

DATA PROCESSING METHOD AND APPARATUS, AND DEVICE AND READABLE STORAGE MEDIUM

[0001] This application claims priority to Chinese Patent Application No. 202111123508.7, entitled "DATA PROCESSING METHOD AND APPARATUS, DEVICE, AND READABLE STORAGE MEDIUM" filed with the China National Intellectual Property Administration on September 24, 2021, which is incorporated by reference in its entirety.

FIELD OF THE TECHNOLOGY

[0002] The present disclosure relates to the field of computer technologies, and in particular, to a data processing method, a data processing apparatus, a device, and a readable storage medium.

BACKGROUND OF THE DISCLOSURE

[0003] With rapid development of technologies related to electronic devices and popularization of the Internet, the game industry depending on electronic devices has been developed by leaps and bounds. Especially when smart terminals such as smart phones and tablet computers are developed, the game industry has more prominent potential.

[0004] In order to provide a user with an immersive experience, a virtual object (for example, a virtual animation object) may be created in a cloud game according to a human body image of the user, and is displayed in the cloud game. That is, the user is placed in a virtual cloud game scenario by using the virtual object, so as to provide the user with the immersive experience in the cloud game. In general, in this process, after a terminal acquires a user image by a camera, a portrait of the user is directly recognized and extracted by the terminal, to obtain and display the corresponding virtual object.

[0005] The computing capability of the terminal is not high, which is very likely to bring low image recognition efficiency, resulting in a long delay for transmitting a portrait recognition result to a cloud. As a result, there is a delay of displaying the virtual object in the game, so that a virtual behavior of the virtual object displayed in the game does not match a current behavior of the user.

SUMMARY

[0006] Embodiments of the present disclosure provide a data processing method, a data processing apparatus, a device, and a readable storage medium, which can reduce an image transmission delay and improve image recognition efficiency.

[0007] An aspect of the embodiments of the present disclosure provides a data processing method. The method is performed by a computer device. The method includes:

obtaining first image data transmitted by a first client, and storing the first image data to a reception queue, the first image data being image data comprising an object and obtained by the first client when running a cloud application;

performing image recognition processing on the first image data in the reception queue, and storing, during the image recognition processing on the first image data, second image data to the reception queue to obtain an updated reception queue, the second image data being image data continuously obtained from the first client; and

transmitting, in a case that a first object region of the object in the first image data is extracted through the image recognition processing, first object image data in the first object region to a target cloud application server, and at the same time performing image recognition processing on second image data having a latest receiving timestamp in the updated reception queue, the target cloud application server being configured to perform rendering on the first object image data to obtain rendered data, and transmit the rendered data to the first client.

[0008] An aspect of the embodiments of the present disclosure provides a data processing apparatus. The apparatus is deployed on a computer device. The apparatus includes:

a data obtaining module, configured to obtain first image data transmitted by a first client, and store the first image data to a reception queue, the first image data being image data comprising an object and obtained by the first client when running a cloud application;

an image recognition module, configured to perform image recognition processing on the first image data in the reception queue;

a queue update module, configured to store, during the image recognition processing on the first image data, second image data to the reception queue to obtain an updated reception queue, the second image data being image data continuously obtained from the first client; and

a region transmission module, configured to transmit, in a case that a first object region of the object in the first image data is extracted through the image recognition processing,

first object image data in the first object region to a target cloud application server, the target cloud application server being configured to perform rendering on the first object image data to obtain rendered data, and transmit the rendered data to the first client.

[0009] The region transmission module is further configured to at the same time perform image recognition processing on second image data with a latest receiving timestamp in the updated reception queue.

[0010] Another aspect of the embodiments of the present disclosure provides a data processing method. The method is performed by a computer device. The method includes:

receiving first object image data transmitted by a service server, and storing the first object image data to a first buffer whose working mode is a storage mode in a buffer set, the first object image data being image data in a first object region, the first object region being a region of an object in first image data and being obtained by the service server performing image recognition processing on the first image data, the first image data being transmitted by a first client to the service server, and the first image data being image data comprising the object and obtained by the first client when running a cloud application;

when a second buffer whose working mode is a read mode in the buffer set does not comprise unprocessed object image data, adjusting the working mode of the first buffer to the read mode and adjusting the working mode of the second buffer to the storage mode, reading the first object region data from the first buffer whose working mode is the read mode, and performing rendering processing on the first object region data;

receiving, during the rendering processing on the first object image data, second object image data transmitted by the service server, and storing the second object image data to the second buffer whose working mode is the storage mode, the second object image data being image data in a second object region, the second object region being obtained by the service server by performing image recognition processing on second image data after extracting the first object region, the second object region being a region of the object in the second image data, the second image data being image data having a latest receiving timestamp and obtained by the service server from an updated reception queue when the first object region is extracted, and second image data in the updated reception queue being continuously obtained by the service server from the first client during the image recognition processing on the first image data; and

when rendered data corresponding to the first object image data is obtained, adjusting the working mode of the first buffer to the storage mode and adjusting the working mode

of the second buffer to the read mode, reading the second object image data from the second buffer whose working mode is the read mode, and performing rendering processing on the second object image data.

[0011] An aspect of the embodiments of the present disclosure provides another data processing apparatus. The apparatus is deployed on a computer device. The apparatus includes:

a region storage module, configured to receive first object image data transmitted by a service server, and store the first object image data to a first buffer whose working mode is a storage mode in a buffer set, the first object image data being image data in a first object region, the first object region being a region of an object in first image data and being obtained by the service server performing image recognition processing on the first image data, the first image data being transmitted by a first client to the service server, and the first image data being image data comprising the object and obtained by the first client when running a cloud application;

a region rendering module, configured to, when a second buffer whose working mode is a read mode in the buffer set does not comprise unprocessed object image data, adjust the working mode of the first buffer to the read mode and adjust the working mode of the second buffer to the storage mode, read the first object image data from the first buffer whose working mode is the read mode, and perform rendering processing on the first object region;

a region receiving module, configured to receive, during the rendering processing on the first object region, second object image data transmitted by the service server, and store the second object image data to the second buffer whose working mode is the storage mode, the second object image data being image data in a second object region, the second object region being obtained by the service server by performing image recognition processing on second image data after extracting the first object region, the second object region being a region of the object in the second image data, the second image data being image data having a latest receiving timestamp and obtained by the service server from an updated reception queue when the first object region is extracted, and second image data in the updated reception queue being continuously obtained by the service server from the first client during the image recognition processing on the first image data; and

a status adjustment module, configured to, when rendered data corresponding to the first object image data is obtained, adjust the working mode of the first buffer to the storage mode and adjust the working mode of the second buffer to the read mode, read the second object image data from the second buffer whose working mode is the read mode, and perform rendering processing on the second object image data.

[0012] An aspect of the embodiments of the present disclosure provides a computer device, including a processor and a memory.

[0013] The memory stores a computer program. When the computer program is executed by the processor, the processor performs the method in the embodiments of the present disclosure.

[0014] An aspect of the embodiments of the present disclosure provides a computer-readable storage medium. The computer-readable storage medium stores a computer program. The computer program includes program instructions. The program instructions are executed by a processor to perform the method in the embodiments of the present disclosure.

[0015] An aspect of the present disclosure provides a computer program product or computer program. The computer program product or computer program includes computer instructions. The computer instructions are stored in a computer-readable storage medium. A processor of a computer device reads the computer instructions from the computer-readable storage medium. The processor executes the computer instructions to enable the computer device to perform the method provided in an aspect of the embodiments of the present disclosure.

[0016] In the embodiment of the present disclosure, when obtaining the first image data including the object, a client (for example, the first client) may transmit the first image data to a related computer device (for example, the service server), the service server performs image recognition processing, instead of requiring the client to perform the image recognition locally. In this way, image recognition processing may be performed on the first image data by using the service server with a high computing capability, so that image recognition efficiency and a definition can be improved. In addition, in the present disclosure, the service server may store the received first image data to the reception queue, continuously obtain the second image data from the first client during the image recognition processing of the first image data, and store the second image data to the reception queue to obtain the updated reception queue. That is, in the present disclosure, the service server does not stop receiving the second image data during the image recognition processing on the first image data, and may implement image processing and image reception at the same time by using the reception queue. In this way, an image transmission delay can be reduced. Further, when the service server extracts, through image recognition processing, the first object region of the object in the first image data, the service server may transmit the first object image data in the first object region to the target cloud application server, and the target cloud application server performs rendering, and transmits the rendered data to the first client. In this way, displaying may be performed in the cloud application. In addition, after extracting the first object region, the service server may obtain the second image data with the latest receiving

timestamp in the reception queue, and continue to process the second image data. It can be seen that in the present disclosure, image data is not recognized one by one according to a time sequence of receiving timestamps. Instead, image data with a latest receiving timestamp is obtained from the reception queue for processing after image recognition processing is performed on specific image data, so that image data recognition efficiency may be improved. In addition, since the image data with the latest receiving timestamp is acquired according to a current behavior of the object, when image recognition and displaying are performed on the image data with the latest receiving timestamp, the current behavior of the object is matched. In summary, according to the present disclosure, the image recognition efficiency may be improved, the image transmission delay may be reduced, and it may be ensured that a virtual behavior of a virtual object displayed in the cloud application matches a current behavior status of the object.

BRIEF DESCRIPTION OF THE DRAWINGS

[0017] In order to describe the technical solutions in the embodiments of the present disclosure or the related art more clearly, the drawings required to be used in descriptions about the embodiments or the related art will be introduced briefly below. Apparently, the drawings in the descriptions below are merely some embodiments of the present disclosure. A person of ordinary skill in the art may further obtain other drawings according to these drawings without creative work.

[0018] FIG. 1 is a diagram of a network architecture according to an embodiment of the present disclosure.

[0019] FIG. 2a is a schematic diagram of a scenario according to an embodiment of the present disclosure.

[0020] FIG. 2b is a schematic diagram of a scenario according to an embodiment of the present disclosure.

[0021] FIG. 3 is a schematic flowchart of a data processing method according to an embodiment of the present disclosure.

[0022] FIG. 4 is a schematic diagram of frame skipping processing according to an embodiment of the present disclosure.

[0023] FIG. 5 is a schematic flowchart of a data processing method according to an embodiment of the present disclosure.

[0024] FIG. 6 is a schematic diagram of a part fusion scenario according to an embodiment of the present disclosure.

[0025] FIG. 7 is a schematic flowchart of transmitting first object image data to a target cloud application server according to an embodiment of the present disclosure.

[0026] FIG. 8 is a schematic flowchart of a data processing method according to an embodiment of the present disclosure.

[0027] FIG. 9 is a schematic diagram of changing statuses of two buffers according to an embodiment of the present disclosure.

[0028] FIG. 10 is a diagram of a system architecture according to an embodiment of the present disclosure.

[0029] FIG. 11 is a schematic diagram of a system process according to an embodiment of the present disclosure.

[0030] FIG. 12 is a diagram of an interaction process according to an embodiment of the present disclosure.

[0031] FIG. 13 is a schematic diagram of a structure of a data processing apparatus according to an embodiment of the present disclosure.

[0032] FIG. 14 is a schematic diagram of a structure of another data processing apparatus according to an embodiment of the present disclosure.

[0033] FIG. 15 is a schematic diagram of a structure of a computer device according to an embodiment of the present disclosure.

DESCRIPTION OF EMBODIMENTS

[0034] The technical solutions in the embodiments of the present disclosure will be described clearly and completely below with reference to the accompanying drawings in the embodiments of the present disclosure. Clearly, the described embodiments are not all but merely some embodiments of the present disclosure. All other embodiments obtained by a person of ordinary skill in the art based on the embodiments of the present disclosure without creative work shall fall within the protection scope of the present disclosure.

[0035] FIG. 1 is a diagram of a network architecture according to an embodiment of the present disclosure. As shown in FIG. 1, the network architecture may include a service server 1000, a terminal device cluster, and a cloud application server cluster 10000. The terminal device cluster may include one or more terminal devices. A quantity of the terminal devices is not limited herein. As shown in FIG. 1, the terminal devices may include a terminal device 100a, a terminal device 100b, a terminal device 100c, ..., and a terminal device 100n. As shown in FIG. 1, each of the

terminal device 100a, the terminal device 100b, the terminal device 100c, ..., and the terminal device 100n may establish a network connection with the service server 1000, such that each terminal device may perform data interaction with the service server 1000 through the network connection. The cloud application server cluster 10000 may include one or more cloud application servers. A quantity of cloud application servers is not limited herein. As shown in FIG. 1, the cloud application servers may include a cloud application server 10001, a cloud application server 10002, ..., and a cloud application server 1000n. As shown in FIG. 1, each of the cloud application server 10001, the cloud application server 10002, ..., and the cloud application server 1000n may establish a network connection with the service server 1000, such that each cloud application server may perform data interaction with the service server 1000 through the network connection. Each cloud application server may be a cloud application server. One terminal device may correspond to one cloud application server (a plurality of terminal devices may correspond to a same cloud application server). When the terminal device runs a cloud application, the corresponding cloud application server provides a corresponding function service (for example, a computing service) for the terminal device. For example, when the cloud application is a cloud game application, the cloud application server may be a cloud game server. When the terminal device runs the cloud game application, the corresponding cloud game server provides a corresponding function service for the terminal device.

[0036] It may be understood that each terminal device shown in FIG. 1 may be installed with a cloud application, and when running the cloud application, may perform data interaction with the service server 1000 shown in FIG. 1, such that the service server 1000 may receive service data from each terminal device.

[0037] The cloud application may include an application with a function of displaying data information such as a text, an image, audio, or a video. The cloud application may be an entertainment application (for example, a game application). The entertainment application may be used for game entertainment of a user. In the present disclosure, the service server 1000 may obtain the service data according to the cloud application. For example, the service data may be image data (which may be referred to as first image data) that is acquired by the terminal device through a camera component and that includes the user (which may be referred to as an object).

[0038] Then, after obtaining the first image data, the service server 1000 may store the first image data to a reception queue, and then obtain the first image data from the reception queue. The service server 1000 may perform image recognition processing on the first image data. It is to be understood that after obtaining the first image data and transmitting the first image data to the

service server 1000, the terminal device may continuously obtain image data (which may be referred to as second image data) including the object. The service server 1000 may also continuously obtain the second image data obtained by the terminal device from the terminal device during the image recognition processing on the first image data. Similar to the first image data, the service server 1000 may store the second image data to the reception queue, thereby obtaining an updated reception queue including one or more pieces of second image data. It is to be understood that when the first image data is the 1st piece of image data received by the service server 1000, the service server 1000 may not store the first image data to the reception queue. The service server 1000 may directly perform image recognition processing on the first image data, continuously obtain the second image data (that is, the 2nd piece of image data, the 3rd piece of image data, the 4th piece of image data... after the 1st piece of image data) acquired by the terminal device from the terminal device during image recognition processing, and store the second image data to the reception queue.

[0039] After the service server 1000 extracts, through the image recognition processing, a region (which may be referred to as a first object region) of the object in the first image data, the service server 1000 may obtain image data (which may be referred to as first object image data) in the first object region from the first image data. The service server 1000 may transmit the first object image data to the cloud application server corresponding to the terminal device. The cloud application server may read and perform rendering on the first object image data, and after obtaining the rendered data, may transmit the rendered data to the terminal device. The terminal device may display and output the rendered data in the cloud application. In addition, it is to be understood that when the service server 1000 extracts the first object region of the object in the first image data, the service server 1000 may continuously perform image recognition processing on other image data. For example, the service server 1000 may obtain second image data with a latest receiving timestamp (that is, the last received image data) from the updated reception queue including the second image data. The service server 1000 then may perform image recognition processing on the second image data (which may be referred to as target image data) with the latest receiving timestamp. It is to be understood that when the service server 1000 performs image recognition processing on the target image data, the service server 1000 may continuously obtain image data (which may be referred to as third image data) including the object from the terminal device, and store the third image data to the updated reception queue, to obtain a new reception queue. When the service server 1000 extracts a region (which may be referred to as a second object region) of the object in the target image data, the service server 1000 may obtain image data (which may be referred to as second object image data) in the second object region from the target image

data. The service server may further transmit the second object image data to the cloud application server corresponding to the terminal device. In addition, the service server 1000 may obtain third image data (which may be referred to as new target image data) with a latest receiving timestamp from the current new reception queue. The service server 1000 then may perform image recognition processing on the new target image data.

[0040] It is to be understood that in the present disclosure, the service server 1000 may continuously receive image data during image recognition processing of other image data, so that recognition and reception are performed at the same time, instead of receiving image data after recognition. In this way, the delay for receiving image data is reduced. In addition, every time when completing recognition on a piece of image data (i.e., after extracting a region of the object), the service server may perform frame skipping processing, that is, the service server may obtain image data currently with a latest receiving timestamp, and perform image recognition processing on the image data, rather than obtaining next image data (image data with a closest receiving timestamp) to the currently processed image data and performing image recognition processing on the next image data). The frame skipping processing can reduce the delay for queuing image data. Since the image data with the latest receiving timestamp is acquired according to a current behavior of the user, by performing the image recognition and the display on the image data with the latest receiving timestamp, rendered data displayed in the cloud application can be synchronously matched with the current behavior of the user.

[0041] In this embodiment of the present disclosure, one of the plurality of terminal devices may be selected to perform data interaction with the service server 1000. The terminal device may include but is not limited to an intelligent terminal with a multimedia data processing function (for example, a video data playing function or a music data playing function), for example, a smartphone, a tablet computer, a notebook computer, a desktop computer, a smart television, a smart speaker, a desk computer, a smartwatch, or an intelligent vehicle terminal. For example, in this embodiment of the present disclosure, the terminal device 100a shown in FIG. 1 may be integrated with the foregoing cloud application. In this case, the terminal device 100a may perform data interaction with the service server 1000 by using the cloud application.

[0042] It may be understood that a method provided in the embodiments of the present disclosure may be performed by a computer device. The computer device includes but is not limited to a user terminal or the service server. The service server may be an independent physical server, a server cluster or distributed system including a plurality of physical servers, or a cloud application server providing a basic cloud computing service such as a cloud service, a cloud database, cloud

computing, a cloud function, cloud storage, a network service, cloud communication, a middleware service, domain name service, a security service, a content delivery network (CDN), or a big data and artificial intelligence platform.

[0043] The terminal device may be directly or indirectly connected to the service server through wired or wireless communication. This is not limited herein in the present disclosure.

[0044] It may be understood that the computer device (for example, the service server 1000, the terminal device 100a, or the terminal device 100b) may be a node in a distributed system. The distributed system may be a blockchain system. The blockchain system may be a distributed system including a plurality of nodes connected in a network communication form. The nodes may form a peer to peer (P2P) network. A P2P protocol is an application-layer protocol run over a transmission control protocol (TCP). In the distributed system, any form of computer device, for example, an electronic device like the service server or the terminal device, may join the P2P network to become a node in the blockchain system. For ease of understanding, the following describes the concept of blockchain. A blockchain is a novel application mode of a computer technology such as distributed data storage, P2P transmission, a consensus mechanism, or an encryption algorithm, and is mainly used for organizing data according to a time sequence, and encrypting the data into a ledger, thereby preventing the data from being tampered and falsified and implementing verification, storage, and update of the data. When the computer device is a blockchain node, an anti-tamper characteristic and an anti-falsification characteristic of the blockchain ensures that data (for example, the first image data, the first object region, or the second object image data) in the present disclosure is authentic and secure, so that a result obtained by performing related data processing based on the data is more reliable.

[0045] For ease of understanding, reference is made to FIG. 2a, which is a schematic diagram of a scenario according to an embodiment of the present disclosure. A terminal device 100a shown in FIG. 2a may be the terminal device 100a in the terminal device cluster 100 in the embodiment corresponding to FIG. 1. A service server 1000 shown in FIG. 2a may be the service server 1000 in the embodiment corresponding to FIG. 1. A cloud application server 10001 shown in FIG. 2a may be the cloud application server 10001 in the embodiment corresponding to FIG. 1.

[0046] As shown in FIG. 2a, the terminal device 100a may include a game application. When a user a (which may be referred to as an object a) starts the game application, the terminal device 100a may acquire a picture (which may be referred to as an original image frame 20a) including the object a by using a camera component 200a. The terminal device may perform encoding processing (for example, H264 encoding processing) on the original image frame to

obtain image data. The terminal device 100a may transmit the image data to the service server 1000. The service server 1000 may store the image data to a reception queue. Then, the service server 1000 may obtain the image data from the reception queue. The service server 1000 may perform decoding processing on the image data to obtain the original image frame 20a. It may be understood that for each piece of received image data (including the 1st piece of image data), the service server 1000 may store the image data to the reception queue. Alternatively, the 1st piece of image data received by the service server 1000 may be not stored, but is directly decoded. For example, if the image data corresponding to the original image frame 20a is the 1st piece of image data received by the service server 1000, the service server 1000 may store the image data to the reception queue according to a storage rule, and then obtain the image data from the reception queue. Alternatively, the service server 1000 may not store the image data, and may directly perform decoding processing on the image data to obtain the original image frame 20a.

[0047] The service server 1000 may perform image recognition processing on the original image frame 20a. The service server 1000 may determine, through image recognition processing, an object edge curve P1 corresponding to the object a in the original image frame 20a. It is to be understood that the user a may continuously conduct moving behaviors (for example, raising both hands, shaking the head, or squatting), and after acquiring the original image frame 20a, the terminal device 100a may continuously acquire original image frames including the object a by using the camera component 200a. Every time when the terminal device 100a successfully acquires an original image frame including the object a, the terminal device 100a may perform encoding processing on the original image frame to obtain image data, and transmit the image data to the service server 1000. The service server 1000 may receive the image data without being affected by currently performing image recognition processing on the original image frame 20a. The service server 1000 may continuously obtain different image data from the terminal device 100a during image recognition processing. The service server 1000 may temporarily store the image data to the reception queue.

[0048] After determining the object edge curve P1, the service server 1000 may extract an entire region (which may be referred to an object region P2) enclosed by the object edge curve P1 from the original image frame 20a, and obtain all image content in the object region from the original image frame 20a, thereby obtaining all image content (which may be referred to as object image data) in the object region. The service server 1000 may obtain a cloud application server (for example, the cloud application server 10001) corresponding to the terminal device 100a. The service server 1000 may transmit the object image data in the object region P2 to the cloud application server 10001. After obtaining the object image data, the cloud application server 10001

may perform rendering processing on the object image data, thereby obtaining rendered data P3. The cloud game server may transmit the rendered data P3 to the corresponding terminal device 100a.

[0049] FIG. 2b is a schematic diagram of a scenario according to an embodiment of the present disclosure. As shown in FIG. 2b, after receiving the rendered data P3, the terminal device 100a may display the rendered data P3 in the game application. As shown in FIG. 2b, a virtual environment (which may be understood as a game scenario) corresponding to a game includes a virtual background (a virtual house background), a dancing virtual object 2000a (which is dancing), and a dancing virtual object 2000b. The rendered data P3 may be displayed in the virtual environment.

[0050] It may be understood that when the service server 1000 extracts the object region P2 from the original image frame 20a, the service server 1000 may continue to process image data in the reception queue. In order to make the rendered data displayed by the terminal device 100a match a current behavior of the object a, the service server 1000 may perform frame skipping processing. That is, the service server 1000 may obtain image data with a latest receiving timestamp in the reception queue, and perform decoding and image recognition processing on the image data. The image data with the latest receiving timestamp may be understood as the last image data transmitted by the terminal device 100a at a current moment. The image data may correspond to a latest real-time behavior of the object a. Therefore, after an object region corresponding to the image data is extracted, rendered and outputted, the displayed rendered data matches the current behavior of the object.

[0051] It is to be understood that if the cloud application is the game application, the object a may be a game player, and rendered portrait data (for example, the rendered data P3) corresponding to the object a is displayed in the game application. That is, a player portrait is projected to the game scenario. In this way, the game player may be "placed" in the game scenario, so that immersion of the game player can be improved. In addition, in this embodiment of the present disclosure, image recognition and reception may be implemented at the same time by using the reception queue, so that an image data receiving delay may be reduced. Moreover, frame skipping processing may increase a matching ratio between the player portrait displayed in the game and the player while improving image data recognition efficiency and further reducing a delay.

[0052] FIG. 3 is a schematic flowchart of a data processing method according to an embodiment of the present disclosure. The method may be performed by a computer device. The

computer device may be a terminal device (for example, any terminal device in the terminal device cluster shown in FIG. 1, for example, the terminal device 100a) or a service server (for example, the service server 1000 shown in FIG. 1). Alternatively, the computer device may include a terminal device and a service server, so that the method is performed by the terminal device and the service server. For ease of understanding, an example in which the method is performed by the foregoing service server is used in this embodiment for description. The data processing method may include at least the following S101 to S103:

[0053] S101: Obtain first image data transmitted by a first client, and store the first image data to a reception queue, the first image data being image data including an object and obtained by the first client when running a cloud application.

[0054] In the present disclosure, the first client may be understood as the terminal device. An application program may be deployed in the first client. The application program may be the cloud application (for example, a game application), or the like. The cloud application is used as an example. When a user uses the first client, the user may start the cloud application in the first client. For example, the user may click the cloud application, and click a start control element to run the cloud application. The first client may be any client.

[0055] It is to be understood that after the user starts the cloud application in the first client, that is, when the first client runs the cloud application, the first client may acquire a picture including the user (which may be referred to as the object) by using a camera component. The picture including the user may be referred to as an original image frame. The first client may perform encoding processing on the original image frame, thereby obtaining an encoded image file. The encoded image file may be referred to as image data herein. The first client may transmit the image data to the service server (the service server may be a server with an image decoding function and an image recognition function, and may be configured to obtain the encoded file transmitted by the first client, and perform decoding and image recognition processing). Among various encoding modes, the H264 encoding is an encoding mode having a higher compression ratio, and a same image subjected to H264 encoding occupies a smaller bandwidth in transmission. Therefore, H264 is widely applied to mobile video applications. Therefore, in the present disclosure, H264 encoding is preferentially selected as an encoding mode for the original image frame, to reduce a transmission bandwidth between the first client and the service server. Certainly, any other mode capable of encoding the original image frame can be used by the first client as the encoding mode for the original image frame. For example, the encoding mode may be H262 encoding, H263 encoding, or H265 encoding. The encoding mode is not limited in the present disclosure.

[0056] In the present disclosure, after obtaining the image data transmitted by the first client, the service server may store the image data to the reception queue. Taking the first image data as an example, after obtaining the first image data, the service server may store the first image data to the reception queue, specifically by the following method: receiving the first image data transmitted by the first client (the first image data is data obtained by the first client by encoding the original image frame); then obtaining a receiving timestamp of receiving the first image data, and storing the first image data and the receiving timestamp associatively to the reception queue. That is, when storing each piece of image data, the service server may store a receiving moment of the image data together. For example, a receiving moment (which may be used as a receiving timestamp) of image data A by the service server is 19:09 on September 5, 2021. In this case, the service server may store the image data A and the receiving moment of 19:09 on September 5, 2021 associatively to the reception queue. It is to be understood that the reception queue may be empty when the reception queue does not store any image data.

[0057] S102: Perform image recognition processing on the first image data in the reception queue, and store, during the image recognition processing on the first image data, second image data to the reception queue to obtain an updated reception queue, the second image data being image data continuously obtained from the first client.

[0058] In the present disclosure, after the first image data is stored to the reception queue, the first image data may be obtained from the reception queue, and image recognition processing may be performed on the first image data. Since the first image data is actually the encoded image file, decoding processing may be first performed on the first image data to restore the original image frame, and then image recognition processing is performed on the original image frame. A specific method may be: decoding the first image data to obtain decoded image data in an initial image format; then, performing format conversion on the decoded image data to obtain the original image frame in a standard image format; and then, performing image recognition processing on the original image frame in the standard image format.

[0059] It is to be understood that the standard image format may be a specified image format for unified image recognition processing. For example, it is specified that an image for image recognition processing is required to be in a red green blue (RGB) color mode, and the RGB mode may be referred to as the standard image format. After decoding processing is performed on the first image data, the decoded image data in the initial image format may be obtained. If the initial image format is the standard image format, the decoded image data may be determined as the original image frame in the standard image format. If the initial image format is different from the

standard image format, format conversion may be performed to convert the decoded image data to the standard image format to obtain the original image frame in the standard image format. For example, when the initial image format is a YUV mode, the YUV mode may be converted into the RGB mode, thereby obtaining the original image frame in the RGB mode.

[0060] After obtaining, through decoding, the original image frame corresponding to the first image data, the service server may perform image recognition processing on the original image frame in the standard image format to determine a region (which may be referred to a first object region) of the object in the original image frame. A specific implementation of performing image recognition processing to determine the first object region is in subsequent descriptions in an embodiment corresponding to FIG. 5.

[0061] It may be understood that in the present disclosure, after the first client transmits the first image data to the service server, the first client may continuously acquire a picture (a new original image frame) including the object, and the first client may perform encoding processing on each original image frame to obtain an encoded image file (which may be referred to as the second image data). The first client may continuously transmit each piece of second image data to the service server. The service server does not stop receiving image data during image recognition processing of the first image data. The service server may continuously obtain the second image data from the first client. The service server may temporarily store the second image data to the reception queue, thereby obtaining the updated reception queue.

[0062] S103: Transmit, in a case that the first object region of the object in the first image data is extracted through image recognition processing, the first object region to a target cloud application server, and at the same time perform image recognition processing on second image data having a latest receiving timestamp in the updated reception queue, the target cloud application server being configured to render the first object region to obtain rendered data, and transmit the rendered data to the first client.

[0063] In the present disclosure, the first object region of the object in the original image frame may be determined through image recognition processing. After the first object region is determined, image content (which may be referred to as first object image data) in the first object region may be obtained from the original image frame. The service server may extract the first object region and the first object image data in the first object region. When extracting the first object region and the first object image data, the service server may obtain the target cloud application server corresponding to the first client. The service server may transmit the first object image data to the target cloud application server. The target cloud application server may be a cloud

application server corresponding to the first client. When the first client runs the cloud application, the cloud application server provides a computing service for the first client. The computing service is, for example, a central processing unit (CPU) computing service or a graphics processing unit (GPU) computing service. The target cloud application server may perform rendering processing on the first object image data, thereby obtaining the rendered data corresponding to the first object image data. The target cloud application server may transmit the rendered data to the first client. The first client may display the rendered data in the cloud application.

[0064] In addition, it is to be understood that after the service server extracts the first object region, the service server may continue to perform decoding and image recognition processing on other image data. For example, the service server may obtain the second image data with the latest receiving timestamp in the reception queue. The service server may perform decoding and image recognition processing on the second image data.

[0065] A process in which the service server obtains the second image data with the latest receiving timestamp in the updated reception queue and performs decoding and image recognition processing on the second image data may be referred to as frame skipping processing. For ease of understanding, reference is also made to FIG. 4 together, which is a schematic diagram of frame skipping processing according to an embodiment of the present disclosure. As shown in FIG. 4, a reception queue 40a may include image data 1, image data 2, image data 3, image data 4, image data 5, image data 6, image data 7, image data 8, and image data 9. The image data 1 to the image data 9 are sorted according to a sequence from early to late receiving timestamps, and are represented by reference signs 1, 2, 3, 4, 5, 6, 7, 8, and 9 respectively in FIG. 4. That is, a receiving timestamp of the image data 1 is an earliest receiving timestamp, and a receiving timestamp of the image data 9 is a latest timestamp. The image data 1, the image data 2, the image data 3, the image data 4, and the image data 5 are processed image data. The image data 6 is image data currently processed by the service server. The image data 7, the image data 8, and the image data 9 are image data received by the service server during processing of the image data 6. The image data 7, the image data 8, and the image data 9 are waiting in the queue for processing.

[0066] As shown in FIG. 4, when the service server extracts an object region of the object in the image data 6, the service server may obtain the image data 9 at a tail of the reception queue 40a (that is, obtain image data with the latest receiving timestamp). The service server may skip the image data 7 and the image data 8, and perform decoding and image recognition processing on the image data 9. This is a frame skipping processing. For ease of understanding, the following uses an example for detailed description. It is assumed that time required by the service server to perform

decoding and image recognition processing is 30 ms, and a time interval between reception of two pieces of image data by the service server is 10 ms (that is, one piece of image data is received at a 0th ms, next image data is received at a 10th ms, and the next to the next image data is received at a 20th ms). In this case, when the service server processes the image data 6, the service server may continuously receive the image data 7 (which may be stored to the tail of the reception queue, that is, after the image data 6), the image data 8 (which may be stored to the tail of the reception queue, that is, after the image data 7), and the image data 9 (which may be stored to the tail of the reception queue, that is, after the image data 8). After the image data 6 is completely processed, the reception queue is as shown by the reception queue 40a. In this case, the service server may directly obtain latest image data (that is, the image data 9) from the tail of the reception queue, skipping the image data 7 and the image data 8 (the image data 7 and the image data 8 are not processed, but are actually skipped and will not be processed, so that the image data 7 and the image data 8 may be determined as processed image data).

[0067] Similarly, as shown in FIG. 4, when processing the image data 9, the service server may also continuously receive image data 10 and image data 11 (represented by reference signs 10 and 11 in FIG. 4), to obtain a reception queue 40b. When an object region of the object in the image data 9 is extracted, image data (that is, the image data 11 with a latest receiving timestamp) at a tail of the reception queue 40b may be obtained. The service server may skip the image data 10, and perform decoding and image recognition processing on the image data 11. Similarly, when processing the image data 11, the service server may continuously receive other image data to obtain a reception queue 40c. When the image data 11 is completely processed, image data at a tail of the reception queue 40c may be obtained. The operations are repeated, and will not be elaborated herein.

[0068] It may be understood that every time when a piece of image data is processed, image data (that is, processed image data) received before the piece of image data may be cleared, thereby saving storage space for the reception queue. For example, if processed image data in the reception queue 40a includes the image data 1 to the image data 5, the image data 1 to the image data 5 may be deleted, and then the reception queue 40 includes only the image data 6 to the image data 9. Alternatively, every time when a piece of image data is completely processed, image data with a latest receiving timestamp in the reception queue may be obtained, and after the image data is obtained, image data (which may be referred to as historical image data) received before the image data (that is, a receiving timestamp is earlier than that of the image data) may be deleted. For example, the image data 9 in the reception queue 40a is obtained as to-be-processed image data. In

this case, historical image data (including the image data 1 to the image data 8) before the image data 9 has been deleted.

[0069] That is, when the first object region in the first image data is extracted, processed image data (including the first image data) may be cleared first, and then the second image data with the latest receiving timestamp is obtained from remaining unprocessed image data. Similarly, when the first object region in the first image data is obtained, the second image data with the latest receiving timestamp may be obtained first, and then historical image data whose receiving timestamp is earlier than that of the second image data is deleted (that is, the second image data with the latest receiving timestamp in the updated reception queue is obtained first, then image recognition processing is performed on the second image data, and the historical image data in the updated reception queue is deleted at the same time, the historical image data being the image data whose receiving timestamp is earlier than that of the second image data in the updated reception queue). It is to be understood that each of the queue clearing methods is for improving the storage space of the queue. The queue emptying method will not be specifically limited in the present disclosure.

[0070] It is to be noted that after the service server receives an encoded bitstream (that is, the image data) transmitted by the first client, if time for decoding and image recognition processing (which is referred to as image processing) is longer than the time interval between reception of two frames of image data (for example, the image processing time is 30 ms, and the time interval between reception of the two frames of image data is 10 ms), and if there is no reception queue, the first client have to wait for the service server to perform image processing on the current image data. As a result, the delay for transmitting image data is greatly prolonged, which seriously affect image data transmission efficiency. In the present disclosure, the image data may be stored by using the reception queue, so that the service server may continuously receive image data transmitted by the first client during image processing. If the service server sequentially performs image processing on the image data, there may be a serious mismatch between the image data recognized by the service server and a latest status of the object, and the image data recognized by the service server lags seriously. Therefore, the frame skipping processing can ensure that the service server performs image processing on latest image data each time, which can reduce the delay for image recognition. In addition, due to a high computing capability of the service server, image recognition efficiency can be improved.

[0071] In this embodiment of the present disclosure, when obtaining the first image data including the object, a client (for example, the first client) may transmit the first image data to a

related computer device (for example, the service server), the service server performs image recognition processing, and the first client is not required to perform image recognition locally. In this way, image recognition processing may be performed on the first image data by using the service server with a high computing capability, so that image recognition efficiency and image definition can be improved. In addition, in the present disclosure, the service server may store the received first image data to the reception queue, and during the image recognition processing on the first image data, continuously obtain the second image data from the first client and store the second image data to the reception queue to obtain the updated reception queue. That is, in the present disclosure, the service server does not stop receiving the second image data during the image recognition processing on the first image data, but implements image processing and image reception at the same time by using the reception queue. In this way, an image transmission delay can be reduced. Further, when the service server extracts, through image recognition processing, the first object region of the object in the first image data, the service server may transmit the first object image data in the first object region to the target cloud application server, and the target cloud application server performs rendering, and transmits the rendered data to the first client, to be displayed in the cloud application. In addition, after extracting the first object region, the service server may obtain the second image data with the latest receiving timestamp in the reception queue, and continue to process the second image data. It can be seen that in the present disclosure, image data is not recognized one by one according to a time sequence of receiving timestamps. Instead, image data with a latest receiving timestamp is obtained from the reception queue for processing after image recognition processing is performed on a piece of image data, so that image data recognition efficiency may be improved. In addition, since the image data with the latest receiving timestamp is acquired according to a current behavior of the object, by performing image recognition and displaying on the image data with the latest receiving timestamp, the displayed data matches the current behavior of the object. In summary, according to the present disclosure, the image recognition efficiency is improved, the image transmission delay is reduced, and it is ensured that a virtual behavior of a virtual object displayed in the cloud application matches a current behavior status of the object.

[0072] FIG. 5 is a schematic flowchart of a data processing method according to an embodiment of the present disclosure. The process may correspond to the process of performing image recognition processing on the original image frame to determine the first object region in the embodiment corresponding to FIG. 3. As shown in FIG. 5, the process may include at least the following S501 to S503:

[0073] S501: Recognize object edge key points of the object in the original image frame.

[0074] The object edge key points herein may be object contour key points of the object. The original image frame includes a key part of the object. The object contour key points herein may be contour key points of the key part. If the key part is the head, the object edge key points may be head contour key points. If the key part is the neck, the object edge key points may be neck contour key points.

[0075] Recognition of the object edge key point of the object may include recognition based on an artificial intelligence algorithm, recognition based on a dedicated GPU, and another manner. This is not limited in the present disclosure.

[0076] S502: Connect the object edge key points to obtain an object edge curve of the object.

[0077] After the object edge key points are determined, the object edge key points may be connected (for example, every two adjacent points are connected) to obtain the object edge curve (which may be understood as an object contour) corresponding to the object. For example, in the scenario embodiment corresponding to FIG. 2a, the curve P1 shown in FIG. 2a may be regarded as an object contour of the object a.

[0078] S503: Determine a region enclosed by the object edge curve in the original image frame as an initial object region of the object in the original image frame, and determine the first object region according to the initial object region.

[0079] After an object edge curve of the object a is determined, a region enclosed by the object edge curve in the original image frame may be determined. The region may be used as the first object region of the object in the original image frame. For example, in the scenario embodiment corresponding to FIG. 2a, a region enclosed by the object edge curve P2 in the original image frame 20a is the region P2 (the region P2 is a region in which the object a is). The region P2 may be determined as a region (which may be referred to as the first object region herein) of the object in the original image frame.

[0080] In a possible embodiment, the region enclosed by the object edge curve may be referred to as the initial object region. After the initial object region is determined, the initial object region may not be determined as a final first object region. Instead, the first object region is determined according to the initial object region. A specific method may be: obtaining an object key part presented by data in the initial object region of the object; then, obtaining object recognition configuration information for the object and determining a configured object recognition part indicated by the object recognition configuration information, and determining whether the

configured object recognition part matches the object key part; and performing, in a case that the configured object recognition part matches the object key part, the operation of determining the first object region according to the initial object region; and determining, in a case that the configured object recognition part does not match the object key part, that the first object region is not extracted through the image recognition processing.

[0081] It is to be understood that after the initial object region of the object is determined, the object key part presented by data in the initial object region of the object may be obtained (the object key part may be a body part of the object, for example, the head, the neck, an arm part, the abdomen, the leg, or the foot). Then, the configured object recognition part (that is, a recognition rule specifying a part of the object required to be included in the original image frame acquired by the terminal device) of the object required to be included in the original image frame may be obtained. For example, the configured object recognition part is the leg. If the original image frame acquired by the terminal device is required to include the leg of the user, but the object key part presented by data in the initial object region extracted by performing decoding and image recognition processing on the received image data is the head and the neck, it is determined that the object key part (the head and the neck) does not match the configured object recognition part (the leg), and the object key part presented in the original image frame does not meet a requirement. In this case, it is directly determined that the first object region cannot be extracted through image recognition processing (that is, the part does not meet the requirement, and the extraction fails). If the original image frame acquired by the terminal device is required to include the leg of the user, but the object key part presented by data in the initial object region extracted by performing decoding and image recognition processing on the received image data is the leg, it is determined that the object key part presented in the original image frame meets the requirement. In this case, the initial object region is determined as the first object region.

[0082] In a possible embodiment, from the above descriptions, if it is determined that the configured object recognition part does not match the object key part, after determining that the first object region is not extracted through image recognition processing, the service server may obtain next image data to the current image data in the updated reception queue, and then perform image processing on the next image data. For example, the current image data is the first image data, the next image data (that is, image data with an earliest receiving timestamp in image data whose receiving timestamp is later than that of the first image data) to the first image data in the reception queue may be obtained, and then the service server may perform image processing on the next image data. A specific method may be: determining second image data with an earliest receiving timestamp in the updated reception queue as to-be-recognized image data; and then, performing

image recognition processing on the to-be-recognized image data, and transmitting, in a case that a to-be-processed object region of the object in the to-be-recognized image data is extracted through image recognition processing, the to-be-processed object region to the target cloud application server.

[0083] In fact, if the service server performs image processing on the current image data (for example, the first image data) fast and efficiently enough, after it is determined that the first object region is not extracted through image recognition processing, the next image data (the image data with the earliest receiving timestamp) to the current image data may be obtained, and image processing is performed on the next image data. A purpose of doing this is that when the user performs an action, the first client obtains and encodes an image frame, and transmits the encoded image frame to the service server, the service server fast recognizes that an object key part in the image frame does not meet the requirement, and cannot extract an object region and image data in the object region. As a result, the cloud application server cannot receive extracted object image data and perform alignment for rendering and displaying. In this case, the service server may perform image processing on next image data to extract an object region in the next image data, and then transmit object image data in the object region to the cloud application server for rendering and output. In this way, jitter of a portrait of the user displayed in the cloud application can be reduced, and coherence of the portrait can be improved. Certainly, after it is determined that the first object region is not extracted through image recognition processing, the second image data with the latest receiving timestamp may alternatively be obtained from the current updated reception queue (rather than obtaining the image data with the earliest receiving timestamp), and image processing is performed on the second image data. Either of the obtaining methods is a possible processing method after it is determined that the first object region is not extracted through image recognition processing. A specific processing method may be selected according to an actual situation, and will not be limited in the present disclosure.

[0084] In a possible embodiment, from the above descriptions, when it is determined that the configured object recognition part matches the object key part, the initial object region may be directly determined as the first object region. In addition, when it is determined that the configured object recognition part matches the object key part, a specific method for determining the first object region may alternatively be: obtaining an object key part presented by data in the initial object region of the object; determining the initial object region as the first object region in a case that the object key part meets a part completeness requirement; and obtaining N (N is a positive integer) sample image frames from a sample database in a case that the object key part does not meet the part completeness requirement, obtaining a to-be-processed sample image frame

corresponding to the object from the N sample image frames, and determining the first object region according to the to-be-processed sample image frame and the initial object region.

[0085] A specific method for determining the first object region according to the to-be-processed sample image frame and the initial object region may be: obtaining overall part information in the to-be-processed sample image frame; then, determining a to-be-fused part region in the overall part information according to the object key part; and fusing the to-be-fused part region and the initial object region to obtain the first object region.

[0086] It is to be understood that in the present disclosure, sample data (complete portrait sample data from the head to the foot) of the complete overall portrait of the user may be pre-collected. One user may correspond to one sample image frame. One sample image frame may present complete overall portrait data of one user. In this case, when the initial object region is extracted, and it is determined that the configured object recognition part matches the object key part, whether the object key part meets a part completeness requirement may be determined. If the object key part meets the part completeness requirement, the initial object region may be directly determined as the first object region. If the object key part does not meet the part completeness requirement, the to-be-processed sample image frame corresponding to the object may be obtained from the sample database, then the overall part information of the object is obtained from the to-be-processed sample image frame, and the initial object region is completed according to the overall part information, to obtain a complete first object region including a complete part.

[0087] For ease of understanding, reference is made to FIG. 6 together, which is a schematic diagram of a part fusion scenario according to an embodiment of the present disclosure. As shown in FIG. 6, it is assumed that the initial object region is an initial object region 600a, and an object key part presented in the initial object region 600a includes the head, the neck, the arm part, the chest, and the abdomen (that is, the upper body of the user). If the configured object recognition part is also the upper body of the user, that is, the first client is required to acquire the upper body of the user, the initial object region meets the requirement. Further, whether the object key part meets a part completeness requirement may be determined. In this case, if the part completeness requirement is specified as completely including the overall portrait of the user (that is, both the upper body and the lower body, that is, from the head to the foot, are required to be included), the object key part presented in the initial object region 600a does not meet the part completeness requirement. In this case, the service server 1000 may obtain the to-be-processed sample image frame (for example, the to-be-processed sample image frame 600b) corresponding to the object in the sample database. As shown in FIG. 6, the overall part information presented in the

to-be-processed sample image frame includes complete information of the object from the head to the foot. Since the initial object region 600a includes the upper body, the lower body in the to-be-processed sample image frame is determined as the to-be-fused part region (that is, a region 600c), and the to-be-fused part region 600c is extracted. Further, the to-be-fused part region 600c and the initial object region 600a are fused (for example, spliced), thereby obtaining a first object region 600d including the upper body and the lower body. It is to be understood that the overall part information (for example, from the head to the foot) of the user is acquired in advance, so that every time when obtaining a picture of the user, the first client does not need to strictly require the user to stand at a fixed position at which complete parts can be acquired. The user may move flexibly, and the first client only needs to obtain partial part information. After obtaining the partial part information from the first client, the service server may supplement and stitch the partial part information according to the overall part information acquired in advance, to obtain the complete parts. In this way, experience and immersion of the user can be improved.

[0088] It is to be noted that in the foregoing process, if the part completeness requirement is specified to be completely including the upper body of the user, the object key part presented by the initial object region 600a meets a part completeness requirement. In this case, the initial object region 600a may be directly determined as the first object region.

[0089] A specific method for obtaining the to-be-processed sample image frame corresponding to the object in the sample database may be implemented by face matching. Alternatively, when a sample image frame of the user is acquired, the sample image frame is identified by using a user identifier (for example, a user name or a user serial number) of the user, such that each sample image frame has a user identifier (which may be referred to as a sample identifier). When transmitting image data to the service server, the first client may also transmit the user identifier of the user in the image data. Then, the service server may obtain a corresponding to-be-processed sample image frame according to the user identifier and the sample identifiers of the sample image frames. Certainly, a specific implementation of obtaining the to-be-processed sample image frame corresponding to the object in the sample database is not limited to the above-described methods. The specific implementation is not limited in the present disclosure.

[0090] In this embodiment of the present disclosure, when obtaining the first image data including the object, a client (for example, the first client) may transmit the first image data to a related computer device (for example, the service server), the service server performs image recognition processing, instead of requiring the client to perform the image recognition locally. In this way, image recognition processing may be performed on the first image data by using the

service server with a high computing capability, so that image recognition efficiency and a definition can be improved. In addition, in the present disclosure, the service server may store the received first image data to the reception queue, continuously obtain the second image data from the first client during the image recognition processing on the first image data, and store the second image data to the reception queue to obtain the updated reception queue. That is, in the present disclosure, the service server does not stop receiving the second image data during the image recognition processing on the first image data, but implements image processing and image reception at the same time by using the reception queue. In this way, the image transmission delay can be reduced. Further, when the service server extracts, through image recognition processing, the first object region of the object in the first image data, the service server may transmit the first object image data in the first object region to the target cloud application server, and the target cloud application server performs rendering, and transmits the rendered data to the first client. In this way, displaying may be performed in the cloud application. In addition, after extracting the first object region, the service server may obtain the second image data with the latest receiving timestamp in the reception queue, and continue to process the second image data. It can be seen that in the present disclosure, image data is not recognized one by one according to a time sequence of receiving timestamps. Instead, image data with a latest receiving timestamp is obtained from the reception queue for processing after image recognition processing is performed on a piece of image data, so that image data recognition efficiency can be improved. In addition, since the image data with the latest receiving timestamp is acquired according to a current behavior of the object, by performing image recognition and displaying on the image data with the latest receiving timestamp, the displayed data matches the current behavior of the object. In summary, according to the present disclosure, the image recognition efficiency is improved, the image transmission delay is reduced, and it is ensured that a virtual behavior of a virtual object displayed in the cloud application matches a current behavior status of the object.

[0091] In a possible embodiment, when each client runs the cloud application for the first time, a cloud application server corresponding to each client may transmit a registration request to the service server. The registration request is used for requesting for device registration to the service server. After registration, the service server may add a device identifier corresponding to the cloud application server to a stored device identifier set, thereby proving that the cloud application server is a registered cloud application server. When the cloud application server is a registered cloud application server, it indicates that the cloud application server is a valid cloud application server. In this case, the service server is allowed to perform data interaction with the valid cloud application server. Then, when the first client transmits image data (for example, the first image

data) to the service server, a device identifier (which may be referred to as a to-be-confirmed device identifier) of the cloud application server (which may be referred to as a bound cloud application server) in a binding relationship may be transmitted together for the service server to confirm, according to the device identifier, whether the cloud application server is registered (valid). When it is determined that the bound cloud application server is registered, the bound cloud application server is determined as the target cloud application server, and then the first object image data is transmitted to the target cloud application server. That is, before the service server transmits the first object image data to the target cloud application server and after determining the first object region, whether the cloud application server corresponding to the first client is registered may be determined first, and when it is determined that the cloud application server is registered, the first object image data is transmitted to the corresponding target cloud application server.

[0092] For ease of understanding, reference is made to FIG. 7, which is a schematic flowchart of transmitting the first object image data to the target cloud application server according to an embodiment of the present disclosure. The process is described by using an example in which the first image data carries the to-be-confirmed device identifier (the to-be-confirmed device identifier is the device identifier of the bound cloud application server, and the bound cloud application server is in the binding relationship with the first client). As shown in FIG. 7, the process may include at least the following S701 to S704:

[0093] S701: Obtain the stored device identifier set, the stored device identifier set including M stored device identifiers, each stored device identifier corresponding to a registered cloud application server, and M being a positive integer.

[0094] When a client runs the cloud application (generally runs the cloud application for the first time), the cloud application server corresponding to the client may transmit the registration request to the service server. The registration request is used for requesting for device registration to the service server. After registration, the service server may add the device identifier corresponding to the cloud application server to the stored device identifier set, thereby proving that the cloud application server is a registered cloud application server. An example in which the client is a second client is used. A specific method may be as follows: when the user starts the cloud application by using the client, the second client may generate an application starting notification in response to this application starting operation. The second client may transmit the application starting notification to a corresponding cloud application server (which may be referred to a to-be-registered cloud application server). Then, the to-be-registered cloud application server may transmit a registration request to the service server based on the application starting notification.

The service server may receive the registration request transmitted by the to-be-registered cloud application server. Then, the service server may detect device index information of the to-be-registered cloud application server according to the registration request. In a case that the device index information satisfies a processing quality condition, a to-be-stored device identifier of the to-be-registered cloud application server is obtained, the to-be-stored device identifier is stored to the stored device identifier set, so that the to-be-registered cloud application server becomes a registered cloud application server, and the to-be-stored device identifier becomes a stored device identifier.

[0095] The device index information may include a network quality parameter, a device version, a functional module quality index, a storage space index, and the like. Detecting the device index information herein may be detecting whether an index meets a requirement. For example, it is detected whether the network quality parameter meets a requirement, and if the network quality parameter meets the requirement, it may be considered that the device index information of the to-be-registered cloud application server satisfies the processing quality condition. Alternatively, detecting the device index information may be detecting whether two or more indexes meet requirements, and only when the indexes are all meets the requirements, it is confirmed that the device index information of the to-be-registered cloud application server satisfies the processing quality condition.

[0096] The following uses an example in which the device index information includes the network quality parameter and the device version to describe a specific method for detecting the device index information of the to-be-registered cloud application server. The specific method may be: obtaining the network quality parameter and the device version of the to-be-registered cloud application server according to the registration request; and determining, in a case that the network quality parameter reaches a parameter threshold and the device version matches a quality standard version (which may be understood as a qualified version), that the device index information satisfies the processing quality condition; or determining, in a case that the network quality parameter does not reach a parameter threshold or the device version does not match a quality standard version, that the device index information does not satisfy the processing quality condition.

[0097] It can be seen from the above that the stored device identifier set stores stored device identifiers corresponding to different registered cloud application servers. Then, after the to-be-confirmed device identifier transmitted by the first client is obtained, the stored device identifier set may be obtained, and then the to-be-confirmed device identifier is matched with the stored device identifier set.

[0098] S702: Determine whether the M stored device identifiers include a stored device identifier matching the to-be-confirmed device identifier.

[0099] If the M stored device identifiers include the stored device identifier matching the to-be-confirmed device identifier, subsequent S703 is performed; or if the M stored device identifiers do not include the stored device identifier matching the to-be-confirmed device identifier, subsequent S704 is performed.

[00100] S703: Determine, in a case that the M stored device identifiers include the stored device identifier matching the to-be-confirmed device identifier, that the bound cloud application server indicated by the to-be-confirmed device identifier is a registered cloud application server, determine the bound cloud application server indicated by the to-be-confirmed device identifier as the target cloud application server, and transmit the first object image data to the target cloud application server.

[00101] The to-be-confirmed device identifier is matched with the M stored device identifiers. If the M stored device identifiers include the stored device identifier matching the to-be-confirmed device identifier, it is determined that the bound cloud application server indicated by the to-be-confirmed device identifier is a registered cloud application server. In this case, the bound cloud application server indicated by the to-be-confirmed device identifier is determined as the target cloud application server, and the first object image data is transmitted to the target cloud application server.

[00102] S704: Generate device exception prompt information in a case that the M stored device identifiers do not include the stored device identifier, and transmit the device exception prompt information to the first client.

[00103] The to-be-confirmed device identifier is matched with the M stored device identifiers. If the stored device identifier set does not include the stored device identifier matching the to-be-confirmed device identifier, it may be determined that the bound cloud application server indicated by the to-be-confirmed device identifier is an unregistered cloud application server. That is, the bound cloud application server is not registered, and the service server does not transmit the first object image data to the bound cloud application server. In this case, the service server may generate the device exception prompt information (which may be server unregistration prompt information). The service server may return the device exception prompt information to the first client. The first client may transmit a registration notification to the corresponding bound cloud application server based on the device exception prompt information. The bound cloud application server may apply to the service server for registration based on the registration notification.

[00104] It is to be understood that in a possible case, the first client is not the only client transmitting image data to the service server, and different clients may transmit image data to the service server. Since different clients correspond to different cloud application servers, there are a plurality of cloud application servers registered to the service server (there are a plurality of cloud application servers connected to the service server). In the present disclosure, the stored device identifier set is pre-stored, and when transmitting image data, the client transmits a device identifier of a corresponding cloud game server together. In this way, a correspondence between a client and a cloud application server may be determined. In addition, whether the cloud application server is registered may be determined. Therefore, a user picture acquired by the client may be transmitted to the right registered cloud application server. This can improve correctness of the user image displayed in the cloud application.

[00105] It may be understood that for each cloud application server, for example, the target cloud application server, the reception of portrait data (that is, object image data, for example, the first object image data) transmitted by the service server may normally include the following three steps:

[00106] 1: Allocate a reception buffer, and write the portrait data (for example, the first object image data) to the reception buffer.

[00107] 2: After reception is completed, copy the portrait data for processing and rendering.

[00108] 3: After rendering is completed, return to step 1.

[00109] That is, the target cloud application server first writes the portrait data to the buffer, then reads and renders the portrait data, and after completing rendering, continues to receive portrait data and write the portrait data to the buffer. The volume of the received portrait data with a high resolution is large. As a result, it takes much time for the target cloud application server to allocate a buffer and copy the data, which seriously affects subsequent portrait data receiving and causes a long delay. Thus, it can be seen from the above that, the frame skipping processing of the service server can reduce the delay for image receiving by the service server side, but there is still a delay on a cloud application server side. The present disclosure provides a data processing method, in order to further reduce the delay. The method is to allocate two buffers on the cloud application server side. For ease of understanding, reference is made to FIG. 8, which is a schematic flowchart of a data processing method according to an embodiment of the present disclosure. The process is performed by a computer device. The computer device may be a target cloud application server, for example, a cloud game server. The process may correspond to a data processing process after the

target cloud application server receives object image data. As shown in FIG. 8, the process may include at least the following S801 to S804:

[00110] S801: Receive first object image data transmitted by a service server, and store the first object image data to a first buffer whose working mode is a storage mode in a buffer set, the first object image data being image data in a first object region, the first object region being a region of an object in the first image data, the first object region being obtained by the service server performing image recognition processing on first image data, the first image data being transmitted by a first client to the service server, and the first image data being image data including the object and obtained by the first client when running a cloud application.

[00111] For the determination and extraction processes of the first object region and the first object image data, one may refer to the descriptions in S102 in the embodiment corresponding to FIG. 3, and will not be elaborated herein. After extracting the first object region and obtaining the first object image data, the service server may transmit the first object image data to the target cloud application server. The target cloud application server may store the first object image data to the first buffer whose working mode is the storage mode in the buffer set.

[00112] The target cloud application server may pre-allocate two reception buffers of the same size, and set a working mode of one of the two buffers to the storage mode. That is, the buffer is actually a storage buffer. The target cloud application server may store received data to the storage buffer. In addition, a working mode of the other buffer may be set to a read mode. That is, the buffer is actually a read buffer. When required to read and render data, the target cloud application server may read the data from the read buffer.

[00113] Taking two buffers as an example, a specific method for allocating two buffers to generate the buffer set may include: pre-allocating the first buffer and a second buffer; then, setting an initial indicator of the first buffer to be a storage indicator, and setting an initial indicator of the second buffer to be a read indicator, where the working mode of the first buffer with the storage indicator is the storage mode, and the working mode of the second buffer with the read indicator is the read mode; and then, generating the buffer set according to the first buffer whose working mode is the storage mode and the second buffer whose working mode is the read mode. In this case, when the first object image data is received, since the working mode of the first buffer is the storage mode, the first object image data is stored to the first buffer.

[00114] S802: Adjust the working mode of the first buffer to the read mode and adjust the working mode of the second buffer to the storage mode in a case that the second buffer whose working mode is the read mode in the buffer set does not include unprocessed object image data,

and read the first object image data from the first buffer whose working mode is the read mode, and perform rendering processing on the first object image data.

[00115] Through the foregoing descriptions, an initial working mode of the first buffer is the storage mode, and an initial working mode of the second buffer is the read mode. When receiving and storing the first image data, the target cloud application server may at the same time read and render data (which may be referred to as stored image data) that has been stored in the second buffer. If the second buffer currently stores no image data, that is, the second buffer does not include unprocessed object image data, after the first object image data is stored to the first buffer, the storage indicator of the first buffer is switched to the read indicator, and the storage indicator of the second buffer is switched to the storage indicator. In this way, the working modes of the first buffer and the second buffer are interchanged, a current working mode of the first buffer is changed to the read mode, and a current working mode of the second buffer is changed to the storage mode. In this case, the first object image data is read from the first buffer and rendered, and meanwhile, second object image data is continuously received and stored to the second buffer.

[00116] It is to be understood that, by setting the two buffers, the target cloud application server can implement read and reception at the same time, and can receive other data without waiting until completing rendering. Therefore, a receiving delay can be reduced greatly.

[00117] It is to be understood that through the foregoing descriptions, the initial working mode of the first buffer is the storage mode, and the initial working mode of the second buffer is the read mode. While receiving and storing the first image data, the target cloud application server may also read and render the data (which may be referred to as the stored image data) that has been stored in the second buffer. If all the image data stored in the second buffer has been read and rendered, the processed image data may be cleared from the second buffer. In this case, it may be determined that the second buffer does not include unprocessed image data, and the indicator switching is performed to change the working mode of the first buffer to the read mode and change the working mode of the second buffer to the storage mode. Then, the first object image data is read from the first buffer in the read mode, and is rendered.

[00118] S803: Receive, during rendering of the first object image data, the second object image data transmitted by the service server, and store the second object image data to the second buffer whose working mode is the storage mode, the second object image data being image data in a second object region, the second object region being obtained by the service server by performing image recognition processing on second image data after extracting the first object region, the second object region being a region of the object in the second image data, the second image data

being image data that is obtained by the service server from an updated reception queue in a case that the first object region is extracted and that has a latest receiving timestamp, and second image data in the updated reception queue being continuously obtained by the service server from the first client during image recognition processing of the first image data.

[00119] The second object region may be a region obtained by the service server by performing image recognition processing on the second image data. The second object image data may be image data in the second object region in the second image data. A specific extraction manner for the second object region may be the same as that for the first object region, and will not be elaborated herein.

[00120] While rendering the first object image data, the second object image data transmitted by the service server is received. That is, the target cloud application server can receive data at the same time when reading data, and store the received data to the second buffer currently in the storage mode. In this way, the delay can be reduced.

[00121] S804: when rendered data corresponding to the first object image data is obtained, adjust the working mode of the first buffer to the storage mode and adjust the working mode of the second buffer to the read mode, read the second object image data from the second buffer whose working mode is the read mode, and perform rendering processing on the second object image data.

[00122] When stored data in the first buffer is completely read and rendered (that is, the rendered data corresponding to the first object image data is obtained if the first buffer includes only the first object image data), the working mode of the first buffer is adjusted to the storage mode, and the working mode of the second buffer is adjusted to the read mode. In this case, the second object image data is read from the second buffer whose working mode is the read mode, and rendering processing is performed on the second object image data. At the same time, other object image data may also be received and stored to the first buffer in the storage mode. A specific implementation of adjusting the working mode of the first buffer to the storage mode and adjusting the working mode of the second buffer to the read mode may be switching the indicators. A specific method may be: when first rendered data corresponding to the first object region is obtained, obtaining the read indicator corresponding to the first buffer and representing the read mode and obtaining the storage indicator corresponding to the second buffer and representing the storage mode; switching the read indicator corresponding to the first buffer to the storage indicator, the working mode of the first buffer with the storage indicator being the storage mode; and switching the storage indicator corresponding to the second buffer to the read indicator, the working mode of the second buffer with the read indicator being the read mode.

[00123] For ease of understanding, reference is made to FIG. 9, which is a schematic diagram of changing the statuses of the two buffers according to an embodiment of the present disclosure. As shown in FIG. 9, an example in which the first buffer is a buffer 900a and the second buffer is a buffer 900b is used. In this case, a working mode of the buffer 900a is the read mode. Object image data stored in the buffer 900a may include object image data 1 to object image data 10, which are sequentially represented by reference signs 1, 2, 3, 4, 5, 6, 7, 8, 9, and 10 in FIG. 9. The object image data 1 to the object image data 7 are already read data. The object image data 8 to the object image data 10 are to-be-read data. In addition, a working mode of the buffer 900b is the storage mode. During a process of reading data from the buffer 900a, the target cloud application server may continuously receive object image data and store the object image data to the buffer 900b. In this case, received data in the buffer 900b includes object image data 11 to object image data 14 (there are six other spaces for receiving object image data in the buffer 900b).

[00124] As shown in FIG. 9, when the data in the buffer 900a is completely read (that is, the object image data 7 to the object image data 9 are completely read), the buffer 900a may be cleared. In this case, the received data in the buffer 900b includes the object image data 11 to object image data 20 (sequentially represented by reference signs 11, 12, 13, 14, 15, 16, 17, 18, 19, and 20 in FIG. 9). Further, the working mode of the buffer 900a is switched to the storage mode, and the working mode of the buffer 900b is switched to the read mode. Then, the target cloud application server may read data (for example, sequentially read from the object image data 11) from the buffer 900b. At the same time, the target cloud application server may receive object image data and store the object image data to the buffer 900a. For example, new object image data 1 to new object image data 3 are received and stored to the buffer 900a.

[00125] It is to be noted that the buffer 900a and the buffer 900b are only examples for illustration.

[00126] In this embodiment of the present disclosure, the frame skipping processing of the service server can reduce the delay for receiving an image at the service server side, and improve image recognition efficiency. Through the two buffers of the target cloud application server, it is only necessary to switch the working modes of the two buffers (for example, switch pointers), without requiring data copy and repeated buffer allocation. In addition, data reception and data processing (for example, data reading and rendering) are performed at the same time without waiting, thereby reducing the delay. That is, the delay is further reduced by setting the two buffers, in addition to reducing the delay on the service server side.

[00127] For ease of understanding, reference is made to FIG. 10, which is a diagram of a system architecture according to an embodiment of the present disclosure. The diagram of the system architecture shown in FIG. 10 uses a cloud application as an example. A cloud application server corresponding to the cloud application may be a cloud game server. As shown in FIG. 10, the system architecture may include a client cluster (which may include a client 1, a client 2, ..., and a client n), a service server (which may include a pushing subserver and an image recognition subserver, where the pushing subserver may be configured to receive an encoded image file uploaded by a client and perform decoding processing on the encoded image file; and the image recognition subserver may perform image recognition processing on decoded image data obtained by the pushing subserver through decoding), and the cloud game server. For ease of understanding, the following makes specific descriptions.

[00128] The client cluster: when running the cloud application (for example, a cloud game application), each client may display a picture of the cloud application (for example, a picture of the cloud game application). When the cloud game application is run, a user picture is acquired by using a camera, encoding processing is performed, and obtained image data is uploaded to the pushing subserver in the service server (the pushing subserver may be any server with a data receiving function and a decoding function, and is mainly configured to receive an encoded image file uploaded by the client, and perform decoding processing on the encoded image file).

[00129] The pushing subserver: it may receive the image data uploaded by the client and perform decoding processing on the image data to obtain a decoded image in an initial image format (for example, a YUV mode), and may transmit the decoded image to the image recognition subserver.

[00130] The image recognition sub-server: it may convert the decoded image from the YUV mode to an RGB mode, then recognize and extract user portrait data or a human body key point in the image, and transmit the user portrait data or the human body key point to the cloud game server. It is to be understood that the pushing subserver and the image recognition subserver may form the service server, such that the service server has image decoding and image recognition functions. Certainly, the pushing subserver and the image recognition subserver may alternatively be implemented as independent servers to perform corresponding tasks respectively (that is, the pushing subserver receives and decodes the encoded image file, and the image recognition subserver performs image recognition processing on decoded data). It is to be understood that both the pushing subserver and the image recognition subserver may perform the frame skipping processing, so as to reduce the data receiving delay.

[00131] The cloud game server: the cloud game server may be a cloud application server corresponding to the client. When the client runs the cloud game application, the cloud game server provides a corresponding computing server for the client. The cloud game server may receive the user portrait data or the human body key points, and render the user portrait data. Alternatively, the cloud game server may operate a virtual cartoon character in the cloud game application by using the human body key points, to implement animation (that is, operate the virtual cartoon character to synchronize a real action status of the user, rather than projecting the user portrait to the cloud game application).

[00132] For ease of understanding, reference is made to FIG. 11, which is a schematic diagram of a system process according to an embodiment of the present disclosure. The process may correspond to the system architecture shown in FIG. 10. As shown in FIG. 11, the process may include S31 to S36:

[00133] S31: The client acquires a camera image.

[00134] S32: The client encodes the acquired image.

[00135] S33: The pushing subserver decodes the encoded image.

[00136] S34: The image recognition subserver converts the decoded image from the YUV mode into the RGB mode.

[00137] S35: The cloud game server recognizes a user portrait or human body key points.

[00138] S36: The portrait or a virtual animation is displayed.

[00139] Specific implementations of S31 to S36 may refer to the descriptions in the embodiments corresponding to FIG. 3, FIG. 5, FIG. 7, and FIG. 8, and will not be elaborated herein. Beneficial effects of S31 to S36 will also not be elaborated herein.

[00140] It may be understood that when acquiring an object, if the client also acquires another object at the same time (it may be understood that another user enters in the view field of camera), the client may generate object selection prompt information, and the user selects a to-be-collected object. Alternatively, the client automatically determines the to-be-collected object according to resolutions and areas of the objects. For example, the client acquires both an object 1 and an object 2. The object 2 is farer away from a camera, and a picture of the object 2 is not so clear. The object 1 is closer to the camera, and a picture of the object 1 is clear. Therefore, the client may automatically determine the object 1 as the to-be-collected object.

[00141] FIG. 12 is a diagram of an interaction process according to an embodiment of the present disclosure. The interaction process may be an interaction process between a client, a pushing subserver, an image recognition subserver, and a cloud application server (taking a cloud game server as an example). As shown in FIG. 12, the interaction process may include at least the following S41 to S54:

[00142] S41: The client establishes a connection with the pushing subserver.

[00143] When a user starts a cloud game application by using the client, the client may establish the connection (for example, establish a Websocket persistent connection) with the pushing subserver.

[00144] S42: The image recognition subserver establishes a connection with the cloud game server.

[00145] When the user opens the cloud game application by using the client, the cloud game server (which may be integrated with a cloud game software development kit (SDK)) may establish the connection (for example, establish a TCP connection) with the image recognition subserver.

[00146] S43: The cloud game server transmits an acquisition notification to the corresponding client.

[00147] The acquisition notification may be a notification message for starting image acquisition.

[00148] S44: The client transmits a pushing message to the pushing subserver.

[00149] The client may turn on a camera based on the acquisition notification, and notify the pushing subserver of a device identifier of the cloud game server, an identifier of the acquired user, and a width and a height of an image acquired by the camera, such that the pushing subserver is prepared to receive data. The pushing message may include the device identifier of the cloud game server, the identifier of the acquired user, and the width and the height of the image acquired by the camera.

[00150] S45: The pushing subserver transmits the pushing message to the image recognition subserver.

[00151] After receiving the pushing message of the client, the pushing subserver may establish a TCP connection with the image recognition subserver, and transmit the pushing message to the image recognition subserver.

[00152] S46: The client starts image acquisition, and performs encoding.

- [00153] S47: The client transmits encoded data to the pushing subserver.
- [00154] S48: The pushing subserver performs decoding to obtain decoded data.
- [00155] S49: The pushing subserver transmits the decoded data to the image recognition subserver.
- [00156] S50: The image recognition subserver performs format conversion on the decoded data, and performs image recognition.
- [00157] S51: The image recognition subserver transmits recognized data to the cloud game server.
- [00158] S52: The cloud game server renders the recognized data to obtain rendered data.
- [00159] S53: The cloud game server transmits the rendered data to the client.
- [00160] Specific implementations of S46 to S53 may refer to the descriptions in the embodiments corresponding to FIG. 3, FIG. 5, FIG. 7, and FIG. 8, and will not be elaborated herein. Beneficial effects of S46 to S53 will also not be elaborated herein.
- [00161] FIG. 13 is a schematic diagram of a structure of a data processing apparatus according to an embodiment of the present disclosure. The data processing apparatus may be a computer program (including program code) run in a computer device. For example, the data processing apparatus is application software. The data processing apparatus may be configured to perform the method shown in FIG. 3. As shown in FIG. 13, the data processing apparatus 1 may include a data obtaining module 11, an image recognition module 12, a queue update module 13, and a region transmission module 14.
- [00162] The data obtaining module 11 is configured to obtain first image data transmitted by a first client, and store the first image data to a reception queue. The first image data is image data that is obtained by the first client when running a cloud application and that includes an object.
- [00163] The image recognition module 12 is configured to perform image recognition processing on the first image data in the reception queue.
- [00164] The queue update module 13 is configured to store, during the image recognition processing on the first image data, second image data to the reception queue to obtain an updated reception queue, the second image data being image data continuously obtained from the first client.
- [00165] The region transmission module 14 is configured to transmit, in a case that a first object region of the object in the first image data is extracted through image recognition processing, first object image data in the first object region to a target cloud application server. The target cloud

application server is configured to perform rendering on the first object image data to obtain rendered data, and transmit the rendered data to the first client.

[00166] The region transmission module 14 is further configured to at the same time perform image recognition processing on second image data with a latest receiving timestamp in the updated reception queue.

[00167] Specific implementations of the data obtaining module 11, the image recognition module 12, the queue update module 13, and the region transmission module 14 may refer to the descriptions about S101 to S103 in the embodiment corresponding to FIG. 3, and will not be elaborated herein.

[00168] In an embodiment, the data obtaining module 11 may include an image receiving unit 111 and a storage unit 112.

[00169] The image receiving unit 111 is configured to receive the first image data transmitted by the first client. The first image data is data obtained by the first client by encoding an original image frame. The original image frame is acquired by the first client running the cloud application.

[00170] The storage unit 112 is configured to obtain a receiving timestamp of receiving the first image data, and store the first image data and the receiving timestamp associatively to the reception queue.

[00171] Specific implementations of the image receiving unit 111 and the storage unit 112 may refer to the descriptions about S101 to S103 in the embodiment corresponding to FIG. 3, and will not be elaborated herein.

[00172] In an embodiment, the image recognition module 12 may include a data decoding unit 121, a format conversion unit 122, and an image recognition unit 123.

[00173] The data decoding unit 121 is configured to perform decoding processing on the first image data to obtain decoded image data in an initial image format.

[00174] The format conversion unit 122 is configured to perform format conversion on the decoded image data to obtain the original image frame in a standard image format.

[00175] The image recognition unit 123 is configured to perform image recognition processing on the original image frame in the standard image format.

[00176] Specific implementations of the data decoding unit 121, the format conversion unit 122, and the image recognition unit 123 may refer to the descriptions about S101 in the embodiment corresponding to FIG. 3, and will not be elaborated herein.

[00177] In an embodiment, the image recognition unit 123 may include a key point recognition subunit 1231, a curve connection subunit 1232, and a region determining subunit 1233.

[00178] The key point recognition subunit 1231 is configured to recognize object edge key points of the object in the original image frame.

[00179] The curve connection subunit 1232 is configured to connect the object edge key points to obtain an object edge curve of the object.

[00180] The region determining subunit 1233 is configured to determine a region enclosed by the object edge curve in the original image frame as an initial object region of the object in the original image frame, and determine the first object region according to the initial object region.

[00181] Specific implementations of the key point recognition subunit 1231, the curve connection subunit 1232, and the region determining subunit 1233 may refer to the descriptions about S102 in the embodiment corresponding to FIG. 3, and will not be elaborated herein.

[00182] In an embodiment, the image recognition unit 123 is further configured to: obtain object recognition configuration information for the object and determine a configured object recognition part indicated by the object recognition configuration information, and determine whether the configured object recognition part matches the object key part; and perform, in a case that the configured object recognition part matches the object key part, the operation of determining the first object region according to the initial object region; and determine, in a case that the configured object recognition part does not match the object key part, that the first object region is not extracted through image recognition processing.

[00183] In an embodiment, the image recognition unit 123 is further configured to determine image data with an earliest receiving timestamp in the updated reception queue as to-be-recognized image data, and perform image recognition processing on the to-be-recognized image data. In a case that a to-be-processed object region of the object in the to-be-recognized image data is extracted through the image recognition processing, the region transmission module 14 is further configured to transmit the to-be-processed object region to the target cloud application server.

[00184] In an embodiment, the image recognition unit 123 is specifically configured to:
obtain an object key part presented by data in the initial object region of the object;
and

determine the initial object region as the first object region in a case that the object key part meets a part completeness requirement; and

obtain N sample image frames from a sample database in a case that the object key part does not meet the part completeness requirement, obtain a to-be-processed sample image frame corresponding to the object from the N sample image frames, and determine the first object region according to the to-be-processed sample image frame and the initial object region, N being a positive integer.

[00185] In an embodiment, the image recognition unit 123 is specifically configured to:

obtain overall part information in the to-be-processed sample image frame;

determine a to-be-fused part region in the overall part information according to the object key part; and

fuse the to-be-fused part region and the initial object region to obtain the first object region.

[00186] In an embodiment, the first image data includes a to-be-confirmed device identifier. The to-be-confirmed device identifier is a device identifier of a bound cloud application server. The bound cloud application server is in a binding relationship with the first client.

[00187] The region transmission module 14 includes a set obtaining unit 141 and an identifier matching unit 142.

[00188] The set obtaining unit 141 is configured to obtain a stored device identifier set. The stored device identifier set includes M stored device identifiers. Each stored device identifier corresponds to a registered cloud application server. M is a positive integer.

[00189] The identifier matching unit 142 is configured to determine, in a case that the M stored device identifiers include a stored device identifier matching the to-be-confirmed device identifier, that the bound cloud application server indicated by the to-be-confirmed device identifier is a registered cloud application server, determine the bound cloud application server indicated by the to-be-confirmed device identifier as the target cloud application server, and transmit the first object image data to the target cloud application server.

[00190] Specific implementations of the set obtaining unit 141 and the identifier matching unit 142 may refer to the descriptions about S103 in the embodiment corresponding to FIG. 3, and will not be elaborated herein.

[00191] In an embodiment, the data processing apparatus 1 may further include a registration request receiving module 15, an index detection module 16, and an identifier addition module 17.

[00192] The registration request receiving module 15 is configured to receive a registration request transmitted by a to-be-registered cloud application server. The registration request is generated by the to-be-registered cloud application server after receiving an application starting notification transmitted by a second client. The application starting notification is generated by the second client in response to an application starting operation on the cloud application.

[00193] The index detection module 16 is configured to detect device index information of the to-be-registered cloud application server according to the registration request.

[00194] The identifier addition module 17 is configured to obtain a to-be-stored device identifier of the to-be-registered cloud application server in a case that the device index information satisfies a processing quality condition, store the to-be-stored device identifier to the stored device identifier set, determine the to-be-registered cloud application server as a registered cloud application server, and determine the to-be-stored device identifier as a stored device identifier.

[00195] Specific implementations of the registration request receiving module 15, the index detection module 16, and the identifier addition module 17 may refer to the descriptions about S103 in the embodiment corresponding to FIG. 3, and will not be elaborated herein.

[00196] In an embodiment, the device index information includes a network quality parameter and a device version.

[00197] The index detection module 16 may include a parameter obtaining unit 161 and an index determining unit 162.

[00198] The parameter obtaining unit 161 is configured to obtain the network quality parameter and the device version of the to-be-registered cloud application server according to the registration request.

[00199] The index determining unit 162 is configured to determine, in a case that the network quality parameter reaches a parameter threshold and the device version matches a quality standard version, that the device index information satisfies the processing quality condition.

[00200] The index determining unit 162 is further configured to determine, in a case that the network quality parameter does not reach a parameter threshold or the device version does not match a quality standard version, that the device index information does not satisfy the processing quality condition.

[00201] Specific implementations of the parameter obtaining unit 161 and the index determining unit 162 may refer to the descriptions about S103 in the embodiment corresponding to FIG. 3, and will not be elaborated herein.

[00202] In this embodiment of the present disclosure, when obtaining the first image data including the object, a client (for example, the first client) may transmit the first image data to a related computer device (for example, the service server), the service server performs image recognition processing, instead of requiring the client to perform the image recognition locally. In this way, image recognition processing may be performed on the first image data by using the service server with a high computing capability, so that image recognition efficiency and a definition can be improved. In addition, in the present disclosure, the service server may store the received first image data to the reception queue, continuously obtain the second image data from the first client during the image recognition processing on the first image data, and store the second image data to the reception queue to obtain the updated reception queue. That is, in the present disclosure, the service server does not stop receiving the second image data during the image recognition processing on the first image data, and may implement image processing and image reception at the same time by using the reception queue. In this way, the image transmission delay can be reduced. Further, when the service server extracts, through image recognition processing, the first object region of the object in the first image data, the service server may transmit the first object image data in the first object region to the target cloud application server, and the target cloud application server performs rendering, and transmits the rendered data to the first client. In this way, displaying may be performed in the cloud application. In addition, after extracting the first object region, the service server may obtain the second image data with the latest receiving timestamp in the reception queue, and continue to process the second image data. It can be seen that in the present disclosure, image data is not recognized one by one according to a time sequence of receiving timestamps. Instead, image data with a latest receiving timestamp is obtained from the reception queue for processing after image recognition processing is performed on a piece of image data, so that image data recognition efficiency can be improved. In addition, since the image data with the latest receiving timestamp is acquired according to a current behavior of the object, by performing image recognition and displaying on the image data with the latest receiving timestamp, the displayed data matches the current behavior of the object. In summary, according to the present disclosure, the image recognition efficiency is improved, the image transmission delay is reduced, and it is ensured that a virtual behavior of a virtual object displayed in the cloud application matches a current behavior status of the object.

[00203] Refer to FIG. 14. FIG. 14 is a schematic diagram of a structure of another data processing apparatus according to an embodiment of the present disclosure. The data processing apparatus may be a computer program (including program code) run in a computer device. For example, the data processing apparatus is application software. The data processing apparatus may

be configured to perform the method shown in FIG. 8. As shown in FIG. 14, the data processing apparatus 2 may include a region storage module 21, a region rendering module 22, a region receiving module 23, and a status adjustment module 24.

[00204] The region storage module 21 is configured to receive first object image data transmitted by a service server, and store the first object image data to a first buffer whose working mode is a storage mode in a buffer set. The first object image data is image data in a first object region. The first object region is a region of an object in first image data and is obtained by the service server performing image recognition processing on the first image data. The first image data is transmitted by a first client to the service server. The first image data is image data including the object and obtained by the first client when running a cloud application.

[00205] The region rendering module 22 is configured to, when a second buffer whose working mode is a read mode in the buffer set does not comprise unprocessed object image data, adjust the working mode of the first buffer to the read mode and adjust the working mode of the second buffer to the storage mode, read the first object region from the first buffer whose working mode is the read mode, and perform rendering processing on the first object region.

[00206] The region receiving module 23 is configured to receive, during the rendering processing on the first object region, second object image data transmitted by the service server, and store the second object image data to the second buffer whose working mode is the storage mode. The second object image data is image data in a second object region. The second object region is obtained by the service server by performing image recognition processing on second image data after extracting the first object region. The second object region is a region of the object in the second image data. The second image data is image data having a latest receiving timestamp and obtained by the service server from an updated reception queue when the first object region is extracted. Second image data in the reception queue is continuously obtained by the service server from the first client during the image recognition processing on the first image data.

[00207] The status adjustment module 24 is configured to, when rendered data corresponding to the first object image data is obtained, adjust the working mode of the first buffer to the storage mode and adjust the working mode of the second buffer to the read mode, read the second object image data from the second buffer whose working mode is the read mode, and perform rendering processing on the second object image data.

[00208] Specific implementations of the region storage module 21, the region rendering module 22, the region receiving module 23, and the status adjustment module 24 may refer to the

descriptions about S801 to S804 in the embodiment corresponding to FIG. 8, and will not be elaborated herein.

[00209] In an embodiment, the status adjustment module 24 may include an identifier obtaining unit 241 and an identifier switching unit 242.

[00210] The identifier obtaining unit 241 is configured to obtain, when the rendered data corresponding to the first object image data is obtained, an indicator corresponding to the first buffer and being a read indicator representing the read mode and an indicator corresponding to the second buffer and being a storage indicator representing the storage mode.

[00211] The identifier switching unit 242 is configured to switch the indicator corresponding to the first buffer from the read indicator to the storage indicator, the working mode of the first buffer with the storage indicator being the storage mode.

[00212] The identifier switching unit 242 is further configured to switch the indicator corresponding to the second buffer from the storage indicator to the read indicator, the working mode of the second buffer with the read indicator being the read mode.

[00213] Specific implementations of the identifier obtaining unit 241 and the identifier switching unit 242 may refer to the descriptions about S804 in the embodiment corresponding to FIG. 8, and will not be elaborated herein.

[00214] In an embodiment, the data processing apparatus 2 may further include a buffer allocation module 25, an identifier setting module 26, and a set generation module 27.

[00215] The buffer allocation module 25 is configured to allocate the first buffer and the second buffer.

[00216] The identifier setting module 26 is configured to set initially the indicator of the first buffer to be the storage indicator, and setting initially the indicator of the second buffer to be a read indicator, the working mode of the first buffer with the storage indicator being the storage mode, and the working mode of the second buffer with the read indicator being the read mode.

[00217] The set generation module 27 is configured to generate the buffer set with the first buffer whose working mode is the storage mode and the second buffer whose working mode is the read mode.

[00218] Specific implementations of the buffer allocation module 25, the identifier setting module 26, and the set generation module 27 may refer to the descriptions about S801 in the embodiment corresponding to FIG. 8.

[00219] In this embodiment of the present disclosure, the frame skipping processing of the service server can reduce the delay for receiving an image at the service server side, and improve image recognition efficiency. Through the two buffers of the cloud game server, it is only necessary to switch the working modes of the two buffers, without requiring data copy and repeated buffer allocation.. In addition, data reception and data processing (for example, data reading and rendering) are performed at the same time without waiting, thereby reducing the delay. That is, the delay is further reduced by setting the two buffers in addition to reducing the delay on the service server side.

[00220] FIG. 15 is a schematic diagram of a structure of a computer device according to an embodiment of the present disclosure. As shown in FIG. 15, the apparatus 1 in the embodiment corresponding to FIG. 13 or the apparatus 2 in the embodiment corresponding to FIG. 14 may be applied to the computer device 8000. The computer device 8000 may include a processor 8001, a network interface 8004, and a memory 8005. In addition, the computer device 8000 further includes a user interface 8003 and at least one communication bus 8002. The communication bus 8002 is configured to implement connection communication between these components. The user interface 8003 may include a display and a keyboard. Optionally, the user interface 8003 may further include a standard wired interface and wireless interface. Optionally, the network interface 8004 may include a standard wired interface and wireless interface (for example, a wireless fidelity (Wi-Fi interface)). The memory 8005 may be a high-speed random access memory (RAM), or a non-volatile memory, for example, at least one disk memory. Optionally, the memory 8005 may be at least one storage apparatus far away from the processor 8001. As shown in FIG. 15, as a computer-readable storage medium, the memory 8005 may include an operating system, a network communication module, a user interface module, and a device control application program.

[00221] In the computer device 8000 shown in FIG. 15, the network interface 8004 may provide a network communication function. The user interface 8003 is mainly configured to provide an input interface for a user. The processor 8001 may be configured to call the device control application program stored in the memory 8005 to implement the data processing method provided in the foregoing embodiment.

[00222] It is to be understood that the computer device 8000 described in this embodiment of the present disclosure may execute the data processing methods in the described embodiments corresponding to FIG. 3 to FIG. 8, or may implement the data processing apparatus 1 in the embodiment corresponding to FIG. 13 or the data processing apparatus 2 in the embodiment

corresponding to FIG. 14. Elaborations are omitted herein. In addition, the description of beneficial effects is similar to that of the methods, and is not repeated herein.

[00223] Further, an embodiment of the present disclosure also provides a computer-readable storage medium. The computer-readable storage medium stores a computer program executed by the data processing computer device 8000 mentioned above. The computer program includes program instructions. The processor, when executing the program instructions, may execute the data processing methods in the described embodiments corresponding to FIG. 3 to FIG. 8. Therefore, elaborations are omitted herein. In addition, the description of beneficial effects is similar to that of the methods, and is not repeated herein. Technical details that are not disclosed in the embodiment of the computer-readable storage medium involved in the present disclosure refer to the descriptions in the method embodiments of the present disclosure.

[00224] The computer-readable storage medium may be an internal storage unit of the data processing apparatus or the computer device provided in any one of the foregoing embodiments, for example, a hard disk or internal memory of the computer device. The computer-readable storage medium may alternatively be an external storage device of the computer device, for example, a plug-in hard disk, a smart media card (SMC), a secure digital (SD) card, or a flash card on the computer device. Further, the computer-readable storage medium may alternatively include both an internal storage unit and an external storage device of the computer device. The computer-readable storage medium is configured to store the computer program and another program and data that are required by the computer device. The computer-readable storage medium may further be configured to temporarily store data that has been output or is to be output.

[00225] An aspect of the present disclosure provides a computer program product or computer program. The computer program product or computer program includes computer instructions. The computer instructions are stored in a computer-readable storage medium. A processor of a computer device reads the computer instructions from the computer-readable storage medium. The processor executes the computer instructions to enable the computer device to perform the method provided in an aspect of the embodiments of the present disclosure.

[00226] Terms "first", "second", and the like in the description, the claims, and the drawings in the embodiments of the present disclosure are used to distinguish between different objects rather than describe a specific sequence. In addition, term "include" and any variation thereof are intended to cover nonexclusive inclusions. For example, a process, method, apparatus, product, or device including a series of steps or units is not limited to the listed steps or modules, but may optionally

further include steps or modules that are not listed, or optionally further include other steps or units that are inherent to the process, the method, the apparatus, the product, or the device.

[00227] A person of ordinary skill in the art may further realize that the units and the algorithm steps in each example described with reference to the embodiments disclosed herein may be implemented by electronic hardware, computer software, or a combination thereof. For clarity of description about the exchangeability of hardware and software, the compositions and the steps in each example have been generally described in the foregoing description according to functions. Whether these functions are executed by hardware or software depends on specific applications and design constraints of the technical solutions. A person skilled in the art may use different methods to implement the described functions for each particular application, but it should not be considered that the implementation goes beyond the scope of the present disclosure.

[00228] The method and the related apparatus provided in the embodiments of the present disclosure are described with reference to the flowchart of the method and/or the schematic diagram of the structure provided in the embodiments of the present disclosure. Specifically, each process and/or box in the flowchart of the method and/or the schematic diagram of the structure and a combination of the processes and/or boxes in the flowchart and/or the block diagram may be implemented by computer program instructions. These computer program instructions may be provided for a general-purpose computer, a special-purpose computer, an embedded processor, or a processor of another programmable data processing device to generate a machine, so that an apparatus configured to realize functions specified in one or more processes in the flowchart and/or one or more boxes in the schematic diagram of the structure is generated by the instructions executed by the general-purpose computer or the processor of the another programmable data processing device. Alternatively, these computer program instructions may be stored in a computer-readable memory capable of guiding a computer or another programmable data processing device to work in a specific manner, so that a product including an instruction apparatus may be generated by the instructions stored in the computer-readable memory, the instruction apparatus realizing functions specified in one or more processes in the flowchart and/or one or more boxes in the schematic diagram of the structure. Alternatively, these computer program instructions may be loaded onto a computer or another programmable data processing device, so that a series of operating steps are executed in the computer or the another programmable data processing device to generate processing implemented by the computer, and steps for realizing functions specified in one or more processes in the flowchart and/or one or more boxes in the schematic diagram of the structure are provided by the instructions executed on the computer or the another programmable data processing device.

[00229] The above is only the preferred embodiment of the present disclosure and certainly not intended to limit the protection scope. Therefore, equivalent variations made according to the claims of the present disclosure also fall within the protection scope.

CLAIMS

What is claimed is:

1. A data processing method, the method being performed by a computer device, and the method comprising:

obtaining first image data transmitted by a first client, and storing the first image data to a reception queue, the first image data being image data comprising an object and obtained by the first client when running a cloud application;

performing image recognition processing on the first image data in the reception queue, and storing, during the image recognition processing on the first image data, second image data to the reception queue to obtain an updated reception queue, the second image data being image data continuously obtained from the first client; and

transmitting, in a case that a first object region of the object in the first image data is extracted through the image recognition processing, first object image data in the first object region to a target cloud application server, and at the same time performing image recognition processing on second image data having a latest receiving timestamp in the updated reception queue, the target cloud application server being configured to perform rendering on the first object image data to obtain rendered data, and transmit the rendered data to the first client.

2. The method according to claim 1, wherein the obtaining first image data transmitted by a first client and storing the first image data to a reception queue comprises:

receiving the first image data transmitted by the first client, the first image data being data obtained by the first client by encoding an original image frame, and the original image frame being acquired by the first client running the cloud application; and

obtaining a receiving timestamp of receiving the first image data, and storing the first image data and the receiving timestamp associatively to the reception queue.

3. The method according to claim 2, wherein the performing image recognition processing on the first image data in the reception queue comprises:

decoding the first image data to obtain decoded image data in an initial image format;

performing format conversion on the decoded image data to obtain the original image frame in a standard image format; and

performing image recognition processing on the original image frame in the standard image format.

4. The method according to claim 3, wherein the performing image recognition processing on the original image frame in the standard image format comprises:

recognizing object edge key points of the object in the original image frame;

connecting the object edge key points to obtain an object edge curve of the object;

determining a region enclosed by the object edge curve in the original image frame as an initial object region of the object in the original image frame; and

determining the first object region according to the initial object region.

5. The method according to claim 4, wherein before the determining the first object region according to the initial object region, the method further comprises:

obtaining an object key part presented by data in the initial object region of the object;

obtaining object recognition configuration information for the object and determining a configured object recognition part indicated by the object recognition configuration information, and determining whether the configured object recognition part matches the object key part;

performing, in a case that the configured object recognition part matches the object key part, the operation of determining the first object region according to the initial object region; and

determining, in a case that the configured object recognition part does not match the object key part, that the first object region is not extracted through the image recognition processing.

6. The method according to claim 5, wherein after the determining, in a case that the configured object recognition part does not match the object key part, that the first object region is not extracted through the image recognition processing, the method further comprises:

determining image data with an earliest receiving timestamp in the updated reception queue as to-be-recognized image data; and

performing image recognition processing on the to-be-recognized image data, and transmitting, in a case that a to-be-processed object region of the object in the to-be-recognized image data is extracted through the image recognition processing, the to-be-processed object region to the target cloud application server.

7. The method according to claim 4, wherein the determining the first object region according to the initial object region comprises:

obtaining an object key part presented by data in the initial object region of the object; and

determining the initial object region as the first object region in a case that the object key part

meets a part completeness requirement; and

obtaining N sample image frames from a sample database in a case that the object key part does not meet the part completeness requirement, obtaining a to-be-processed sample image frame corresponding to the object from the N sample image frames, and determining the first object region according to the to-be-processed sample image frame and the initial object region, N being a positive integer.

8. The method according to claim 7, wherein the determining the first object region according to the to-be-processed sample image frame and the initial object region comprises:

obtaining overall part information in the to-be-processed sample image frame;

determining a to-be-fused part region in the overall part information according to the object key part; and

fusing the to-be-fused part region and the initial object region to obtain the first object region.

9. The method according to claim 1, wherein the first image data comprises a to-be-confirmed device identifier; the to-be-confirmed device identifier is a device identifier of a bound cloud application server, and the bound cloud application server is in a binding relationship with the first client; and

the transmitting first object image data in the first object region to a target cloud application server comprises:

obtaining a stored device identifier set, the stored device identifier set comprising M stored device identifiers, each stored device identifier corresponding to a registered cloud application server, and M being a positive integer; and

determining, in a case that the M stored device identifiers comprise a stored device identifier matching the to-be-confirmed device identifier, that the bound cloud application server indicated by the to-be-confirmed device identifier is a registered cloud application server, determining the bound cloud application server indicated by the to-be-confirmed device identifier as the target cloud application server, and transmitting the first object image data in the first object region to the target cloud application server.

10. The method according to claim 9, wherein the method further comprises:

receiving a registration request transmitted by a to-be-registered cloud application server, the registration request being generated by the to-be-registered cloud application server after receiving an application starting notification transmitted by a second client, and the application starting

notification being generated by the second client in response to an application starting operation on the cloud application;

detecting device index information of the to-be-registered cloud application server according to the registration request; and

obtaining a to-be-stored device identifier of the to-be-registered cloud application server in a case that the device index information satisfies a processing quality condition, storing the to-be-stored device identifier to the stored device identifier set, determining the to-be-registered cloud application server as a registered cloud application server, and determining the to-be-stored device identifier as a stored device identifier.

11. The method according to claim 10, wherein the device index information comprises a network quality parameter and a device version; and

the detecting device index information of the to-be-registered cloud application server according to the registration request comprises:

obtaining the network quality parameter and the device version of the to-be-registered cloud application server according to the registration request;

determining, in a case that the network quality parameter reaches a parameter threshold and the device version matches a quality standard version, that the device index information satisfies the processing quality condition; and

determining, in a case that the network quality parameter does not reach a parameter threshold or the device version does not match a quality standard version, that the device index information does not satisfy the processing quality condition.

12. The method according to claim 1, wherein the performing image recognition processing on second image data having a latest receiving timestamp in the updated reception queue comprises:

obtaining the second image data having the latest receiving timestamp in the updated reception queue; and

performing image recognition processing on the second image data, and deleting historical image data in the updated reception queue, the historical image data being image data whose receiving timestamp is earlier than that of the second image data in the updated reception queue.

13. A data processing method, the method being performed by a computer device, and the method comprising:

receiving first object image data transmitted by a service server, and storing the first object

image data to a first buffer whose working mode is a storage mode in a buffer set, the first object image data being image data in a first object region, the first object region being a region of an object in first image data and being obtained by the service server performing image recognition processing on the first image data, the first image data being transmitted by a first client to the service server, and the first image data being image data comprising the object and obtained by the first client when running a cloud application;

when a second buffer whose working mode is a read mode in the buffer set does not comprise unprocessed object image data, adjusting the working mode of the first buffer to the read mode and adjusting the working mode of the second buffer to the storage mode, reading the first object image data from the first buffer whose working mode is the read mode, and performing rendering processing on the first object image data;

receiving, during the rendering processing on the first object image data, second object image data transmitted by the service server, and storing the second object image data to the second buffer whose working mode is the storage mode, the second object image data being image data in a second object region, the second object region being obtained by the service server by performing image recognition processing on second image data after extracting the first object region, the second object region being a region of the object in the second image data, the second image data being image data having a latest receiving timestamp and obtained by the service server from an updated reception queue when the first object region is extracted, and second image data in the updated reception queue being continuously obtained by the service server from the first client during the image recognition processing on the first image data; and

when rendered data corresponding to the first object image data is obtained, adjusting the working mode of the first buffer to the storage mode and adjusting the working mode of the second buffer to the read mode, reading the second object image data from the second buffer whose working mode is the read mode, and performing rendering processing on the second object image data.

14. The method according to claim 13, wherein the when rendered data corresponding to the first object image data is obtained, adjusting the working mode of the first buffer to the storage mode and adjusting the working mode of the second buffer to the read mode comprises:

obtaining, when the rendered data corresponding to the first object image data is obtained, an indicator corresponding to the first buffer and being a read indicator representing the read mode and an indicator corresponding to the second buffer and being a storage indicator representing the storage mode;

switching the indicator corresponding to the first buffer from the read indicator to the storage indicator, the working mode of the first buffer with the storage indicator being the storage mode; and

switching the indicator corresponding to the second buffer from the storage indicator to the read indicator, the working mode of the second buffer with the read indicator being the read mode.

15. The method according to claim 13, wherein the method further comprises:

allocating the first buffer and the second buffer;

setting initially the indicator of the first buffer to be the storage indicator, and setting initially the indicator of the second buffer to be a read indicator, the working mode of the first buffer with the storage indicator being the storage mode, and the working mode of the second buffer with the read indicator being the read mode; and

generating the buffer set with the first buffer whose working mode is the storage mode and the second buffer whose working mode is the read mode.

16. A data processing apparatus, the apparatus being deployed on a computer device, and the apparatus comprising:

a data obtaining module, configured to obtain first image data transmitted by a first client, and store the first image data to a reception queue, the first image data being image data comprising an object and obtained by the first client when running a cloud application;

an image recognition module, configured to perform image recognition processing on the first image data in the reception queue;

a queue update module, configured to store, during the image recognition processing on the first image data, second image data to the reception queue to obtain an updated reception queue, the second image data being image data continuously obtained from the first client; and

a region transmission module, configured to transmit, in a case that a first object region of the object in the first image data is extracted through the image recognition processing, first object image data in the first object region to a target cloud application server,

the region transmission module being further configured to at the same time perform image recognition processing on second image data having a latest receiving timestamp in the updated reception queue, and the target cloud application server being configured to perform rendering on the first object image data to obtain rendered data, and transmit the rendered data to the first client.

17. A data processing apparatus, the apparatus being deployed on a computer device, and the

apparatus comprising:

a region storage module, configured to receive first object image data transmitted by a service server, and store the first object image data to a first buffer whose working mode is a storage mode in a buffer set, the first object image data being image data in a first object region, the first object region being a region of an object in first image data and being obtained by the service server performing image recognition processing on the first image data, the first image data being transmitted by a first client to the service server, and the first image data being image data comprising the object and obtained by the first client when running a cloud application;

a region rendering module, configured to, when a second buffer whose working mode is a read mode in the buffer set does not comprise unprocessed object image data, adjust the working mode of the first buffer to the read mode and adjust the working mode of the second buffer to the storage mode, read the first object image data from the first buffer whose working mode is the read mode, and perform rendering processing on the first object region;

a region receiving module, configured to receive, during the rendering processing on the first object region, second object image data transmitted by the service server, and store the second object image data to the second buffer whose working mode is the storage mode, the second object image data being image data in a second object region, the second object region being obtained by the service server by performing image recognition processing on second image data after extracting the first object region, the second object region being a region of the object in the second image data, the second image data being image data having a latest receiving timestamp and obtained by the service server from an updated reception queue when the first object region is extracted, and second image data in the updated reception queue being continuously obtained by the service server from the first client during the image recognition processing on the first image data; and

a status adjustment module, configured to, when rendered data corresponding to the first object image data is obtained, adjust the working mode of the first buffer to the storage mode and adjust the working mode of the second buffer to the read mode; read the second object image data from the second buffer whose working mode is the read mode; and perform rendering processing on the second object image data.

18. A computer device, comprising: a processor, a memory, and a network interface,

the processor being connected to the memory and the network interface, the network interface being configured to provide a network communication function, the memory being configured to

store program code, and the processor is configured to call the program code to cause the computer device to perform the method according to any one of claims 1 to 15.

19. A computer-readable storage medium, the computer-readable storage medium storing a computer program, and the computer program, when being loaded and executed by a processor, causes the processor to implement the method according to any one of claims 1 to 15.

20. A computer program product or computer program, the computer program product or computer program comprising computer instructions, the computer instructions being stored in a computer-readable storage medium, and the computer instructions , when being loaded and executed by a processor, causes a computer device with the processor to perform the method according to any one of claims 1 to 15.