

MODELING METHOD FOR METAVERSE SCENE MATERIAL AND RELATED DEVICE

[0001] The present application claims priority to Chinese Patent Application No.
5 202210454866.4, titled "METHOD FOR MODELING ELEMENT IN METAVERSE
SCENARIO AND RELATED APPARATUSES", filed on April 24, 2022 with the China
National Intellectual Property Administration, which is incorporated herein by reference in its
entirety.

FIELD

[0002] The embodiments of the present disclosure relate to the technical field of image
processing, and in particular to a method for modeling an element in a metaverse scenario and
related apparatuses.

BACKGROUND

[0003] In order to meet the demand for adding elements to a 2D metaverse scenario, a
common method is to manually draw the elements. A designer draws the elements in advance,
and then colors the elements separately.

[0004] The method has at least the following disadvantages.

[0005] 1. The expansion of the metaverse scenario necessitates a lot of elements, while
developing these elements is time-consuming and labor-intensive.

[0006] 2. The developed elements fail to meet the demand for personalized customization
of users.

SUMMARY

[0007] A method for modeling an element in a metaverse scenario is provided according to
an embodiment of the present disclosure, to generate a model customizable and editable by
users from an acquired image of a to-be-modeled object, so as to meet personalized demands
of users on a premise of improving an efficiency of generating the element.

[0008] In a first aspect of the embodiments of the present disclosure, a method for modeling an element in a metaverse scenario is provided. The method includes: acquiring an image of a to-be-modeled object; performing edge detection on an object in the image to extract edges of the object in the image; vectorizing the edges of the object in the image to obtain a vector graphic of the edges of the object in the image; and editing vector lines in the vector graphic and a closed region defined by the vector lines to obtain a model of the object in the image.

[0009] In an embodiment, the performing edge detection on an object in the image to extract edges of the object in the image includes: acquiring an edge detection model, where the edge detection model includes an edge extraction module and an up-sampling module; inputting the image to the edge extraction module to gradually extract edge features of the object in the image through a convolutional network in the edge extraction module; and inputting the edge features of the object in the image to the up-sampling module to up-sample the edge features of the object in the image, and averaging the up-sampled edge features to extract the edges of the object in the image.

[0010] In an embodiment, after the edges of the object in the image are extracted and before the edges of the object in the image are vectorized, the method further includes: de-noising the edges of the object in the image to obtain de-noised edges of the object in the image.

[0011] In an embodiment, the de-noising the edges of the object in the image to obtain de-noised edges of the object in the image includes: acquiring a structural line extraction model, where the structural line extraction model includes a preset number of down-sampling modules and a preset number of up-sampling modules, and the down-sampling modules and the up-sampling modules each are followed by a residual neural network module; inputting the edges of the image to the preset number of down-sampling modules and respective residual neural network modules following the down-sampling modules sequentially to down-sample the edges of the object in the image, and memorizing features of down-sampled edges of the object in the image; and inputting the features of the down-sampled edges of the object in the image to the preset number of up-sampling modules and respective residual neural network modules following the up-sampling modules sequentially to up-sample the features of the edges of the object in the image, and memorizing up-sampled features of the edges of the object in the image to obtain de-noised edges of the object in the image.

[0012] In an embodiment, after the de-noised edges of the object in the image are obtained,

the method further includes: normalizing the de-noised edges of the object in the image, to even the de-noised edges of the object in the image in thickness and color.

[0013] In an embodiment, the normalizing the de-noised edges of the object in the image, to even the de-noised edges of the object in the image in thickness and color includes: acquiring
5 a pre-trained line-width standardization model, where the line-width standardization model includes at least one of a wide network model and a flexible network model; and inputting the de-noised edges of the object in the image to the line-width standardization model, to even the de-noised edges of the object in the image in thickness and color.

[0014] In an embodiment, the wide network model has the same network structure as the
10 flexible network model, and the number of convolution cores in the wide network model is different from the number of convolution cores in the flexible network model. A first layer of the wide network model is a convolution kernel of $N \times N$, a last layer of the wide network model is a sigmoid function, and other layer of the wide network model is a convolution kernel of $M \times M$. Each convolution layer is followed by a normalization function and an
15 activation function. N is greater than M .

[0015] In an embodiment, the vectorizing the edges of the object in the image includes: vectorizing the edges of the object in the image using a preset vectorizing image processing tool, where the preset vectorizing image processing tool includes a Potrace tool and an Imagemagick tool.

[0016] In a second aspect of the embodiments of the present disclosure, an apparatus for
20 modeling an element in a metaverse scenario is provided. The apparatus includes an acquisition unit, an edge detection unit, a vectorizing unit, and an editing unit. The acquisition unit is configured to acquire an image of a to-be-modeled object. The edge detection unit is configured to perform edge detection on an object in the image to extract edges of the object
25 in the image. The vectorizing unit is configured to vector the edges of the object in the image to obtain a vector graphic of the edges of the object in the image. The editing unit is configured to edit vector lines in the vector graphic and a closed region defined by the vector lines to obtain a model of the object in the image.

[0017] In an embodiment, the edge detection unit is configured to acquire an edge detection
30 model, where the edge detection model includes an edge extraction module and an up-sampling module; input the image to the edge extraction module to gradually extract edge

features of the object in the image through a convolutional network in the edge extraction module; and input the edge features of the object in the image to the up-sampling module to up-sample the edge features of the object in the image, and average the up-sampled edge features to extract the edges of the object in the image.

5 **[0018]** In an embodiment, the apparatus further includes a de-noising unit. The de-noising unit is configured to de-noise the edges of the object in the image to obtain de-noised edges of the object in the image.

10 **[0019]** In an embodiment, the de-noising unit is configured to: acquire a structural line extraction model, where the structural line extraction model includes a preset number of down-sampling modules and a preset number of up-sampling modules, the down-sampling modules and the up-sampling modules each are followed by a residual neural network module; input the edges of the image to the preset number of down-sampling modules and the respective residual neural network modules behind the down-sampling modules sequentially to down-sample the edges of the object in the image, and memorize features of down-sampled
15 edges of the object in the image; and input the features of the down-sampled edges of the object in the image to the preset number of up-sampling modules and the respective residual neural network modules following the up-sampling modules sequentially to up-sample the features of the edges of the object in the image, and memorize up-sampled features of the edges of the object in the image to obtain de-noised edges of the object in the image.

20 **[0020]** In an embodiment, the apparatus further includes a line normalization unit. The line normalization unit is configured to acquiring a pre-trained line-width standardization model, where the line-width standardization model includes at least one of a wide network model and a flexible network model; and input the de-noised edges of the object in the image to the line-width standardization model, to even the de-noised edges of the object in the image in
25 thickness and color.

30 **[0021]** In an embodiment, the wide network model has the same network structure as the flexible network model, and the number of convolution cores in the wide network model is different from the number of convolution cores in the flexible network model. A first layer of the wide network model is a convolution kernel of $N \times N$, a last layer of the wide network model is a sigmoid function, and other layers of the wide network model each are a convolution kernel of $M \times M$. In addition, each convolution layer is followed by a

normalization function and an activation function, and N is greater than M.

[0022] In an embodiment, the vectorizing unit is configured to vector the edges of the object in the image using a preset vectorizing image processing tool. The preset vectorizing image processing tool includes a Potrace tool and an Imagemagick tool.

5 **[0023]** A computer apparatus is further provided according to the embodiments of the present disclosure. The computer apparatus includes a processor and a memory. The processor is configured to implement the method for modeling an element in a metaverse scenario according to the first aspect of the present disclosure when executing a computer program stored in the memory.

10 **[0024]** A computer-readable storage medium storing a computer program is further provided according to the embodiments of the present disclosure. The computer program, when being executed by a processor, implements the method for modeling an element in a metaverse scenario according to the first aspect of the present disclosure.

[0025] It can be seen from the above technical solutions that the embodiments of the present disclosure have the following advantages.

[0026] In the embodiments of the present disclosure, an image of a to-be-modeled object is acquired. Edge detection is performed on an object in the image to extract edges of the object in the image. The edges of the object in the image are vectorized to obtain a vector graphic of the edges of the object in the image. The vector lines in the vector graphic and the closed region defined by the vector lines are edited to obtain a model of the object in the image.

20 **[0027]** In the embodiments of the present disclosure, the computer apparatus directly acquires the image of the to-be-modeled object, performs edge detection on the image of the to-be-modeled object and vectorizes the image of the to-be-modeled object, so as to obtain the model, of the object in the metaverse scenario, editable by users. On the one hand, the efficiency of generating a model of an object in a metaverse scenario can be improved. On the other hand, the generated model of the object in the metaverse scenario is customizable and editable by users, meeting the personalized demands of users.

BRIEF DESCRIPTION OF THE DRAWINGS

30 **[0028]** Figure 1 is a schematic diagram showing architecture of a system for modeling an

element in a metaverse scenario according to an embodiment of the present disclosure;

[0029] Figure 2 is a schematic diagram showing a method for modeling an element in a metaverse scenario according to an embodiment of the present disclosure;

[0030] Figure 3 is a schematic diagram showing an image of a chair and edges of the chair according to an embodiment of the present disclosure;

[0031] Figure 4 shows details of step 202 in the embodiment shown in Figure 2 according to the present disclosure;

[0032] Figure 5 is a schematic diagram showing an edge detection model according to an embodiment of the present disclosure;

[0033] Figure 6 is a schematic diagram showing comparison between edges of the chair before de-noising and edges of the chair after de-noising according to an embodiment of the present disclosure;

[0034] Figure 7 is a schematic diagram showing comparison between edges of the chair before normalization and edges of the chair after normalization according to an embodiment of the present disclosure;

[0035] Figure 8 is a schematic diagram of a method for modeling an element in a metaverse scenario according to another embodiment of the present disclosure;

[0036] Figure 9 is a schematic diagram of a structural line extraction model in a U-Net mode according to an embodiment of the present disclosure;

[0037] Figure 10 is a schematic diagram of editing edges in a vector graphic according to an embodiment of the present disclosure;

[0038] Figure 11 is a schematic diagram of coloring a vector graphic according to an embodiment of the present disclosure; and

[0039] Figure 12 is a schematic diagram of an apparatus for modeling an element in a metaverse scenario according to an embodiment of the present disclosure.

DETAILED DESCRIPTION OF EMBODIMENTS

[0040] A method for modeling an element in a metaverse scenario is provided according to the embodiments of the present disclosure, to perform edge detection on an acquired image of

a to-be-modeled object and vector the acquired image of the to-be-modeled object, so as to obtain a model, of an object in the metaverse scenario, editable by users. On the one hand, the efficiency of generating the model of the object in the metaverse scenario can be improved. On the other hand, the generated model of the object in the metaverse scenario is customizable and editable by the users, meeting the personalized demands of the users.

[0041] In order to better understand the solutions according to the embodiments of the present disclosure by those skilled in the art, the technical solutions according to the embodiments of the present disclosure will be described clearly and completely as follows in conjunction with the drawings of the embodiments of the present disclosure. It is apparent that the described embodiments are only some embodiments rather than all the embodiments of the present disclosure. Other embodiments obtained by those skilled in the art without any creative efforts based on the embodiments of the present disclosure fall within the protection scope of the present disclosure.

[0042] The terms "first", "second", "third", "fourth" and the like in the specification, the claims, and the drawings of the embodiments of the present disclosure are intended to distinguish similar objects from each other rather than indicate a specific order or sequence. It should be understood that terms used in such manner are interchangeable when appropriate, such that the embodiments described herein can be implemented in an order different from the order shown or described herein. Moreover, the terms "include," "comprise", and any other variants thereof are intended to be non-exclusive. For example, a process, method, system, product, or apparatus that includes a series of steps or units is unnecessarily limited to those expressly listed steps or units, but may include other steps or units not expressly listed or inherent in such process, method, product, or apparatus.

[0043] In related technologies, generation of a model of an element in a metaverse scenario is time-consuming and labor-intensive. In order to solve the problem, a method for modeling an element in a metaverse scenario is provided according to the present disclosure, to improve the efficiency of generating a model of an object in the metaverse scenario.

[0044] In order to better implement the method for modeling an element in a metaverse scenario, a system for modeling an element in a metaverse scenario is provided according to the present disclosure. Referring to Figure 1, which is a schematic diagram of architecture of a system for modeling an element in a metaverse scenario according to an embodiment of the

present disclosure. The system includes at least one terminal apparatus 101 and a server 102. Different types of applications are installed on the terminal apparatus 101. For example, an online image processing application, a photo library application, and the like may be installed on the terminal apparatus 101. The terminal apparatus 101 may be an intelligent phone, a tablet computer, a notebook computer, a desktop computer, an intelligent vehicle apparatus, a wearable apparatus, or the like. The server 102 may be an independent physical server, or may be a server cluster including multiple physical servers or a distributed system, or may be a cloud server providing basic cloud computing services, such as a cloud service, a cloud database, cloud computer, a cloud function, cloud storage, a network service, cloud communication, a middleware service, a domain name service, a security service, a content delivery service, big data, and an artificial intelligence platform.

[0045] The method for modeling an element in a metaverse scenario may be performed by the terminal apparatus 101 and/or the server 102. In a case that the method is performed by the terminal apparatus 101, an image, of a to-be-modeled object, obtained by the terminal apparatus 101 in different types of mute applications is stored in the server. The terminal apparatus 101 acquires the image of the to-be-modeled object from the server 102 before processing the image of the to-be-modeled object. After acquiring the image of the to-be-modeled object from the server 102, the terminal apparatus 101 performs edge detection on an object in the image to extract edges of the object in the image, then vectorizes the edges of the object in the image to obtain a vector graphic of the edges of the object in the image, and finally edits vector lines in the vector graphic and a closed region defined by the vector lines to obtain a model of the object in the image.

[0046] Based on the method and the system for modeling an element in a metaverse scenario described above, reference is made to Figure 2. Figure 2 is a schematic diagram of a method for modeling an element in a metaverse scenario according to an embodiment of the present disclosure. The method may be performed by an intelligent apparatus. The intelligent apparatus may be the terminal apparatus 101 or the server 102. The method includes the following steps 201 to 204.

[0047] In step 201, an image of a to-be-modeled object is acquired.

[0048] In the related technology, a model of an element in a metaverse scenario is generated by manually drawing the material by a designer, which results in a low efficiency of obtaining

the material. Different from the related technology, in the embodiments of the present disclosure, the acquired image of the to-be-modeled object is processed as follows to generate the model of the object in the metaverse scenario.

[0049] Metaverse is a virtual digital world independent of the real world, in which a social platform with a strong sense of immersion is created with exclusive hardware devices by fusing virtual reality technology. In a metaverse scenario, based on the virtual digital world, interaction between users and the metaverse scenario necessitates some elements (models of objects) supporting the scenario.

[0050] It can be understood that the image of the to-be-modeled object is acquired before the image of the to-be-modeled object is processed. The image of the to-be-modeled object is a static image, a dynamic image, or the like, and is not limited in the embodiments of the present disclosure.

[0051] In practice, the intelligent apparatus may acquire the image of the to-be-modeled object from an image resource or a video resource pre-stored in a local space. The image of the to-be-modeled object is acquired from the image resource as follows. In a case that the image resource is a static image, the resource image is determined as the image of the to-be-modeled object. In a case that the image resource is a video library, the image of the to-be-modeled object is determined from various frames of images in the video library.

[0052] In an embodiment, the user, for obtaining a model of an object in the metaverse scenario, may submit an image of a to-be-modeled object. The intelligent apparatus receives a request for sending the image of the to-be-modeled object from the user. The request carries the image of the to-be-modeled object.

[0053] In step 202, edge detection is performed on an object in the image to extract edges of the object in the image.

[0054] In practice, the intelligent apparatus may call an edge detection model to perform edge detection on the object in the image, so as to extract the edges of the object in the image. For example, in Figure 3, the edge detection model performs edge detection on a chair in the image, to extract edges of the chair.

[0055] In the present disclosure, a process of extracting the edges of the object in the image using the edge detection model will be described in the following embodiments, and are not

repeated here.

[0056] In step 203, the edges of the object in the image are vectorized to obtain a vector graphic of the edges of the object in the image.

[0057] The edges, of the object in the image, obtained in step 202 generally form a bitmap, and the bitmap is not customizable, editable and modifiable by users. In order to cause the edges of the object in the image to be customizable and modifiable by the users, the edges of the object in the image are vectorized in the embodiments of the present disclosure to obtain a vector graphic of the edges of the object in the image. Vector lines in the vector graphic and a closed region defined by the vector lines are customizable and modifiable by the users.

[0058] A process of vectorizing the edges of the object in the image to obtain the vector graphic is described in the following embodiments, and is not repeated here.

[0059] In step 204, the vector lines in the vector graphic and the closed region defined by the vector lines are edited to obtain a model of the object in the image.

[0060] The vector lines in the vector graphic in step 203 and the closed region defined by the vector lines are customizable and modifiable by the users, so that the intelligent apparatus can edit the vector lines in the vector graphic and the closed region defined by the vector lines to obtain the model of the object in the image.

[0061] For example, a thickness or a length of each of the vector lines in the vector graphic may be changed or the closed region defined by the vector lines may be colored to obtain the model of the object in the image.

[0062] In the embodiments of the present disclosure, an image of a to-be-modeled object is acquired. Edge detection is performed on an object in the image to extract edges of the object in the image. The edges of the object in the image are vectorized to obtain a vector graphic of the edges of the object in the image. The vector lines in the vector graphic and the closed region defined by the vector lines are edited to obtain a model of the object in the image.

[0063] In the embodiments of the present disclosure, the intelligent apparatus performs edge detection on the image of the to-be-modeled object and vectorizes the image of the to-be-modeled object to obtain the vector graphic of the edges of the object in the to-be-modeled image. Then, the intelligent apparatus edits the vector lines in the vector graphic and the closed region defined by the vector lines to obtain the model of the object in

the image. In this way, on the one hand, the efficiency of modeling the material in the metaverse scenario can be improved. On the other hand, the vector graphic of the to-be-modeled object is customizable and modifiable by the users, meeting the personalized demands of the users.

5 **[0064]** Based on the embodiment shown in Figure 2, step 202 in Figure 2 is described in detail below. Referring to Figure 4, Figure 4 shows details of step 202 in the embodiment shown in Figure 2, and step 202 includes the following steps 401 to 403.

[0065] In step 401, an edge detection model is acquired, where the edge detection model includes an edge extraction module and an up-sampling module.

10 **[0066]** In an embodiment, the edge detection model includes the edge extraction module and the up-sampling module. The edge extraction module and the up-sampling module each include a convolutional network.

[0067] In step 402, the image is inputted to the edge extraction module to gradually extract edge features of the object in the image through the convolution network in the edge
15 extraction module.

[0068] The intelligent apparatus inputs the image of the to-be-modeled object to the edge extraction module to gradually extract the edge features of the object in the image through the convolution network in the edge extraction module.

[0069] In step 403, the edge features of the object in the image are inputted to the
20 up-sampling module to up-sample the edge features of the object in the image, and the edge features subjected to up-sampling are averaged to extract the edges of the object in the image.

[0070] In order to generate the edges of the object in the image based on the edge features of the object in the image, the intelligent apparatus inputs the edge features outputted by all convolution layers in the edge extraction module to the up-sampling module to up-sample the
25 edge features of the object in the image, so as to generate the edges of the object in the image.

[0071] A large number of convolution layers may result in a loss of a significant edge feature. Therefore, in the embodiments of the present disclosure, the up-sampling module averages all the outputted edge features to generate final edges of the object in the to-be-modeled image.

30 **[0072]** In order to facilitate understanding, Figure 5 shows a schematic diagram of an edge

detection model. The edge detection model includes an edge extraction module and an up-sampling module. The edge extraction module includes a convolution network, for example, multiple convolution cores each are of $W \times W$ shown in Figure 5, for performing a convolution operation on an inputted image, so as to gradually extract edges of an image in a picture. In order to ensure integrity of the edges, of the image, extracted from the picture, for each of the convolution kernels, an up-sampling unit follows the convolution kernel, and the up-sampling unit is configured to receive an edge feature, of the image, extracted by the convolution kernel and up-sample the edge feature. Multiple up-sampling units in Figure 5 form the up-sampling module in the embodiments of the present disclosure.

[0073] The up-sampling units in the embodiments of the present disclosure are divided into two types, namely, a first up-sampling unit and a second up-sampling unit. The first up-sampling unit and the second up-sampling unit are similar in structure and are different in the number of convolution kernels. In addition, the first up-sampling unit and the second up-sampling unit each have a structure including two layers, namely one convolution layer and one transposed convolution layer. For example, the first up-sampling unit includes a convolution of 1×1 , a Relu activation function and a transposed convolution of $s \times s$. A value of s is a size of a feature map of the inputted image, and the last convolution layer is provided with no activation function. The first up-sampling unit and the second up-sampling unit are applicable to different cases respectively. The first up-sampling unit is used in a case that a scale difference between a feature map and a real value of the edges of the image is greater than 2, and the second up-sampling unit is used in a case that the scale difference between the feature map and the real value of the edges of the image is equal to 2, facilitating generating the edges of the to-be-modeled object rapidly.

[0074] In an embodiment, a large number of layers of the up-sampling module may result in the loss of some significant edge features. In order to ensure the uniformity of the edges, of the image, outputted by the up-sampling module, the up-sampling module may further average the edge features subjected to up-sampling in the embodiments of the present disclosure, to generate the edges of the to-be-modeled object.

[0075] In the embodiments of the present disclosure, the edge detection model for extracting the edges, of the to-be-modeled object, in the image is described in detail, thereby improving the reliability of the process of extracting the edges of the object in the image.

[0076] Based on the embodiment shown in Figure 2, an experiment shows that the edges, of the image, extracted in step 202 usually include noise and the edges of the image are usually uniform in thickness, resulting in poor visibility of the image. Figure 6 is a schematic diagram showing a comparison between edges, of a chair, with noise and edges, of the chair, subjected to de-noising. Figure 7 is a schematic diagram showing a comparison between edges, of the chair, before normalization process and edges, of the chair, after normalization process.

[0077] To solve the problem, the following steps 801 to 806 may further be performed in the embodiments of the present disclosure to improve the image quality of the edges, of the image, extracted in step 202. Referring to Figure 8, Figure 8 shows a method for modeling an element in a metaverse scenario according to another embodiment of the present disclosure.

[0078] In step 801, an image of a to-be-modeled object is acquired.

[0079] In step 802, edge detection is performed on an object in the image to extract edges of the object in the image.

[0080] It should be noted that step 801 and step 802 in the embodiments of the present disclosure are similar to step 101 and step 102 in Figure 1 respectively, and are not repeated here.

[0081] In step 803, de-noising is performed on the edges of the object in the image to obtain de-noised edges of the object in the image.

[0082] In order to eliminate the noise existing in the edges, of the image, extracted in step 202, de-noising is performed on the edges of the image in the embodiments of the present disclosure to obtain the de-noised edges of the object in the image.

[0083] Specifically, in the embodiments of the present disclosure, a structural line extraction model is invoked to de-noise the edges of the object in the image, to eliminate the noise existing in the edges of the object in the image, so as to obtain the de-noised edges of the object in the image.

[0084] In an embodiment, the structural line extraction model may be a convolutional neural network. The intelligent apparatus calls the convolutional neural network to de-noise the edges of the object in the image, to obtain the de-noised edges of the object in the image.

[0085] The convolutional neural network includes a preset number of down-sampling modules and a preset number of up-sampling modules. In addition, the up-sampling modules

and the down-sampling modules each are followed by a residual neural network module. The convolutional neural network may adopt a U-Net mode as shown in Figure 9. The convolutional neural network includes down-sampling modules 901, residual neural network modules 902 and up-sampling modules 903. In addition, a corresponding residual neural network module 902 follows each of the down-sampling modules 901, and a corresponding residual neural network module 902 follows each of the up-sampling modules 903.

[0086] The edges of the object in the to-be-modeled image are inputted to each of the down-sampling module 901 and each of the residual neural network module 902 sequentially, then down-sampling is performed on the edges of the object in the image, and features of down-sampled edges of the object in the image are memorized to obtain a feature map having a size smaller than a size of the image of the to-be-modeled object. The up-sampling module 903 is configured to restore the feature map having a size smaller than the size of the image of the to-be-modeled object to obtain an image with the same size as the size of the image of the to-be-modeled object. A corresponding convolution kernel parameter of the up-sampling module is selected based on a convolution kernel parameter of the down-sampling module and up-sampling is repeatedly performed to ensure that the size of the feature image is restored to be equal to the size of the image of the to-be-modeled object.

[0087] It should be noted that a corresponding residual neural network module 902 follows each upper sampling module 903, and the residual neural network module 902 is configured to memorize features of the edges of the object in an up-sampled image, to avoid gradient explosion or gradient disappearance with an increase of the number of network layers of the structural line extraction model. In this way, the number of network layers of the structural line extraction model can be rapidly restored to the number of network layers of the features of the edges of the object in the original learned image, so as to ensure the completeness of the structural line extraction model.

[0088] In the structural line extraction model, there may be a cross line link between a down-sampling module 901 and an up-sampling module 903 respectively corresponding to feature maps that are equal in size. The cross line link is used to reduce information loss rapidly.

[0089] In step 804, normalization is performed on the de-noised edges of the object in the image, to even the de-noised edges of the object in the image in thickness and color.

[0090] After the de-noised edges of the object in the image are obtained in step 803, normalization is further performed on the de-noised edges of the object in the image in order to ensure that the lines are uniform in thickness and color.

[0091] In an embodiment, the intelligent apparatus may call a pre-trained line-width standardization model, and input the de-noised edges of the object in the image to the line-width standardization model to even the de-noised edges of the object in the image in thickness and color. The line-width standardization model includes at least one of a wide network model and a flexible network model.

[0092] In the embodiments of the present disclosure, the wide network model has the same network structure as the flexible network model, and the number of convolution cores in the wide network model is different from the number of convolution cores in the flexible network model. A first layer of the wide network model is a convolution kernel of $N \times N$, a last layer of the wide network model is a sigmoid function, and other layers of the wide network model each are a convolution kernel of $M \times M$. In addition, each convolution layer is followed by a normalization function and an activation function, and N is greater than M .

[0093] For facilitating understanding, an example is described below.

[0094] Assuming that for each of the wide network model and the flexible network model, the first layer is a convolution core of 9×9 and remaining layers each are a convolution core of 3×3 . The wide network model and the flexible network model are different only in that the wide network model is provided with 64 convolution cores while the flexible network model is provided with 32 convolution cores. In order to prevent the neural network from being affected by data distribution and ensure nonlinearity of the neural network, each convolution layer is followed by a normalization function and an activation function in the embodiments of the present disclosure. In order to cause an outputted result to be within a range (0, 1), the last layer of the wide network model and the last layer of the flexible network model each are set as a sigmoid function in the embodiments of the present disclosure.

[0095] It can be understood that the wide network model and flexible network model are trained before the edges of the image are inputted to the wide network model and/or the flexible network model. A data source for training in the embodiments of the present disclosure is normally a TUD-Berlin data set, a QuickDraw data set, a KanjiVG data set, a pattern of a manual receipt and the like.

[0096] In the embodiments of the present disclosure, a line inputted to the wide network model generally has a width within a range from 0.5 pixels to 10 pixels, and a line inputted to the flexible network model generally has a width with a range from 0.5 pixels to 3 pixels. A line having a width of 2 pixels is outputted after training.

5 **[0097]** In step 805, the edges of the object in the image are vectorized to obtain a vector graphic of the edges of the object in the image.

[0098] The edges, of the object, extracted from the image of the to-be-modeled object generally form a bitmap, and a resolution of the image generally changes greatly in a process of modifying the bitmap. To solve the problem, the edges of the object in the to-be-modeled
10 image are vectorized using a preset image processing tool in the embodiments of the present disclosure, to obtain the vector graphic of the edges of the object in the image. Vector lines in the vector graphic and a closed region defined by the vector lines are customizable and modifiable by users.

[0099] Assuming that a graphic of the edges of the object in the image is a picture in a sketch.png format, a bitmap in a pnm format may be obtained by using a command "convert
15 sketch.png sketch.pnm" in a Linux system, and then a final vector graphic is obtained by using a command "potrace sketch.pnm-s-o sketch.svg". In an enlarged view of the vector graphic, an edge of the object is smooth, without jagged pixels.

[0100] In step 806, vector lines in the vector graphic and a closed region defined by the
20 vector lines are edited to obtain a model of the object in the image.

[0101] In an embodiment, the intelligent apparatus may further zoom and modify the edges in the vector graphic using a vector graphic editor such as scratch-paint, and may further color each closed region. As shown in Figure 10 and Figure 11, Figure 10 is a schematic diagram of editing the edges in the vector graphic, and Figure 11 is a schematic diagram of coloring the
25 closed region defined by the vector lines in the vector graphic.

[0102] The method for modeling an element in a metaverse scenario according to the embodiments of the present disclosure is described above, and an apparatus for modeling an element in a metaverse scenario according to the embodiments of the present disclosure is described below. Referring to Figure 12, which shows an apparatus for modeling an element
30 in a metaverse scenario according to an embodiment of the present disclosure. The apparatus

includes an acquisition unit 1201, an edge detection unit 1202, a vectorizing unit 1203, and an editing unit 1204.

[0103] The acquisition unit 1201 is configured to acquire an image of a to-be-modeled object.

5 **[0104]** The edge detection unit 1202 is configured to perform edge detection on an object in the image to extract edges of the object in the image.

[0105] The vectorizing unit 1203 is configured to vector the edges of the object in the image to obtain a vector graphic of the edges of the object in the image.

10 **[0106]** The editing unit 1204 is configured to edit vector lines in the vector graphic and a closed region defined by the vector lines to obtain a model of the object in the image.

[0107] In an embodiment, the edge detection unit 1202 is configured to acquire an edge detection model, where the edge detection model includes an edge extraction module and an up-sampling module; input the image to the edge extraction module to gradually extract edge features of the object in the image through a convolutional network in the edge extraction module; and input the edge features of the object in the image to the up-sampling module to up-sample the edge features of the object in the image, and average the up-sampled edge features to extract the edges of the object in the image.

[0108] In an embodiment, the apparatus further includes a de-noising unit 1205.

20 **[0109]** The de-noising unit 1205 is configured to de-noise the edges of the object in the image to obtain de-noised edges of the object in the image.

[0110] In an embodiment, the de-noising unit 1205 is configured to: acquire a structural line extraction model, where the structural line extraction model includes a preset number of down-sampling modules and a preset number of up-sampling modules, and the down-sampling modules and the up-sampling modules each are followed by a residual neural network module; input the edges of the image to the preset number of down-sampling modules and respective residual neural network modules behind the down-sampling modules sequentially to down-sample the edges of the object in the image, and memorize features of down-sampled edges of the object in the image; and input the features of the down-sampled edges of the object in the image to the preset number of up-sampling modules and the respective residual neural network modules following the up-sampling modules sequentially

to up-sample the features of the edges of the object in the image, and memorize up-sampled features of the edges of the object in the image to obtain de-noised edges of the object in the image.

[0111] In an embodiment, the apparatus further includes a line normalization unit 1206.

5 **[0112]** The line normalization unit 1206 is configured to acquiring a pre-trained line-width standardization model, where the line-width standardization model includes at least one of a wide network model and a flexible network model; and input the de-noised edges of the object in the image to the line-width standardization model, to even the de-noised edges of the object in the image in thickness and color.

10 **[0113]** In an embodiment, the wide network model has the same network structure as the flexible network model, and the number of convolution cores in the wide network model is different from the number of convolution cores in the flexible network model. A first layer of the wide network model is a convolution kernel of $N \times N$, a last layer of the wide network model is a sigmoid function, and other layers of the wide network model each are a
15 convolution kernel of $M \times M$. In addition, each convolution layer is followed by a normalization function and an activation function, and N is greater than M .

[0114] In an embodiment, the vectorizing unit 1203 is configured to vector the edges of the object in the image using a preset vectorizing image processing tool. The preset vectorizing image processing tool includes a Potrace tool and an Imagemagick tool.

20 **[0115]** It should be noted that functions of the units in the embodiments of the present disclosure are similar to the description of Figure 1 to Figure 11, and are not repeated here.

[0116] In the embodiments of the present disclosure, the edge detection unit 1202 performs edge detection on the image of the to-be-modeled object and the vectorizing unit 1203 vectorizes the image of the to-be-modeled object, to obtain the vector graphic of the edges of
25 the object in the to-be-modeled image. Then, the editing unit 1204 edits the vector lines in the vector graphic and the closed region defined by the vector lines to obtain the model of the object in the image. In this way, on the one hand, the efficiency of modeling the material in the metaverse scenario can be improved. On the other hand, the vector graphic of the to-be-modeled object is customizable and modifiable by the users, meeting the personalized
30 demands of the users.

[0117] The apparatus for modeling an element in a metaverse scenario according to the embodiments of the present disclosure is described above from a perspective of modular functional entities, and a computer apparatus according to the embodiments of the present disclosure is described below from a perspective of hardware processing.

5 **[0118]** The computer apparatus is configured to implement functions of the apparatus for modeling an element in a metaverse scenario. A computer apparatus according to an embodiment of the present disclosure includes a processor and a memory.

[0119] The memory is configured to store a computer program. The processor is configured to, when executing the computer program stored in the memory, implement the following
10 steps of: acquiring an image of a to-be-modeled object; performing edge detection on an object in the image to extract edges of the object in the image; vectorizing the edges of the object in the image to obtain a vector graphic of the edges of the object in the image; and editing vector lines in the vector graphic and a closed region defined by the vector lines to obtain a model of the object in the image.

15 **[0120]** In some embodiments of the present disclosure, the processor may further be configured to implement the following steps of: acquiring an edge detection model, where the edge detection model includes an edge extraction module and an up-sampling module; inputting the image to the edge extraction module to gradually extract edge features of the object in the image through a convolutional network in the edge extraction module; and
20 inputting the edge features of the object in the image to the up-sampling module to up-sample the edge features of the object in the image, and averaging the up-sampled edge features to extract the edges of the object in the image.

[0121] In some embodiments of the present disclosure, the processor, after extracting the edges of the object in the image and before vectorizing the edges of the object in the image,
25 may further be configured to implement the following step of: de-noising the edges of the object in the image to obtain de-noised edges of the object in the image.

[0122] In some embodiments of the present disclosure, the processor may further be configured to implement the following steps of: acquiring a structural line extraction model, where the structural line extraction model includes a preset number of down-sampling
30 modules and a preset number of up-sampling modules, and the down-sampling modules and the up-sampling modules each are followed by a residual neural network module; inputting

the edges of the image to the preset number of down-sampling modules and the respective residual neural network modules following the down-sampling modules sequentially to down-sample the edges of the object in the image, and memorizing features of down-sampled edges of the object in the image; and inputting the features of the down-sampled edges of the object in the image to the preset number of up-sampling modules and the respective residual neural network modules following the up-sampling modules sequentially to up-sample the features of the edges of the object in the image, and memorizing up-sampled features of the edges of the object in the image to obtain de-noised edges of the object in the image.

[0123] In some embodiments of the present disclosure, the processor, after obtaining the de-noised edges of the object in the image, may further be configured to implement the following step of: normalizing the de-noised edges of the object in the image, to even the de-noised edges of the object in the image in thickness and color.

[0124] In some embodiments of the present disclosure, the processor may further be configured to implement the following steps of: acquiring a pre-trained line-width standardization model, where the line-width standardization model includes at least one of a wide network model and a flexible network model; and inputting the de-noised edges of the object in the image to the line-width standardization model, to even the de-noised edges of the object in the image in thickness and color.

[0125] In some embodiments of the present disclosure, the wide network model has the same network structure as the flexible network model, and the number of convolution cores in the wide network model is different from the number of convolution cores in the flexible network model. A first layer of the wide network model is a convolution kernel of $N*N$, a last layer of the wide network model is a sigmoid function, and other layers of the wide network model each are a convolution kernel of $M*M$. In addition, each convolution layer is followed by a normalization function and an activation function, and N is greater than M .

[0126] In some embodiments of the present disclosure, the processor may further be configured to implement the following step of: vectorizing the edges of the object in the image using a preset vectorizing image processing tool. The preset vectorizing image processing tool includes a Potrace tool and an Imagemagick tool.

[0127] It can be understood that when executing the computer program, the processor in the computer apparatus as described above can implement the functions of the units in the

apparatus embodiments, which is not repeated herein. For example, the computer program may be divided into one or more modules/units. The one or more modules/units are stored in the memory and executed by the processor to implement the embodiments of the present disclosure. The one or more modules/units may be a series of computer program instruction
5 segments capable of completing specific functions. The instruction segments are used to describe a process that the computer program is executed in the apparatus. For example, the computer program may be divided into the units in the apparatus, and each of the units can realize the specific functions of the apparatus.

[0128] The computer apparatus may be a computing apparatus such as a desktop computer,
10 a notebook computer, a handheld computer and a cloud server. The computer apparatus may include, but not limited to a processor and a memory. It can be understood by those skilled in the art that the processor and the memory are only examples of the computer apparatus, and are not intended to limit the computer apparatus. The computer apparatus may include more or less components, or a combination of some components, or a component different from the
15 processor and the memory. For example, the computer apparatus may further include an input/output device, a network access device, a bus, and the like.

[0129] The processor may be a central processing unit (CPU), or other general-purpose processor, a digital signal processor (DSP), an application specific integrated circuit (ASIC), a field-programmable gate array (FPGA) or other programmable logic device, a discrete gate or
20 transistor logic device, a discrete hardware component, or the like. The general-purpose processor may be a microprocessor or any conventional processor. The processor is a control center of the computer apparatus, and various parts of the computer apparatus are connected via various interfaces and wirings.

[0130] The memory is configured to store the computer program and/or module, and the
25 processor implements various functions of the computer apparatus by running or executing the computer program and/or module stored in the memory and calling data stored in the memory. The memory includes a program storage area and a data storage area. The program storage area stores an operating system and an application program required by at least one function, and the like. The data storage area stores data created based on the use of a terminal,
30 and the like. In addition, the memory may include a high-speed random access memory, and may further include a non-volatile memory, such as a hard disk, an internal memory, a plug-in

hard disk, a smart media card (SMC), a secure digital (SD) card, a flash card, at least one disk storage device, a flash device or other volatile solid-state storage devices.

[0131] A computer-readable storage medium is further provided according to the embodiments of the present disclosure. The computer-readable storage medium is used to implement the functions of the apparatus for modeling an element in a metaverse scenario. The computer-readable storage medium stores a computer program. When the computer program is executed by a processor, the processor is configured to implement the following steps of: acquiring an image of a to-be-modeled object; performing edge detection on an object in the image to extract edges of the object in the image; vectorizing the edges of the object in the image to obtain a vector graphic of the edges of the object in the image; and editing vector lines in the vector graphic and a closed region defined by the vector lines to obtain a model of the object in the image.

[0132] In some embodiments of the present disclosure, when the computer program is executed by the processor, the processor is further configured to implement the following steps of: acquiring an edge detection model, where the edge detection model includes an edge extraction module and an up-sampling module; inputting the image to the edge extraction module to gradually extract edge features of the object in the image through a convolutional network in the edge extraction module; and inputting the edge features of the object in the image to the up-sampling module to up-sample the edge features of the object in the image, and averaging the up-sampled edge features to extract the edges of the object in the image.

[0133] In some embodiments of the present disclosure, when the computer program is executed by the processor, the processor, after extracting the edges of the object in the image and before vectorizing the edges of the object in the image, is further configured to implement the following step of: de-noising the edges of the object in the image to obtain de-noised edges of the object in the image.

[0134] In some embodiments of the present disclosure, when the computer program is executed by the processor, the processor is further configured to implement the following steps of: acquiring a structural line extraction model, where the structural line extraction model includes a preset number of down-sampling modules and a preset number of up-sampling modules, and the down-sampling modules and the up-sampling modules each are provided by a residual neural network module; inputting the edges of the image to the preset

number of down-sampling modules and respective residual neural network modules following the down-sampling modules sequentially to down-sample the edges of the object in the image, and memorizing features of down-sampled edges of the object in the image; and inputting the features of the down-sampled edges of the object in the image to the preset number of up-sampling modules and respective residual neural network modules following the up-sampling modules sequentially to up-sample the features of the edges of the object in the image, and memorizing up-sampled features of the edges of the object in the image to obtain de-noised edges of the object in the image.

[0135] In some embodiments of the present disclosure, when the computer program is executed by the processor, the processor, after obtaining the de-noised edges of the object in the image, is further configured to implement the following step of: normalizing the de-noised edges of the object in the image, to even the de-noised edges of the object in the image in thickness and color.

[0136] In some embodiments of the present disclosure, when the computer program is executed by the processor, the processor is further configured to implement the following steps of: acquiring a pre-trained line-width standardization model, where the line-width standardization model includes at least one of a wide network model and a flexible network model; and inputting the de-noised edges of the object in the image to the line-width standardization model, to even the de-noised edges of the object in the image in thickness and color.

[0137] In some embodiments of the present disclosure, the wide network model has the same network structure as the flexible network model, and the number of convolution cores in the wide network model is different from the number of convolution cores in the flexible network model. A first layer of the wide network model is a convolution kernel of $N \times N$, a last layer of the wide network model is a sigmoid function, and other layers of the wide network model each are a convolution kernel of $M \times M$. In addition, each convolution layer is followed by a normalization function and an activation function, and N is greater than M .

[0138] In some embodiments of the present disclosure, when the computer program is executed by the processor, the processor is further configured to implement the following step of: vectorizing the edges of the object in the image using a preset vectorizing image processing tool. The preset vectorizing image processing tool includes a Potrace tool and an

Imagemagick tool.

[0139] It can be understood that if the integrated unit is implemented in a form of a software functional unit and sold or used as an independent product, the integrated unit may be stored in a computer-readable storage medium. Based on such understanding, the technical solution according to the embodiments of the present disclosure is essentially or part of the technical solution that make contribution to the conventional technology or all or part of the technical solution may be embodied in a form of a software product, and the computer software product is stored in a storage medium. The computer software product includes several instructions for enabling a computer device (which may be a personal computer, a server, or a network device, etc.) to execute all or part of the steps of the method described in the various embodiments of the present disclosure. The aforementioned storage medium includes: a U disk, a mobile hard disk, a read-only memory (ROM), a random access memory (RAM), a magnetic disk, an optical disk, or other medium that can store program codes.

[0140] In the several embodiments provided in the present disclosure, it should be understood that the disclosed system, apparatus and method can be implemented in other manners. For example, the apparatus embodiments described above are only illustrative, and the division of the units is based on logical function. In practice, the units may be divided in another way. For example, multiple units or components may be combined or integrated into another system, and some features may be ignored or not implemented. In addition, coupling, direct coupling, or communication connection shown or discussed may be indirect coupling or communication connection via some interfaces, devices or units, and may be in electrical, mechanical or in other forms.

[0141] The units described as separate components may be or may be not separated physically. The components shown as units may be or may be not physical units, that is, may be located at a same place or may be distributed on multiple network units. Some or all of the units may be selected according to actual needs to achieve the objectives of the technical solutions according to the present disclosure.

[0142] In addition, the functional units in the embodiments of the present disclosure may be integrated into a same processing unit, or each of the units may exist alone physically, or two or more of the units may be integrated into a same unit. The above integrated units may be implemented in a form of hardware or software functional units.

[0143] The foregoing embodiments are merely intended for describing the technical solutions according to the present disclosure rather than limiting the present disclosure. Although the present disclosure is described in detail with reference to the foregoing embodiments, it should be understood by those skilled in the art that modifications can be made to the technical solutions described in the foregoing embodiments, or equivalent substitutions can be made to some technical features in the technical solutions. Such modifications or substitutions do not cause the essence of corresponding technical solutions to depart from the spirit and the scope of the technical solutions according to the embodiments of the present disclosure.

[0144] In the embodiments of the present disclosure, the computer apparatus directly acquires the image of the to-be-modeled object, performs edge detection on the image of the to-be-modeled object and vectorizes the image of the to-be-modeled object, so as to obtain the model, of the object in the metaverse scenario, editable by users, meeting the demand for adding an element to a 2D metaverse scenario. On the one hand, the efficiency of generating a model of an object in a metaverse scenario can be improved. On the other hand, the generated model of the object in the metaverse scenario is customizable and editable by users, meeting the personalized demands of users.

CLAIMS

1. A method for modeling an element in a metaverse scenario, comprising:

acquiring an image of a to-be-modeled object;

5 performing edge detection on an object in the image to extract edges of the object in the image;

vectorizing the edges of the object in the image to obtain a vector graphic of the edges of the object in the image; and

10 editing vector lines in the vector graphic and a closed region defined by the vector lines, to obtain a model of the object in the image.

2. The method according to claim 1, wherein the performing edge detection on an object in the image to extract edges of the object in the image comprises:

acquiring an edge detection model, wherein the edge detection model comprises an edge extraction module and an up-sampling module;

15 inputting the image to the edge extraction module to gradually extract edge features of the object in the image through a convolutional network in the edge extraction module; and

inputting the edge features of the object in the image to the up-sampling module to up-sample the edge features of the object in the image, and averaging the up-sampled edge features to extract the edges of the object in the image.

20 3. The method according to claim 1, wherein after the edges of the object in the image are extracted and before the edges of the object in the image are vectorized, the method further comprises:

de-noising the edges of the object in the image to obtain de-noised edges of the object in the image.

25 4. The method according to claim 3, wherein the de-noising the edges of the object in the image to obtain de-noised edges of the object in the image comprises:

acquiring a structural line extraction model, wherein the structural line extraction model comprises a preset number of down-sampling modules and a preset number of up-sampling modules, and the down-sampling modules and the up-sampling modules each are followed by

a residual neural network module;

inputting the edges of the image to the preset number of down-sampling modules and the respective residual neural network modules following the down-sampling modules sequentially to down-sample the edges of the object in the image, and memorizing features of the down-sampled edges of the object in the image; and

inputting the features of the down-sampled edges of the object in the image to the preset number of up-sampling modules and the respective residual neural network modules following the up-sampling modules sequentially to up-sample the features of the edges of the object in the image, and memorizing the up-sampled features of the edges of the object in the image to obtain the de-noised edges of the object in the image.

5. The method according to claim 3, wherein after the de-noised edges of the object in the image are obtained, the method further comprises:

normalizing the de-noised edges of the object in the image to even the de-noised edges of the object in the image in thickness and color.

6. The method according to claim 5, wherein the normalizing the de-noised edges of the object in the image to even the de-noised edges of the object in the image in thickness and color comprises:

acquiring a pre-trained line-width standardization model, wherein the line-width standardization model comprises at least one of a wide network model and a flexible network model; and

inputting the de-noised edges of the object in the image to the line-width standardization model, to even the de-noised edges of the object in the image in thickness and color.

7. The method according to claim 6, wherein the wide network model has the same network structure as the flexible network model, and the number of convolution cores in the wide network model is different from the number of convolution cores in the flexible network model; and

a first layer of the wide network model is a convolution kernel of $N \times N$, a last layer of the wide network model is a sigmoid function, other layer of the wide network model is a convolution kernel of $M \times M$, and each convolution layer is followed by a normalization function and an activation function, wherein N is greater than M .

8. The method according to any one of claims 1 to 7, wherein the vectorizing the edges of the object in the image comprises:

vectorizing the edges of the object in the image using a preset vectorizing image processing tool, wherein the preset vectorizing image processing tool comprises a Potrace tool
5 and an Imagemagick tool.

9. A computer apparatus, comprising:

a memory; and

a processor configured to execute a computer program stored in the memory to implement the method according to any one of claims 1 to 8.

10 10. A computer-readable storage medium storing a computer program, wherein the computer program, when being executed by a processor, implements the method according to any one of claims 1 to 8.