second pixel point in the second feature map and the feature value corresponding to the first pixel point in the first feature map.

[00176] At S363, the second texture loss value is determined based on each third pixel loss value and each fourth pixel loss value.

[00177] Herein, a proper manner may be used to determine the second texture loss value based on each third pixel loss value and each fourth pixel loss value according to the actual situations, and this embodiment of this application is not limited thereto.

[00178] In some implementations, the sum of each third pixel loss value and each fourth pixel loss value may be determined as the second texture loss value.

[00179] In some implementations, the mean value of each third pixel loss value and each fourth pixel loss value may be determined as the second texture loss value.

[00180] In some implementations, the sum of the mean value of each third pixel loss value and the mean value of each fourth pixel loss value may be determined as the second texture loss value.

[00181] In the above embodiment, for each first pixel point in the first pixel point set, the third distance between the first pixel point and each second pixel point in the second pixel point set is determined based on the first feature map and the second feature map, and the minimum third distance is determined as the third pixel loss value of the first pixel point. For each second pixel point in the second pixel point set, the fourth distance between the second pixel point and each first pixel point in the first pixel point set is determined based on the first feature map and the second feature map, and the minimum fourth distance is determined as the fourth pixel loss value of the second pixel point. The second texture loss value is determined based on each third pixel loss value and each fourth pixel loss value. In this way, the loss between the first pixel point set and the second pixel point set at the feature level may be considered from an overall angle. That is to say, the texture loss may be calculated by searching the pixel points with matched features in local areas respectively corresponding to the first pixel point set and the second pixel point set, so that texture dislocation due to the non-alignment of the pixel points in the first image and the second image is reduced. Therefore, the more accurate second potential feature and the more realistic second three-dimensional virtual object may be determined.

[00182] In some embodiments, the above S361 in which the third distance between the first pixel point and each second pixel point in the second pixel point set is determined based on the first feature map and the second feature map may include S371.

[00183] At S371, for each second pixel point in the second pixel point set, the third distance between the first pixel point and the second pixel point is determined based on the distance between the feature value corresponding to the first pixel point in the first feature map and the feature value corresponding to the second pixel point in the second feature map and the distance between the coordinate value of the first pixel point and the

coordinate value of the second pixel point.

- [00184] Herein, the pixel points in the first feature map are in one-to-one correspondence with the pixel points in the first image. Each pixel point in the first feature map has the corresponding feature value. Therefore, the feature value of the pixel point corresponding to each first pixel point in the first feature map is the feature value corresponding to the first pixel point in the first feature map.
- [00185] The pixel points in the second feature map are in one-to-one correspondence with the pixel points in the second image. Each pixel point in the second feature map has the corresponding feature value. Therefore, the feature value of the pixel point corresponding to each second pixel point in the second feature map is the feature value corresponding to the second pixel point in the second feature map.
- [00186] The distance between the feature value of the first pixel point in the first feature map and the feature value of the second pixel point in the second feature map may include, but is not limited to, at least one of a cosine distance, a Euclidean distance, or the like.
- [00187] During implementation, the third distance between the first pixel point and the second pixel point may be determined based on the distance between the feature value of the first pixel point in the first feature map and the feature value of the second pixel point in the second feature map and the distance between the coordinate value of the first pixel point and the coordinate value of the second pixel point by using any proper manner, and this embodiment of this application is not limited thereto.
- [00188] In some implementations, the third distance between the first pixel point and the second pixel point may be obtained by summing the distance between the feature value of the first pixel point in the first feature map and the feature value of the second pixel point in the second feature map and the distance between the coordinate value of the first pixel point and the coordinate value of the second pixel point.
- [00189] In some implementations, the third distance between the first pixel point and the second pixel point may be obtained by performing quadrature on the distance between the feature value of the first pixel point in the first feature map and the feature value of the second pixel point in the second feature map and the distance between the coordinate value of the first pixel point and the coordinate value of the second pixel point. For example, the third distance between the first pixel point and the second pixel point may be determined in a manner shown in the following formula 5:

$$D_2 = \max((D_2^s + \dot{o}_{s2})^{\alpha 2}, 1) \times (D_2^a + \dot{o}_{a2}) \quad (5)$$

where, the third distance D_2 between the first pixel point and the second pixel point may be obtained by multiplying a third factor $\max((D_2^s + \dot{o}_{s2})^{\alpha 2}, 1)$ and a fourth factor $(D_2^a + \dot{o}_{a2})$; D_2^s is the distance between the coordinate value of the first pixel point and the coordinate value of the second pixel point; D_2^a is the distance between the feature value of the first pixel point in the first feature map and the feature value of the second pixel point in the second feature map; $\alpha 2$ is a set scale factor; $\dot{o}_{s2}$ and $\dot{o}_{a2}$ are preset residual terms greater than 0; $\dot{o}_{s2}$ and $\dot{o}_{a2}$ may ensure that the third distance D_2 between

the first pixel point and the second pixel point is not 0; the third factor $\max((D_2^s + \dot{o}_{s2})^{\alpha_2}, 1)$ may be used as the weight of the fourth factor $(D_2^a + \dot{o}_{a2})$, and may set $\dot{o}_{s2}$ to be less than 1, to cause $(D_2^s + \dot{o}_{s2})^{\alpha_2}$ to be less than 1 in a case that the degree of the non-alignment of the first pixel point and the second pixel point in space is relatively small, so that the third factor $\max((D_2^s + \dot{o}_{s2})^{\alpha_2}, 1)$ may be maintained at 1. In this case, the offset of the first pixel point and the second pixel point in space is not required to be considered, so that local dislocation between the first three-dimensional virtual object in the generated second image and the first object in the first image is tolerable. Therefore, the second three-dimensional virtual object that is more semantic may be obtained.

[00190] The above S362 in which the fourth distance between the second pixel point and each first pixel point in the first pixel point set is determined based on the first feature map and the second feature map may include S372.

[00191] At S372, for each first pixel point in the first pixel point set, the fourth distance between the second pixel point and the first pixel point is determined based on the distance between the feature value corresponding to the second pixel point in the second feature map and the feature value corresponding to the first pixel point in the first feature map and the distance between the coordinate value of the second pixel point and the coordinate value of the first pixel point.

[00192] Herein, a process of determining the fourth distance between the second pixel point and the first pixel point in S372 corresponds to a process of determining the third distance between the first pixel point and the second pixel point in S371, which may refer to a specific implementation in S371 during implementation.

[00193] In the above embodiment, since the distance between the feature value of the first pixel point in the first feature map and the feature value of the second pixel point in the second feature map may reflect the loss of the first pixel point and the second pixel point at the feature level, and the distance between the coordinate value of the first pixel point and the coordinate value of the second pixel point may reflect the loss of the first pixel point and the second pixel point in space, the third distance between the first pixel point and the second pixel point is determined based on the distance between the feature value of the first pixel point in the first feature map and the feature value of the second pixel point in the second feature map and the distance between the coordinate value of the first pixel point and the coordinate value of the second pixel point; and the fourth distance between the second pixel point and the first pixel point is determined based on the distance between the feature value of the second pixel point in the second feature map and the feature value of the first pixel point in the first feature map and the distance between the coordinate value of the second pixel point and the coordinate value of the first pixel point. In this way, the loss of the first pixel point and the second pixel point at the feature level and in space, so that the robustness of the third distance between the first pixel point and the second pixel point and the fourth distance between the second pixel point and the first pixel point may be improved. Therefore, the accuracy of the second texture loss value may be enhanced, so as to obtain the more accurate second potential feature and the more realistic second three-dimensional virtual object.

[00194] In some embodiments, the first loss value further includes a geometric loss

value. S305 may further include S381 to S383.

- [00195] At S381, according to the first camera pose, each vertex in the first grid is projected to an image plane where the first image is located, to obtain a third pixel point set.
- [00196] Herein, the vertex in the first grid may include a vertex that is visible in the first image and a vertex that is invisible in the first image.
- [00197] During implementation, according to the first camera pose, each vertex in the first grid may be projected to the image plane where the first image is located, to obtain a third pixel point corresponding to each vertex, so as to obtain the third pixel point set.
- [00198] At S382, a fourth pixel point set is determined from the first image based on a contour mask of the first object in the first image.
- [00199] Herein, the contour mask of the first object in the first image may be preset, or may be obtained by segmenting the first object in the first image by using a pre-trained target segmentation network, and this embodiment of this application is not limited thereto.
- [00200] Based on the contour mask of the first object in the first image, at least one fourth pixel point in the contour of the first object may be extracted from the first image, so that the fourth pixel point set may be determined.
- [00201] In some implementations, the coordinate value of each fourth pixel point in the fourth pixel point set may be normalized within a set value range, so that a coordinate of each fourth pixel point is normalized within the value range. Herein, the value range may be preset, and may include, but is not limited to, [-1, 1], [0, 1], and the like.
- [00202] At S383, the geometric loss value is determined based on the third pixel point set and the fourth pixel point set.
- [00203] Herein, the geometric loss value may characterize a geometric loss between each pixel point in the third pixel point set and each pixel point in the fourth pixel point set, that is, the geometric loss between a surface contour point of the first object in the first image and each vertex of the first three-dimensional virtual object in the second image. During implementation, those skilled in the art may use a proper constraint manner to constrain the third pixel point set and the fourth pixel point set according to the actual situations, so as to determine the loss between each pixel point in the third pixel point set and each pixel point in the fourth pixel point set, that is, the geometric loss value. This embodiment of this application is not limited thereto.
- [00204] In some implementations, the geometric loss value may be determined based on the distance between the coordinate value of each pixel point in the third pixel point set and the coordinate value of each pixel point in the fourth pixel point set. During implementation, the distance between a coordinate value of each pixel point in the third pixel point set and a coordinate value of each pixel point in the fourth pixel point set may include, but is not limited to, a cosine distance, a Euclidean distance, a chamfer distance, and the like.
- [00205] In the related art, since a process of obtaining a mask of a grid by rendering the grid usually relates to a rasterization process of discretizing the grid into a pixel grid, the process inevitably leads to partial loss of position information of each contour point on a

surface of the grid, and small perturbation points of the potential features in the generative adversarial network usually correspond to slight changes in three-dimensional structures of the corresponding three-dimensional virtual objects, which causes the mask obtained by rendering the updated three-dimensional virtual objects to be unchanged, resulting in insensitive gradient of back propagation during the updating of the first potential feature, thereby affecting the learning of geometric features of the first object during updating.

[00206] In the above embodiment, according to the first camera pose, each vertex in the first grid is projected to the image plane where the first image is located, to obtain the third pixel point set. The fourth pixel point set is determined from the first image based on the contour mask of the first object in the first image. The geometric loss value is determined based on the third pixel point set and the fourth pixel point set. In this way, each third pixel point in the third pixel point set is obtained by projecting each vertex in the first grid. Since the projection process is not related to coordinate discrete processing in the rasterization process, the coordinates of each third pixel in the image plane are continuous, so that the loss of the position information of the vertex of the grid may be reduced. Therefore, the geometric loss value obtained based on the third pixel point set and the fourth pixel point set may better characterize the geometric loss between the surface contour point of the first object in the first image and each vertex of the first three-dimensional virtual object in the second image, and by means of the supervision of the geometric loss value, the more accurate second potential feature and the more realistic second three-dimensional virtual object may be obtained.

[00207] In some embodiments, S383 may include S391 to S393.

[00208] At S391, for each third pixel point in the third pixel point set, a fifth distance between a coordinate value of the third pixel point and a coordinate value of each fourth pixel point in the fourth pixel point set is determined, and the minimum fifth distance is determined as a first geometric loss sub-value of the third pixel point.

[00209] Herein, the fifth distance between the coordinate value of the third pixel point and the coordinate value of the fourth pixel point may include, but is not limited to, at least one of a cosine distance, a Euclidean distance, or the like.

[00210] At S392, for each fourth pixel point in the fourth pixel point set, a sixth distance between a coordinate value of the fourth pixel point and a coordinate value of each third pixel point in the third pixel point set is determined, and the minimum sixth distance is determined as a second geometric loss sub-value of the fourth pixel point.

[00211] Herein, the sixth distance between the coordinate value of the fourth pixel point and the coordinate value of the third pixel point may include, but is not limited to, at least one of a cosine distance, a Euclidean distance, or the like.

[00212] At S393, the geometric loss value is determined based on each first geometric loss sub-value and each second geometric loss sub-value.

[00213] Herein, a proper manner may be used to determine the geometric loss value based on each first geometric loss sub-value and each second geometric loss sub-value according to the actual situations, and this embodiment of this application is not limited thereto.

[00214] In some implementations, the sum of each first geometric loss sub-value and

each second geometric loss sub-value may be determined as the geometric loss value.

[00215] In some implementations, the mean value of each first geometric loss sub-value and each second geometric loss sub-value may be determined as the geometric loss value.

[00216] In some implementations, the sum of the mean value of each first geometric loss sub-value and the mean value of each second geometric loss sub-value may be determined as the geometric loss value.

[00217] In the above embodiment, for each third pixel point in the third pixel point set, the fifth distance between the coordinate value of the third pixel point and the coordinate value of each fourth pixel point in the fourth pixel point set is determined, and the minimum fifth distance is determined as the first geometric loss sub-value of the third pixel point. For each fourth pixel point in the fourth pixel point set, the sixth distance between the coordinate value of the fourth pixel point and the coordinate value of each third pixel point in the third pixel point set is determined, and the minimum sixth distance is determined as the second geometric loss sub-value of the fourth pixel point. The geometric loss value is determined based on each first geometric loss sub-value and the mean value of each second geometric loss sub-value. In this way, the loss between the third pixel point set and the fourth pixel point set may be considered from an overall angle. That is to say, the geometric loss may be calculated by searching matched pixel points in local areas respectively corresponding to the third pixel point set and the fourth pixel point set, so that geometric dislocation due to the non-alignment of the pixel points in the first image and the second image is reduced. Therefore, the more accurate second potential feature and the more realistic second three-dimensional virtual object may be determined.

[00218] The application of the method for generating a virtual object provided in this embodiment of this application in an actual scenario is described below by using a three-dimensional reconstruction scenario of a monocular image as an example.

[00219] According to the method for generating a virtual object provided in this embodiment of this application, three-dimensional reconstruction is performed on the three-dimensional structure and texture of the first object by using prior knowledge of the object category to which the first object included in the monocular image (corresponding to the first image in the foregoing embodiments) learned by the trained generative adversarial network belongs. Therefore, three-dimensional reconstruction quality and generalization capability may be enhanced. The method may be performed by a processor of a computer device.

[00220] FIG. 4 is a schematic implementation flowchart of a method for generating a virtual object according to an embodiment of this application. The method for generating a virtual object provided in this embodiment of this application is described below with reference to FIG. 4. The method includes the following steps.

[00221] At S401, a monocular image I_{in} including the first object is acquired, a first camera pose π of the monocular image I_{in} is extracted by using a pre-trained camera pose estimation network, and a contour mask M_{in} of the first object in the monocular image I_{in} is extracted by using a pre-trained target segmentation network.

- [00222] At S402, based on the first potential feature z of the generative adversarial network, the first geometric map S_1 and the first texture map I_1 of the first three-dimensional virtual object is generated by using the trained generative network G . The generative adversarial network is trained by using the sample image set. Each sample image in the sample image set includes the second object that belongs to the same object category as the first object.
- [00223] At S403, a renderer R is used to perform, according to the first camera pose π , image rendering based on the first geometric map S_1 and the first texture map T_1 , to obtain a rendered image I_r (corresponding to the second image in the foregoing embodiments).
- [00224] At S404, the texture loss value L_{CT} is determined based on the monocular image I_{in} and the rendered image I_r . The texture loss value L_{CT} is determined based on space loss D_1^s and appearance loss D_1^a .
- [00225] At S405, a first grid M_1 is determined based on the first geometric map S_1 and a set first grid template.
- [00226] At S406, according to the first camera pose π , each vertex in the first grid M_1 is projected to the image plane where the monocular image I_{in} is located, to obtain a vertex projection point set S_v (corresponding to the third pixel point set).
- [00227] At S407, based on the contour mask M_{in} , a contour point set (corresponding to the fourth pixel point set) is determined from the monocular image I_{in} .
- [00228] At S408, based on the vertex projection point set S_v and the contour point set, the geometric loss value L_{CM} is determined.
- [00229] At S409, based on the texture loss value L_{CT} and the geometric loss value L_{CM} , a total loss value (corresponding to the first loss value in the foregoing embodiments) is determined.
- [00230] Herein, the sum of the texture loss value L_{CT} and the geometric loss value L_{CM} is determined as the total loss value.
- [00231] In some implementations, a cosine distance between normal of adjacent surfaces of the first grid and a L2 norm of the first potential feature z may alternatively be determined. Based on the texture loss value L_{CT} , the geometric loss value L_{CM} , the cosine distance between the normal of the adjacent surfaces of the first grid, and the L2 norm of the first potential feature z , the total loss value may be determined. For example, the sum of the texture loss value L_{CT} , the geometric loss value L_{CM} , the cosine distance between the normal of the adjacent surfaces of the first grid, and the L2 norm of the first

potential feature z may be determined as the total loss value.

- [00232] At S410, based on the total loss value, the first potential feature z and the first camera pose π are updated to obtain the second potential feature and the second three-dimensional virtual object corresponding to the second potential feature. The second three-dimensional virtual object is generated based on the second potential feature by using a generative network G .
- [00233] Herein, the second potential feature is the first potential feature in the manifold of the generative adversarial network that minimizes the total loss value.
- [00234] According to the method for generating a virtual object provided in this embodiment of this application, the accuracy and authenticity of three-dimensional reconstruction may be greatly enhanced. The three-dimensional reconstruction of an object that is partially occluded can be well resolved. During three-dimensional reconstruction, an excellent generalization effect can be shown for geometric shapes that are not common in a class of objects.
- [00235] The method for generating a virtual object provided in this embodiment of this application is applicable to industries such as media editing, films, and live broadcast, or scenarios such as virtual reality, metaverse, and teleconferencing. For example, during teleconferencing of a user A and a user B, in a case that the user A needs to show a complete three-dimensional structure of some object to the user B but a camera for collecting video information and the object are inconvenient to move, a second terminal of the user B may acquire the monocular image sent by a first terminal of the user A. Then, by using the method for generating a virtual object provided in this embodiment of this application, the three-dimensional structure and texture of the object is reconstructed from the monocular image, so that immersive teleconferencing experience can be achieved.
- [00236] Based on the foregoing embodiments, an embodiment of this application provides an apparatus for generating a virtual object. The apparatus includes the included units and modules included in the units, and may be implemented by using a processor in a computer device. Definitely, the apparatus may alternatively be implemented by using a specific logic circuit. During implementation, the processor may be a Central Processing Unit (CPU), a Microprocessor Unit (MPU), a Digital Signal Processor (DSP), or a Field Programmable Gate Array (FPGA).
- [00237] FIG. 5 is a schematic diagram of a composition structure of an apparatus for generating a virtual object according to an embodiment of this application. As shown in FIG. 5, the apparatus 500 for generating a virtual object includes a first acquisition module 510, a first generation module 520, a first rendering module 530, a first determination module 540, and a first updating module 550.
- [00238] The first acquisition module 510 is configured to acquire a first image including a first object.
- [00239] The first generation module 520 is configured to generate, based on a first potential feature of a generative adversarial network, a first three-dimensional virtual object by using the generative adversarial network. The generative adversarial network is trained by using a sample image set. Each sample image in the sample image set includes a second object that belongs to a same object category as the first object.

- [00240] The first rendering module 530 is configured to render the first three-dimensional virtual object to a second image according to a first camera pose of the first image.
- [00241] The first determination module 540 is configured to determine a first loss value based on the first image and the second image.
- [00242] The first updating module 550 is configured to update, based on the first loss value, the first potential feature and the first camera pose to obtain a second potential feature and a second three-dimensional virtual object corresponding to the second potential feature. The second three-dimensional virtual object is generated based on the second potential feature by using the generative adversarial network.
- [00243] In some embodiments, the first updating module is further configured to: in a case that the first loss value does not satisfy a set loss condition, respectively updating the first potential feature and the first camera pose, to obtain the updated first potential feature and the updated first camera pose; and determine the second potential feature and the second three-dimensional virtual object corresponding to the second potential feature based on the updated first potential feature and the updated first camera pose.
- [00244] In some embodiments, the first updating module is further configured to: generate, based on the updated first potential feature, the updated first three-dimensional virtual object by using the generative adversarial network; determine an updated second image based on the updated first camera pose and the updated first three-dimensional virtual object; determine an updated first loss value based on the first image and the updated second image; and in a case that the updated first loss value satisfies the set loss condition, determine the updated first potential feature as the second potential feature, and determine the updated first three-dimensional virtual object as the second three-dimensional virtual object.
- [00245] In some embodiments, the first three-dimensional virtual object has a first geometric map and a first texture map. The first loss value includes a texture loss value. The first rendering module is further configured to: determine a first grid based on the first geometric map and a set first grid template; and perform image rendering based on the first texture map and the first grid according to the first camera pose, to obtain a rendered second image. The first determination module is further configured to determine a texture loss value based on the first image and the second image.
- [00246] In some embodiments, the texture loss value includes a first texture loss value. The first determination module is further configured to: respectively perform foreground extraction on the first image and the second image to obtain a first pixel point set and a second pixel point set; and determine the first texture loss value based on the first pixel point set and the second pixel point set.
- [00247] In some embodiments, the first determination module is further configured to: for each first pixel point in the first pixel point set, determine a first distance between the first pixel point and each second pixel point in the second pixel point set, and determine the minimum first distance as a first pixel loss value of the first pixel point; for each second pixel point in the second pixel point set, determine a second distance between the second pixel point and each first pixel point in the first pixel point set, and determine the minimum second distance as a second pixel loss value of the second pixel point; and determine the first texture loss value based on each first pixel loss value and each second

pixel loss value.

[00248] In some embodiments, the first determination module is further configured to: for each second pixel point in the second pixel point set, determine the first distance between the first pixel point and the second pixel point based on a distance between a pixel value of the first pixel point and a pixel value of the second pixel point and a distance between a coordinate value of the first pixel point and a coordinate value of the second pixel point; and for each first pixel point in the first pixel point set, determine the second distance between the second pixel point and the first pixel point based on a distance between a pixel value of the second pixel point and a pixel value of the first pixel point and a distance between a coordinate value of the second pixel point and a coordinate value of the first pixel point.

[00249] In some embodiments, the texture loss value further includes a second texture loss value. The first determination module is further configured to: respectively perform feature extraction on the first image and the second image to obtain a first feature map and a second feature map; and determine the second texture loss value based on the first feature map and the second feature map.

[00250] In some embodiments, the first determination module is further configured to: for each first pixel point in the first pixel point set, determine a third distance between the first pixel point and each second pixel point in the second pixel point set based on the first feature map and the second feature map, and determining the minimum third distance as a third pixel loss value of the first pixel point; for each second pixel point in the second pixel point set, determine a fourth distance between the second pixel point and each first pixel point in the first pixel point set based on the first feature map and the second feature map, and determine the minimum fourth distance as a fourth pixel loss value of the second pixel point; and determine the second texture loss value based on each third pixel loss value and each fourth pixel loss value.

[00251] In some embodiments, the first determination module is further configured to: for each second pixel point in the second pixel point set, determine the third distance between the first pixel point and the second pixel point based on a distance between a feature value corresponding to the first pixel point in the first feature map and a feature value corresponding to the second pixel point in the second feature map and a distance between a coordinate value of the first pixel point and a coordinate value of the second pixel point; and for each first pixel point in the first pixel point set, determine the fourth distance between the second pixel point and the first pixel point based on a distance between a feature value corresponding to the second pixel point in the second feature map and a feature value corresponding to the first pixel point in the first feature map and a distance between a coordinate value of the second pixel point and a coordinate value of the first pixel point.

[00252] In some embodiments, the first loss value further includes a geometric loss value. The first determination module is further configured to: projecting, according to the first camera pose, each vertex in the first grid to an image plane where the first image is located, to obtain a third pixel point set; determine a fourth pixel point set from the first image based on a contour mask of the first object in the first image; and determine the geometric loss value based on the third pixel point set and the fourth pixel point set.

[00253] In some embodiments, the first determination module is further configured to:

for each third pixel point in the third pixel point set, determine a fifth distance between a coordinate value of the third pixel point and a coordinate value of each fourth pixel point in the fourth pixel point set, and determine the minimum fifth distance as a first geometric loss sub-value of the third pixel point; for each fourth pixel point in the fourth pixel point set, determine a sixth distance between a coordinate value of the fourth pixel point and a coordinate value of each third pixel point in the third pixel point set, and determine the minimum sixth distance as a second geometric loss sub-value of the fourth pixel point; and determine the geometric loss value based on each first geometric loss sub-value and each second geometric loss sub-value.

[00254] In some embodiments, the second three-dimensional virtual object has a second geometric map and a second texture map. The apparatus further includes: a second determination module, configured to determine a second grid based on the second geometric map and a set second grid template; and a second rendering module, configured to perform image rendering based on the second texture map and the second grid according to a set second camera pose, to obtain a rendered third image.

[00255] In some embodiments, the apparatus further includes: a second acquisition module, configured to acquire a fourth image including a third object, where the third object and the first object belong to a same object category; a second generation module, configured to generate, based on a third potential feature, a third three-dimensional virtual object by using the generative adversarial network; a third rendering module, configured to render the third three-dimensional virtual object as a fifth image according to a third camera pose of the fourth image; a third determination module, configured to determine a second loss value based on the fourth image and the fifth image; a second updating module, configured to update, based on the second loss value, the third potential feature and the third camera pose to obtain a fourth potential feature and a fourth three-dimensional virtual object corresponding to the fourth potential feature, where the fourth three-dimensional virtual object is generated based on the fourth potential feature by using the generative adversarial network; and a migration module, configured to perform texture migration between the second three-dimensional virtual object and the fourth three-dimensional virtual object, to obtain a fifth three-dimensional virtual object and a sixth three-dimensional virtual object.

[00256] The descriptions of the above apparatus embodiments are similar to the descriptions of the above method embodiments, and have similar beneficial effects to the method embodiments. In some embodiments, functions or modules of the apparatus provided in the embodiments of this application may be configured to execute the method described in the above method embodiment. For technical details not disclosed in the apparatus embodiments of this application, refer to the descriptions of the method embodiments of this application for understanding.

[00257] It is to be noted that, in the embodiments of this application, if the above method for generating a virtual object is implemented in the form of a software functional module and sold or used as an independent product, it can be stored in the computer-readable storage medium. Based on such an understanding, the technical solutions of the embodiments of this application substantially or parts making contributions to the related art may be embodied in form of a software product, and the software product is stored in a storage medium, including a plurality of instructions for causing a computer device (which may be a personal computer, a server, a network

device or the like) to execute all or part of the method in each embodiment of the present disclosure. The foregoing storage medium includes: a USB flash disk, a mobile Hard Disk Drive (HDD), a Read-Only Memory (ROM), and various media that can store program codes, such as a magnetic disk, or an optical disk. In this way, the embodiments of this application are not limited to any specific hardware, software, or firmware, or any combination of hardware, software, and firmware.

[00258] An embodiment of this application provides a computer device. The computer device includes a memory and a processor. The memory stores a computer program executable on the processor. The processor, when executing the computer program, implements the above method.

[00259] An embodiment of the present disclosure provides a computer-readable storage medium. The computer-readable storage medium stores a computer program. The above method is implemented when the computer program is performed by a processor. The computer-readable storage medium may be transitory or non-transitory.

[00260] An embodiment of this application provides a computer program. The computer program includes a computer-readable code. When the computer-readable code is operated in a computer device, a processor in the computer device executes and implements part or all of steps of the above method.

[00261] An embodiment of this application provides a computer program product. The computer program product includes a non-transitory computer-readable storage medium storing a computer program. When the computer program is read and executed by a computer, part or all of steps in the above method are implemented. The computer program product can be specifically implemented in a manner of hardware, software or a combination thereof. In some embodiments, the computer program product is embodied as a computer storage medium. In some other embodiments, the computer program product is embodied as a software product, such as a Software Development Kit (SDK) and the like.

[00262] Herein, it is to be noted that, the above description of the various embodiments tends to emphasize the differences between the various embodiments, and the identicalness or similarities can be referred to each other. The descriptions of the above device, storage medium, computer program, and computer program product embodiments are similar to the descriptions of the above method embodiments, and have similar beneficial effects to the method embodiments. For technical details not disclosed in the embodiments of the device, storage medium, computer program and computer program product of this application, refer to the description of the method embodiments of this application for understanding.

[00263] It is to be noted that, FIG. 6 is a schematic diagram of a hardware entity of a computer device according to an embodiment of this application. As shown in FIG. 6, a hardware entity of the computer device 600 includes a processor 601, a communication interface 602, and a memory 603.

[00264] The processor 601 usually controls an overall operation of the computer device 600.

[00265] The communication interface 602 may cause the computer device to achieve communication with other terminals or servers through a network.

- [00266] The memory 603 is configured to store an instruction and an application executable by the processor 601, or may cache data to be processed or processed by the processor 601 and each module in the computer device 600 (such as, image data, audio data, voice communication data, and video communication data), which may be implemented by using FLASH or a Random Access Memory (RAM). Data transmission among the processor 601, the communication interface 602, and the memory 603 may be achieved by using a bus 604.
- [00267] It is to be understood that "one embodiment" and "an embodiment" mentioned in the whole specification mean that specific features, structures or characteristics related to the embodiment is included in at least one embodiment of this application. Therefore, "in one embodiment" or "in an embodiment" appearing at any place of the whole specification does not always refer to the same embodiment. In addition, these specific features, structures or characteristics may be combined in one or more embodiments in any proper manner. It is to be understood that, in various embodiments of this application, the sequence number of each step/process does not mean the sequence of execution. The execution sequence of each step/process should be determined by its functions and internal logic, which should not constitute any limitation on the implementation process of the embodiments of this application. The serial numbers of the foregoing embodiments of this application are merely for description, and do not represent the superiority or inferiority of the embodiments.
- [00268] It is to be noted that terms "include" and "comprise" or any other variant thereof is intended to cover nonexclusive inclusions herein, so that a process, method, object or apparatus including a series of components not only includes those components but also includes other components which are not clearly listed or further includes components intrinsic to the process, the method, the object or the apparatus. Under the condition of no more limitations, a component defined by the statement "including a/an..." does not exclude existence of the same other components in a process, method, object or apparatus including the component.
- [00269] In several embodiments provided by this application, it is to be understood that the disclosed device and method may be implemented in other ways. The device embodiment described above is only schematic, and for example, division of the units is only logic function division, and other division manners may be adopted during practical implementation. For example, a plurality of units or components may be combined or integrated into another system, or some characteristics may be neglected or not executed. In addition, coupling or direct coupling or communication connection between each displayed or discussed component may be indirect coupling or communication connection, implemented through some interfaces, of the device or the units, and may be electrical and mechanical or adopt other forms.
- [00270] The units described as separate parts may or may not be physically separated, and parts displayed as units may or may not be physical units, and namely may be located in the same place, or may also be distributed to multiple network units. Part of all of the units may be selected according to a practical requirement to achieve the purposes of the solutions of the embodiments.
- [00271] In addition, each functional unit in each embodiment of this application may be integrated into a processing unit, each unit may also serve as an independent unit and two or more than two units may also be integrated into a unit. The integrated unit may be

implemented in a hardware form and may also be implemented in form of hardware and software functional unit.

[00272] Those of ordinary skill in the art should know that all or part of the steps of the method embodiment may be implemented by related hardware instructed through a program, the program may be stored in a computer-readable storage medium, and the program is executed to execute the steps of the method embodiment. The storage medium which may be a volatile storage medium includes: a mobile storage device, an ROM, and various media that can store program codes, such as a magnetic disk, or an optical disk.

[00273] If the integrated unit is implemented in the form of the software functional unit and sold or used as an independent product, it can be stored in the computer readable storage medium. Based on such an understanding, the technical solutions of this application substantially or parts making contributions to the related art may be embodied in form of a software product, and the computer software product is stored in a storage medium, including a plurality of instructions for causing a computer device (which may be a personal computer, a server, a network device or the like) to execute all or part of the method in each embodiment of this application. The foregoing storage medium includes a portable storage device, an ROM, and various media that can store program codes, such as a magnetic disk, or an optical disk.

[00274] The above is only the implementations of this application and not intended to limit the scope of protection of this application. Any variations or replacements apparent to those skilled in the art within the technical scope disclosed by this application shall fall within the scope of protection of this application.

CLAIMS

1. A method for generating a virtual object, comprising:

acquiring a first image comprising a first object;

generating, based on a first potential feature of a generative adversarial network, a first three-dimensional virtual object by using the generative adversarial network, wherein the generative adversarial network is trained by using a sample image set, and each sample image in the sample image set comprises a second object that belongs to a same object category as the first object;

rendering the first three-dimensional virtual object as a second image according to a first camera pose of the first image;

determining a first loss value based on the first image and the second image; and

updating, based on the first loss value, the first potential feature and the first camera pose to obtain a second potential feature and a second three-dimensional virtual object corresponding to the second potential feature, wherein the second three-dimensional virtual object is generated based on the second potential feature by using the generative adversarial network.

2. The method of claim 1, wherein the updating, based on the first loss value, the first potential feature and the first camera pose to obtain a second potential feature and a second three-dimensional virtual object corresponding to the second potential feature comprises:

in a case that the first loss value does not satisfy a set loss condition, respectively updating the first potential feature and the first camera pose, to obtain the updated first potential feature and the updated first camera pose; and

determining the second potential feature and the second three-dimensional virtual object corresponding to the second potential feature based on the updated first potential feature and the updated first camera pose.

3. The method of claim 2, wherein the determining the second potential feature and the second three-dimensional virtual object corresponding to the second potential feature based on the updated first potential feature and the updated first camera pose comprises:

generating, based on the updated first potential feature, an updated first three-dimensional virtual object by using the generative adversarial network;

determining an updated second image based on the updated first camera pose and the updated first three-dimensional virtual object;

determining an updated first loss value based on the first image and the updated second image; and

in a case that the updated first loss value satisfies the set loss condition, determining the updated first potential feature as the second potential feature, and determining the updated first three-dimensional virtual object as the second three-dimensional virtual object.

4. The method of any of claims 1 to 3, wherein the first three-dimensional virtual object has a first geometric map and a first texture map; the first loss value comprises a texture loss value;

wherein the rendering the first three-dimensional virtual object as a second image according to a first camera pose of the first image comprises:

determining a first grid based on the first geometric map and a set first grid template;
and

performing image rendering based on the first texture map and the first grid according to the first camera pose, to obtain a rendered second image;

wherein the determining a first loss value based on the first image and the second image comprises:

determining a texture loss value based on the first image and the second image.

5. The method of claim 4, wherein the texture loss value comprises a first texture loss value;

wherein the determining a texture loss value based on the first image and the second image comprises:

respectively performing foreground extraction on the first image and the second image to obtain a first pixel point set and a second pixel point set; and

determining the first texture loss value based on the first pixel point set and the second pixel point set.

6. The method of claim 5, wherein the determining the first texture loss value based on the first pixel point set and the second pixel point set comprises:

for each first pixel point in the first pixel point set, determining a first distance between the first pixel point and each second pixel point in the second pixel point set, and determining a minimum first distance as a first pixel loss value of the first pixel point;

for each second pixel point in the second pixel point set, determining a second distance between the second pixel point and each first pixel point in the first pixel point set, and determining a minimum second distance as a second pixel loss value of the second pixel point; and

determining the first texture loss value based on each first pixel loss value and each second pixel loss value.

7. The method of claim 6, wherein the determining a first distance between the first pixel point and each second pixel point in the second pixel point set comprises:

for each second pixel point in the second pixel point set, determining the first distance between the first pixel point and the second pixel point based on a distance between a pixel value of the first pixel point and a pixel value of the second pixel point and a distance between a coordinate value of the first pixel point and a coordinate value of the second pixel point; and

wherein the determining a second distance between the second pixel point and each first pixel point in the first pixel point set comprises:

for each first pixel point in the first pixel point set, determining the second distance between the second pixel point and the first pixel point based on a distance between a pixel value of the second pixel point and a pixel value of the first pixel point and a distance between a coordinate value of the second pixel point and a coordinate value of the first pixel point.

8. The method of any of claims 5 to 7, wherein the texture loss value further comprises a second texture loss value;

wherein the determining a texture loss value based on the first image and the second image further comprises:

respectively performing feature extraction on the first image and the second image to obtain a first feature map and a second feature map; and

determining the second texture loss value based on the first feature map and the second feature map.

9. The method of claim 8, wherein the determining the second texture loss value based on the first feature map and the second feature map comprises:

for each first pixel point in the first pixel point set, determining a third distance between the first pixel point and each second pixel point in the second pixel point set based on the first feature map and the second feature map, and determining a minimum third distance as a third pixel loss value of the first pixel point;

for each second pixel point in the second pixel point set, determining a fourth distance between the second pixel point and each first pixel point in the first pixel point set based on the first feature map and the second feature map, and determining a minimum fourth distance as a fourth pixel loss value of the second pixel point; and

determining the second texture loss value based on each third pixel loss value and each fourth pixel loss value.

10. The method of claim 9, wherein the determining a third distance between the first pixel point and each second pixel point in the second pixel point set based on the first feature map and the second feature map comprises:

for each second pixel point in the second pixel point set, determining the third distance between the first pixel point and the second pixel point based on a distance between a feature value corresponding to the first pixel point in the first feature map and a feature value corresponding to the second pixel point in the second feature map and a distance between a coordinate value of the first pixel point and a coordinate value of the second pixel point; and

wherein the determining a fourth distance between the second pixel point and each first pixel point in the first pixel point set based on the first feature map and the second feature map comprises:

for each first pixel point in the first pixel point set, determining the fourth distance between the second pixel point and the first pixel point based on a distance between a feature value corresponding to the second pixel point in the second feature map and a feature value corresponding to the first pixel point in the first feature map and a distance between a coordinate value of the second pixel point and a coordinate value of the first pixel point.

11. The method of any of claims 4 to 10, wherein the first loss value further comprises a geometric loss value;

wherein the determining a first loss value based on the first image and the second image further comprises:

projecting, according to the first camera pose, each vertex in the first grid to an image plane where the first image is located, to obtain a third pixel point set;

determining a fourth pixel point set from the first image based on a contour mask of the first object in the first image; and

determining the geometric loss value based on the third pixel point set and the fourth pixel point set.

12. The method of claim 11, wherein the determining the geometric loss value based on the third pixel point set and the fourth pixel point set comprises:

for each third pixel point in the third pixel point set, determining a fifth distance between a coordinate value of the third pixel point and a coordinate value of each fourth pixel point in the fourth pixel point set, and determining a minimum fifth distance as a first geometric loss sub-value of the third pixel point;

for each fourth pixel point in the fourth pixel point set, determining a sixth distance between a coordinate value of the fourth pixel point and a coordinate value of each third pixel point in the third pixel point set, and determining a minimum sixth distance as a second geometric loss sub-value of the fourth pixel point; and

determining the geometric loss value based on each first geometric loss sub-value and each second geometric loss sub-value.

13. The method of any of claims 1 to 12, wherein the second three-dimensional virtual object has a second geometric map and a second texture map;

wherein the method further comprises:

determining a second grid based on the second geometric map and a set second grid template; and

performing image rendering based on the second texture map and the second grid according to a set second camera pose, to obtain a rendered third image.

14. The method of any of claims 1 to 13, further comprising:

acquiring a fourth image comprising a third object, wherein the third object and the first object belong to a same object category;

generating, based on a third potential feature, a third three-dimensional virtual object by using the generative adversarial network;

rendering the third three-dimensional virtual object as a fifth image according to a third camera pose of the fourth image;

determining a second loss value based on the fourth image and the fifth image;

updating, based on the second loss value, the third potential feature and the third camera pose to obtain a fourth potential feature and a fourth three-dimensional virtual object corresponding to the fourth potential feature, wherein the fourth three-dimensional virtual object is generated based on the fourth potential feature by using the generative adversarial network; and

performing texture migration between the second three-dimensional virtual object and the fourth three-dimensional virtual object, to obtain a fifth three-dimensional virtual object and a sixth three-dimensional virtual object.

15. An apparatus for generating a virtual object, comprising:

a first acquisition module, configured to acquire a first image comprising a first object;

a first generation module, configured to generate, based on a first potential feature of a generative adversarial network, a first three-dimensional virtual object by using the generative adversarial network, wherein the generative adversarial network is trained by using a sample image set, and each sample image in the sample image set comprises a second object that belongs to a same object category as the first object;

a first rendering module, configured to render the first three-dimensional virtual object to a second image according to a first camera pose of the first image;

a first determination module, configured to determine a first loss value based on the first image and the second image; and

a first updating module, configured to update, based on the first loss value, the first potential feature and the first camera pose to obtain a second potential feature and a second three-dimensional virtual object corresponding to the second potential feature, wherein the second three-dimensional virtual object is generated based on the second potential feature by using the generative adversarial network.

16. The apparatus of claim 15, wherein the first updating module is configured to:

in a case that the first loss value does not satisfy a set loss condition, respectively update the first potential feature and the first camera pose, to obtain the updated first potential feature and the updated first camera pose; and

determine the second potential feature and the second three-dimensional virtual object corresponding to the second potential feature based on the updated first potential feature and the updated first camera pose.

17. The apparatus of claim 16, wherein the first updating module is configured to:

generate, based on the updated first potential feature, an updated first three-dimensional virtual object by using the generative adversarial network;

determine an updated second image based on the updated first camera pose and the updated first three-dimensional virtual object;

determine an updated first loss value based on the first image and the updated second image; and

in a case that the updated first loss value satisfies the set loss condition, determine the updated first potential feature as the second potential feature, and determine the updated first three-dimensional virtual object as the second three-dimensional virtual object.

18. A computer device, comprising a memory and a processor, wherein the memory stores a computer program executable on the processor; and the processor, when executing the program, implements steps in the method of any of claims 1 to 14.

19. A computer-readable storage medium, having a computer program stored thereon, wherein steps in the method of any of claims 1 to 14 are implemented when the computer program is performed by a processor.

20. A computer program product, comprising a non-transitory computer-readable storage medium storing a computer program, wherein, when the computer program is read and executed by a computer, steps in the method of any of claims 1 to 14 are implemented.