(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
05 January 2023 (05.01.2023)

WIPO | PCT

(10) International Publication Number
WO 2023/277907 A1

(51) International Patent Classification:
G06V 10/22 (2022.01) *G06T 19/20* (2011.01) *G06T 15/08* (2011.01)

(21) International Application Number: PCT/US2021/039871

(22) International Filing Date: 30 June 2021 (30.06.2021)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P.** [US/US]; 10300 Energy Drive, Spring, Texas 77389 (US).

(72) Inventors: **LIN, Qian**; 1501 Page Mill Rd., Palo Alto, California 94304 (US). **NEVES, Deangeli Gomes**; Av. Alan Turing, 275, 13083-898 Sao Paulo (BR). **VIANA, Tharsis Salathiel de Souza**; Av. Alan Turing, 275, 13083 Sao Paulo (BR). **VALENTE, Augusto**; Av. Alan Turing, 275, 13083-898 Sao Paulo (BR). **ALVES MENDES MOTTA, Cezanne**; Av. Alan Turing, 275, 13083-898 Sao Paulo (BR).

(74) Agent: **JENNEY, Michael** et al.; HP Inc., 3390 E. Harmony Road, Mail Stop 35, Fort Collins, Colorado 80528 (US).

(81) **Designated States** *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, IT, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) **Designated States** *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:
— *as to the identity of the inventor (Rule 4.17(i))*
— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

(54) **Title:** SYNTHETIC IMAGES FOR OBJECT DETECTION

(57) **Abstract:** In one example in accordance with the present disclosure, an electronic device is described. An example electronic device includes a processor and memory storing executable instructions that when executed cause the processor to receive a three-dimensional (3D) computer aided design (CAD) model of an object. The instructions also cause the processor to generate multiple synthetic images of the object from the 3D CAD model based on a 3D scene and randomized visual parameters. The instructions further cause the processor to generate annotations for the multiple synthetic images using the 3D CAD model.

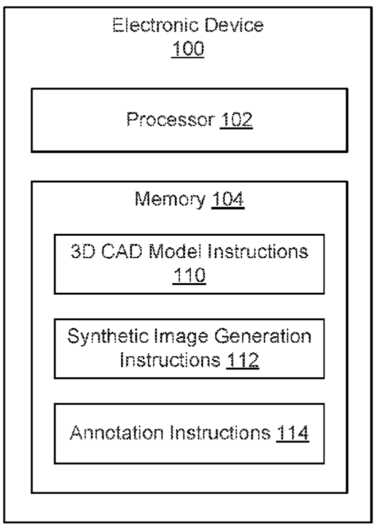

Fig. 1

WO 2023/277907 A1

[Continued on next page]

Published:
— *with international search report (Art. 21(3))*

SYNTHETIC IMAGES FOR OBJECT DETECTION

BACKGROUND

[0001] Electronic technology has advanced to become virtually ubiquitous in society and has been used to enhance many activities in society. For example, electronic devices are used to perform a variety of tasks, including work activities, communication, research, and entertainment. Different varieties of electronic circuits may be utilized to provide different varieties of electronic technology.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] The accompanying drawings illustrate various examples of the principles described herein and are part of the specification. The illustrated examples are given merely for illustration, and do not limit the scope of the claims.

[0003] Fig. 1 is a block diagram of an electronic device to generate synthetic images and annotations, according to an example.

[0004] Fig. 2 is a flow diagram illustrating a method for generating synthetic images and annotations, according to an example.

[0005] Fig. 3 is a flow diagram illustrating another method for generating synthetic images and annotations, according to an example.

[0006] Fig. 4 is a flow diagram illustrating yet another method for generating synthetic images and annotations, according to an example.

[0007] Fig. 5 illustrates a parallel synthetic image generation approach, according to an example.

[0008] Fig. 6 illustrates another parallel synthetic image generation approach, according to an example.

[0009] Fig. 7 depicts a non-transitory machine-readable storage medium for generating synthetic images and annotations, according to an example.

[0010] Throughout the drawings, identical reference numbers designate similar, but not necessarily identical, elements. The figures are not necessarily to scale, and the size of some parts may be exaggerated to more clearly illustrate the example shown. Moreover, the drawings provide examples and/or implementations consistent with the description; however, the description is not limited to the examples and/or implementations provided in the drawings.

DETAILED DESCRIPTION

[0011] Electronic devices may include memory resources and processing resources to perform computing tasks. For example, memory resources may include volatile memory (e.g., random access memory (RAM)) and non-volatile memory (e.g., read-only memory (ROM), data storage devices (e.g., hard drives, solid-state devices (SSDs), etc.) to store data and instructions. In some examples, processing resources may include circuitry to execute instructions. Examples of processing resources include a central processing unit (CPU), a graphics processing unit (GPU), or other hardware device that executes instructions.

[0012] An electronic device may be a device that includes electronic circuitry. For instance, an electronic device may include integrated circuitry (e.g., transistors, digital logic, semiconductor technology, etc.). Examples of electronic devices include computing devices, workstations, servers, laptop computers, desktop computers, smartphones, tablet devices, wireless communication devices, game consoles, game controllers, smart appliances, printing devices, vehicles with electronic components, aircraft, drones, robots, smart appliances, etc.

[0013] In some examples, electronic devices may be used for object detection. In some examples of object recognition, an image may be captured by a camera and an object may be detected in the image. In some examples, artificial intelligence techniques may be used to perform object detection. For example, a machine-learning (ML) model (also referred to as deep learning) may be trained to detect objects in images.

[0014] ML-based approaches may use many (e.g., hundreds or thousands) of examples per object type for training. For example, for a given object, hundreds or thousands of training images may be used to train an ML model to detect the object in the images. Furthermore, the object may be annotated in each image. In some examples, these annotations may include an object description (e.g., object name), bounding box, a segmentation map, or a combination thereof. Therefore, it may be time consuming and expensive to produce training datasets to train ML learning models for custom object types, such as industrial parts. For example, numerous pictures may be taken with the object in different positions, orientations, settings, etc. Then, each of the pictures are annotated. As seen by this example, the annotation process may be especially time consuming when pixel-wise detection (e.g., segmentation maps) is desired, as the object contour has to be carefully drawn.

[0015] To avoid the difficulty, time and expense of generating training image datasets, the examples described herein generate training images from three-dimensional (3D) computer-aided design (CAD) models using computer graphics rendering techniques. The described examples are able to automatically generate training data for object detection in a short amount of time (e.g., a few hours) on an electronic device.

[0016] As mentioned, ML-based techniques use a large amount of training data to train an ML model. However, in many scenarios, acquiring the training data is time-consuming. In these scenarios, synthetic data generation can be employed to generate the data for training. In some approaches, synthetic data may be formed from 3D rendering engines. However, many of these approaches do not perform randomization in features in the 3D environment (e.g., illumination, context, background objects, etc.) and randomization in the object of interest (e.g., textures, 6-Dof pose, camera position, etc.). This variability in the data generation may be used to train ML models to be more robust for detecting objects in the real world.

[0017] Some approaches may use photogrammetry (also referred to as 3D scanning) in which a 3D model is produced through scanning of a real object. However, the process of photogrammetry may generate an imperfect representation of the real object and these imperfections can introduce a bias in the machine-learning model. Thus, the examples described herein, use a 3D CAD model for generating synthetic images for a training dataset.

[0018] The examples described herein provide a framework to train a machine learning-based computer vision system, for object detection and segmentation, using synthetically rendered images from 3D CAD models. To train the machine learning models, an automated process may be used to generate the training dataset. This process may be applied to generate a training dataset for any object that has a 3D representation. The synthetic image training dataset may be applied to many implementations in artificial intelligence, such as machine vision solutions for identifying 3D parts in manufacturing (e.g., additive manufacturing), and retail solutions (e.g., auto-checkout of produce or other items in grocery stores).

[0019] In some aspects, the 3D CAD model representation of objects may be obtained. For example, in industrial contexts, a 3D CAD model may be used for additive manufacturing. The 3D CAD model may be placed in different 3D environments (referred to herein as 3D scenes). This way, visual aspects, such as illumination, are made coherent and consistent within the synthetic images. To produce a robust machine learning model, many aspects, such as textures, illumination, and occlusion may be randomized during the creation of the synthetic images.

[0020] The present specification describes examples of an electronic device. The electronic device includes a processor and memory storing instructions that cause the processor to receive a 3D CAD model of an object. The instructions also cause the processor to generate multiple synthetic images of the object from the 3D CAD model based on a 3D scene and randomized visual parameters. The instructions further cause the processor to generate annotations for the multiple synthetic images using the 3D CAD model.

[0021] In another example, the present specification also describes a non-transitory machine-readable storage medium that includes instructions, when executed by a processor of an electronic device, cause the processor to receive a 3D CAD model of an object and multiple 3D scenes. The instructions also cause the processor to generate multiple synthetic images of the object from the 3D CAD model and the multiple 3D scenes based on randomized visual parameters. The instructions further cause the processor to generate annotations for the multiple synthetic images using the 3D CAD model. The instructions also cause the processor to train an ML model for detecting an object in an image captured by a camera based on the multiple synthetic images and annotations.

[0022] In yet another example, the present specification also describes a method that includes generating multiple synthetic images of an object based on a 3D CAD model of the object, a 3D scene and randomized visual parameters. The method also includes generating annotations for the multiple synthetic images using the 3D CAD model. The method further includes training an ML model to detect the object in the multiple synthetic images based on the annotations. The method additionally includes running the ML model to detect an observed object in an image captured by a camera.

[0023] As used in the present specification and in the appended claims, the term “processor” may be a processor resource, a controller, an application-specific integrated circuit (ASIC), a semiconductor-based microprocessor, a central processing unit (CPU), and a field-programmable gate array (FPGA), and/or other hardware device that executes instructions.

[0024] As used in the present specification and in the appended claims, the term “memory” may include a computer-readable storage medium, which computer-readable storage medium may contain, or store computer-usable program code for use by or in connection with an instruction execution system, apparatus, or device. The memory may take many types of memory including volatile memory (e.g., RAM) and non-volatile memory (e.g., ROM).

[0025] As used in the present specification and in the appended claims, the term “data storage device” may include a non-volatile computer-readable storage medium. Examples of the data storage device include hard disk drives, solid-state drives, writable optical memory disks, magnetic disks, among others. The executable instructions may, when executed by the respective component, cause the component to implement the functionality described herein.

[0026] Turning now to the figures, Fig. 1 is a block diagram of an electronic device 100 to generate synthetic images and annotations, according to an example. As used herein, examples of an electronic device 100 may include computing devices, workstations, servers, laptop computers, desktop computers, smartphones, tablet devices, wireless communication devices, game consoles, game controllers, smart appliances, printing devices, vehicles with electronic components, aircraft, drones, robots, smart appliances, or other devices having memory resources and processing resources.

[0027] As described above, the electronic device 100 includes a processor 102. The processor 102 of the electronic device 100 may be implemented as dedicated hardware circuitry or a virtualized logical processor. The dedicated hardware circuitry may be implemented as a central processing unit (CPU). A dedicated hardware CPU may be implemented as a single to many-core general purpose processor. A dedicated hardware CPU may also be implemented as a multi-chip solution, where more than one CPU are linked through a bus and schedule processing tasks across the more than one CPU.

[0028] A virtualized logical processor may be implemented across a distributed computing environment. A virtualized logical processor may not have a dedicated piece of hardware supporting it. Instead, the virtualized logical processor may have a pool of resources supporting the task for which it was provisioned. In this implementation, the virtualized logical processor may be executed on hardware circuitry; however, the hardware circuitry is not dedicated. The hardware circuitry may be in a shared environment where utilization is time sliced. Virtual machines (VMs) may be implementations of virtualized logical processors.

[0029] In some examples, a memory 104 may be implemented in the electronic device 100. The memory 104 may be dedicated hardware circuitry to host instructions for the processor 102 to execute. In another implementation, the memory 104 may be virtualized logical memory. Analogous to the processor 102, dedicated hardware circuitry may be implemented with dynamic random-access memory (DRAM) or other hardware implementations for storing processor instructions. Additionally, the virtualized logical memory may be implemented in an abstraction layer which allows the instructions to be executed on a virtualized logical processor, independent of any dedicated hardware implementation.

[0030] The electronic device 100 may also include instructions. The instructions may be implemented in a platform specific language that the processor 102 may decode and execute. The instructions may be stored in the memory 104 during execution. In some examples, the instructions may include 3D CAD model instructions 110, synthetic image generation instructions 112, and annotation instructions 114, according to the examples described herein.

[0031] As described above, there are scenarios where the training dataset acquisition and preparation is costly or impractical. For example, to train an ML model for object detection, a training dataset may include many (e.g., thousands) of different images where each image is annotated with training information (e.g., object type, object bounding box, object segmentation map, etc.). In such scenarios, the described examples of synthetic image generation may be employed to create training data for an ML model.

[0032] In some examples, synthetic images may be generated data using a 3D rendering engine (e.g., BLENDER, MAYA, UNREAL ENGINE etc.). These 3D rendering engines may provide a set of tools that can be used to create realistic virtual spaces and environments that can be used to generate synthetic images.

[0033] The described examples enhance the performance of 3D rendering engines. For example, these examples provide for generating multiple virtual spaces with multiple textured objects and lighting variations. These examples may also mitigate the domain shift between the synthetic images and the real images.

[0034] In some examples, the processor 102 may execute the 3D CAD model instructions 110 to cause the processor 102 to receive a 3D CAD model of an object. In some examples, the 3D CAD model may be stored in memory 104 and retrieved by the processor 102. As used herein, a 3D CAD model includes data to represent an object in 3D. Examples of 3D CAD models include 3D meshes, 3D wireframes, 3D solids. In some examples, a 3D CAD model may be generated by a CAD program (e.g., AUTOCAD, SOLIDWORKS) a 3D rendering engine (e.g., BLENDER, MAYA, etc.) or other 3D modeling program.

[0035] In some examples, the processor 102 may execute the synthetic image generation instructions 112 to cause the processor 102 to generate multiple synthetic images of the object from the 3D CAD model based on a 3D scene and randomized visual parameters. To mitigate the domain shift between the synthetic images and real images (e.g., RGB images captured by a camera), the synthetic images may be generated to appear realistic. To accomplish this, the synthetic image generation may provide images of satisfactory realism. For example, the synthetic image generation may provide high-quality textured objects, a variety of 3D environments (e.g., to generate a variety of backgrounds), photorealism, physics (e.g., collision, gravity, etc.), or a combination thereof.

[0036] In some examples, the 3D scene may be a 3D model of an environment. For example, the 3D scene may be created using a 3D modeling program to depict an environment in which the target object may be placed. Examples of a 3D scene include an office setting, a factory setting, a home setting, etc. The 3D scene may include multiple 3D features. For example, in the case of an office setting, the 3D scene may include 3D models of a desk, chair, bookcase, walls, windows, lamps, etc. In the case of a factory setting, the 3D scene may include machinery, conveyor belts, shelving, walls, ceilings, lighting, etc.

[0037] The 3D CAD model and the 3D scene may be provided to a 3D rendering engine. For example, the 3D rendering engine may load the 3D CAD model and the 3D scene to generate the synthetic images.

[0038] In some examples, the processor 102 may also provide randomized visual parameters to the 3D rendering engine. The randomized visual parameters may include instructions for adjusting the visual appearance of the synthetic images. In some examples, the randomized visual parameters may be provided to the 3D rendering engine via an application programming interface (API). The API may allow for varying the visual parameters in the 3D scene and the 3D CAD model without manual intervention by a user.

[0039] In some examples, a randomized visual parameter may include the object position. For example, the object position may be a location that the 3D CAD model is placed within a 3D scene108. In some examples, the processor 102 may randomly select different locations for placing the object in the 3D scene in the multiple synthetic images. For example, for each of the synthetic images, the processor 102 may randomly select a different location for placing the object in the 3D scene. In other words, the spot where the object is located in 3D scene may be randomized such that the 3D CAD model is in a different location in each of the synthetic images.

[0040] In some examples, the object position may account for the physical interaction of the 3D CAD model with surfaces in the 3D scene. For example, the 3D CAD model may be positioned in open areas (e.g., unoccupied volumes) of the 3D scene.

[0041] In some examples, a randomized visual parameter may include the distance and position of the virtual camera with respect to the object. As used herein, the virtual camera is the viewpoint used by a 3D rendering engine to produce a synthetic image. The virtual camera may simulate visual features of a real camera (e.g., focal length, focus, field of view, etc.). In some examples, the distance of the virtual camera may be the simulated space between the 3D CAD model and the virtual camera. In some examples, the position of the virtual camera may be coordinates in 3D space for the virtual camera. The position of the virtual camera may also include the 3D pose of the camera with respect to the 3D CAD model. In some examples, the processor 102 may randomly select distances and positions of the virtual camera with respect to the object. For example, for each of the synthetic images, the processor 102 may randomly select a different distance and position of the virtual camera with respect to the object.

[0042] In some examples, a randomized visual parameter may include the texture of the object, the 3D scene, or a combination thereof. For example, the memory 104 may store a library of textures. In other examples, the library of textures may be included in a 3D rendering engine. In some examples, the textures may represent different materials (e.g., metal, stone, wood, glass, fabric, etc.). In some examples, the processor 102 may randomize textures for the object in the 3D scene. For example, for each of the synthetic images, the processor 102 may randomly select a different texture for the 3D CAD model.

[0043] In some examples, a randomized visual parameter may include the dimensions of the object in the synthetic images. For example, the object may be depicted with different shapes and sizes by adjusting the dimensions of the 3D CAD model. In some examples, the processor 102 may randomize dimensions of the object in the synthetic images. For example, for each of the synthetic images, the processor 102 may randomly adjust one or multiple dimensions of the 3D CAD model. In some examples, the object may be constrained by a scale of the object within the 3D scene. In other words, changes in the dimensions of the 3D CAD model may be bounded by the context of the 3D scene. Therefore, when generating randomized dimensions of the 3D CAD model, the processor 102 may ensure that the 3D CAD model fits within the 3D scene. In other examples, the randomized dimensions may be allowed to change within a certain amount (e.g., a percentage) of the original 3D CAD model.

[0044] In some examples, a randomized visual parameter may include the lighting of the object, the 3D scene, or a combination thereof. For example, lighting of the 3D CAD model and the 3D scene may be simulated. The lighting may have a single or multiple sources (e.g., lights, windows, sun, etc.). In some examples, the light model that is used by the 3D rendering engine may include ray tracing. In some examples, the processor 102 may randomize lighting of the 3D scene containing the object for the multiple synthetic images. For example, for each of the synthetic images, the processor 102 may randomly select a different number of light sources, different locations of the light sources, different intensities of the light sources, etc. It should be noted that because the 3D CAD model is placed in a 3D scene, lighting of the 3D CAD model may affect the 3D scene and vice versa. For example, shadows from the 3D CAD model may appear on the 3D scene. In another example, shadows from features in the 3D scene may appear on the 3D CAD model.

[0045] In some examples, the processor 102 may generate multiple synthetic images of the object placed in multiple 3D scenes. For example, multiple 3D scenes may be stored in memory 104. The processor 102 may provide the multiple 3D scenes to the 3D rendering engine. Multiple synthetic images may be generated for each 3D scene. For example, the processor 102 may instruct the 3D rendering engine to apply randomized visual parameters to the 3D CAD model and a first 3D scene to produce a first set of synthetic images. The processor 102 may instruct the 3D rendering engine to apply randomized visual parameters to the 3D CAD model and a second 3D scene to produce a second set of synthetic images, and so forth.

[0046] In some examples, the processor 102 may generate an object grid in the synthetic images. For example, the processor 102 may instruct the 3D rendering engine to place a number of instances of the 3D CAD model in a grid. The object grid may include a number of rows and columns. The instances of the 3D CAD model may be separated by a specified amount. In some examples, the instances of the 3D CAD model may be placed near each other to simulate cases where a set of parts appear close together in a single image. Therefore, the object grid may simulate multiple instances of the object being placed in close proximity. For example, during additive manufacturing, multiple objects may be formed near each other. To help the ML model differentiate between closely spaced objects, the synthetic images may include an object grid in which the instances of the 3D CAD model are placed near each other.

The 3D rendering engine may render synthetic image to capture the object grid in the 3D scene.

[0047] In some examples, the processor 102 may generate a given number of synthetic images. For example, the number of synthetic images that the 3D rendering engine is to render may be specified (e.g., by a user). For example, the user may specify that 3,000 synthetic images are to be generated. In some examples, the number of synthetic images may be determined based on time constraints. For example, generating more synthetic images may produce better training results for the ML model at the expense of processing time. Therefore, a threshold number of synthetic images may be generated to ensure acceptable ML model performance.

[0048] In some examples, the processor 102 may execute the synthetic image generation instructions 112 to cause the processor 102 to generate annotations for the multiple synthetic images using the 3D CAD model. For example, the annotations may be ground truth information used to train the ML model about the object within the synthetic images. Because the synthetic images are generated using the 3D CAD model, the processor 102 may use information about the 3D CAD model to generate the annotations without human input.

[0049] In some examples, the annotations may include an object type. For example, the object type may be obtained from the 3D CAD model. In some examples, the object type may be the name of the 3D CAD model. In other examples, the object type may be included as metadata in the 3D CAD model file.

[0050] In some examples, the annotations may include a bounding box. For example, the processor 102 may determine the bounding box of the object in a given synthetic image using the 3D CAD model. The processor 102 may repeat this process for each of the synthetic images to determine a bounding box of the object for each synthetic image.

[0051] In some examples, the annotations may include a segmentation mask of the object in a given synthetic image. In some examples, a segmentation mask may be the contour (e.g., polygon) of an object that was generated and placed in a synthetic image.

[0052] The processor 102 may generate a segmentation mask of the object in each of the multiple synthetic images. In some examples, different colors can be used to generate the segmentation mask annotations. For example, the color of the 3D CAD model in a given synthetic image may be changed to a first color (e.g., blue). The remainder of the given synthetic image (e.g., the 3D scene) may be changed to a second color (e.g., red). A snapshot of the 3D CAD model may be captured using the same camera parameters as were used to create the given synthetic image. Color-thresholding may be used to obtain the contour (e.g., a surrounding polygon) of the selected object that is the foreground. In the case of multiple objects, multiple colors can be used to differentiate them.

[0053] In some examples, the annotations may be saved. For example, the bounding box, object type and segmentation mask for a given synthetic image may be saved and associated with the given synthetic image. In some examples, the annotations may be saved as a file (e.g., a JSON file) or metadata of the given synthetic image.

[0054] In some examples, the processor 102 may train an ML model for detecting an object in an image captured by a camera based on the multiple synthetic images and annotations. Examples of the ML models may be used include convolutional neural networks (CNNs) (e.g., basic CNN, R-CNN, Mask R-CNN, inception model, residual neural network, etc.) and recurrent neural networks (RNNs) (e.g., basic RNN, multi-layer RNN, bi-directional RNN, fused RNN, clockwork RNN, etc.). Some approaches may utilize a variant or variants of RNN (e.g., Long Short Term Memory Unit (LSTM), peephole LSTM, no input gate (NIG), no forget gate (NFG), no output gate (NOG), no input activation function (NIAF), no output activation function (NOAF), no peepholes (NP), coupled input and forget gate (CIFG), full gate recurrence (FGR), gated recurrent unit (GRU), etc.). Different depths (e.g., layers) of a neural network or multiple neural networks may be utilized in accordance with some examples of the techniques described herein.

[0055] The synthetic images and annotations may form a dataset for training the ML model. The synthetic images and annotations may be processed by the ML model to learn how to detect the object in the synthetic images. In some examples, the training refers to determining the best set of weights for maximizing the accuracy of the ML model. In some examples, to train the ML model, three types of data may be used: training data, test data, and validation data. The synthetic images may be used for each of the training data, test data, and validation data. For example, a part of the dataset may be reserved to validate the ML model. Before validation, the ML model may be tested with the test dataset.

[0056] In some examples, the dataset (i.e., the synthetic images) may be divided into a first portion (e.g., 60%) of training data and a second portion (e.g., 20%) of test data. In some examples, a third portion (e.g., 20%) of the training set may be used for a first validation before the definitive validation is performed. After training with the selected features, the ML model is evaluated. When the resulting metrics are satisfactory, the ML model may be saved.

[0057] In an example of training, a batch size may be set as 4 and the maximum number of iterations may be 720,000. A Stochastic Gradient Descent may be used with a base learning rate of 0.00125 and a weight decay of 0.0001. After all iterations, the best ML model may be selected based on the validation set performance.

[0058] In some examples, the processor 102 may create multiple subprocesses to divide the synthetic image generation and annotation. This approach may be referred to as data generation parallelization. In some examples, the processor 102 may start multiple subprocesses that can individually generate a subset of the synthetic images and annotations. Then, the processor 102 may merge the subsets of the multiple synthetic images and annotations generated by the multiple subprocesses into a combined dataset.

[0059] In some examples, each subprocess may include an instance of a 3D rendering engine to generate the subset of synthetic images. Each subprocess may use a fixed amount of computer resources (e.g., CPU, GPU, RAM).

[0060] In some examples, each subprocess may generate the synthetic images and annotations for a given 3D scene. In this approach, data generation may be scaled to leverage powerful hardware and reduce processing time. An example of this approach to parallelize the data generation is shown in Fig. 5.

[0061] In another example approach, the processor 102 may create multiple subprocesses where each subprocess generates a subset of synthetic images and annotations for a given 3D scene. Upon completion of the synthetic image generation and annotation for the given 3D scene, the subsets of synthetic images and annotations may be merged to form a dataset for that 3D scene. The processor 102 may then continue this process of creating subprocesses to generate synthetic images and annotations for each 3D scene. Once synthetic images and annotations have been generated for all of the 3D scenes, then the datasets may be merged to form a combined training dataset. An example of this approach is described in Fig. 6.

[0062] In some examples, once the ML model is trained using the synthetic images and annotations, the ML model may be used to detect an observed object in an image captured by a camera. For example, the ML model may be trained to detect objects produced by additive manufacturing process. In this case, the 3D CAD model may be an object that is to be produced in an additive manufacturing process. Thus, the same 3D CAD model that is used to make a part using an additive manufacturing process may be used to generate the synthetic images and annotations to train an ML model to detect the completed parts. A camera may capture an image of the completed parts. The ML model may then detect the completed parts in the captured image based on the training from the synthetic images and annotations.

[0063] In another example, the ML model may be used to detect objects in a retail environment. For example, the 3D CAD model may be an object that is representative of a type of object offered in a retail environment. For example, a retail store may use an automated checkout system where customers place their goods for purchase within view of a camera. The camera may capture images of these goods. In an example, the 3D CAD model may be a model of a good for sale. Multiple 3D CAD models may be used to train the ML model to detect different goods. For example, one 3D CAD model may be an apple, a second 3D CAD model may be a banana, a third 3D CAD model may be an orange, and so forth. The ML model may detect the real goods in a captured image based on the synthetic images and annotations generated from the 3D CAD models. Once the goods are detected, the customer may then be charged for these goods.

[0064] The examples described herein provide computer vision systems that are robust to illumination and background conditions due to the use of domain randomization techniques. The described examples also allow for synthetic training data to be generated and customized according to the target application (e.g., indoor or outdoor environments, fixed or varied object textures and colors, fixed or varied camera-to-object distance, and other parameters). In some examples, the synthetic image generation and annotation may be offered as a web service.

[0065] Fig. 2 is a flow diagram illustrating a method 200 for generating synthetic images and annotations, according to an example. In some examples, the method 200 may be performed by a processor, such as the processor 102 of Fig. 1.

[0066] At 202, a 3D CAD model may be received of an object. For example, the 3D CAD model includes data to represent an object in 3D. Examples of 3D CAD models include 3D meshes, 3D wireframes, 3D solids.

[0067] At 204, multiple synthetic images of the object may be generated from the 3D CAD model based on a 3D scene and randomized visual parameters. For example, the 3D scene may include a 3D model of an environment. The 3D scene may include multiple 3D features (e.g., surfaces, walls, light sources, etc.).

[0068] In some examples, the processor may randomly select different locations for placing the object in the 3D scene. For example, the 3D CAD model may be placed in a different random location in each synthetic image.

[0069] In some examples, the processor may randomly select distances and positions of a virtual camera with respect to the object. For example, the viewpoint of the object in each synthetic image may be randomly changed by varying the distance of the virtual camera from the 3D CAD model and position of the virtual camera in the 3D space.

[0070] In some examples, the processor may randomize textures for the object in the 3D scene. For example, the processor may randomly vary the texture of the 3D CAD model in each of the synthetic images. In some examples, the processor may select a random texture from a texture library. The selected texture may be applied to the 3D CAD model. In some examples, the processor may instruct the 3D rendering engine to vary the texture.

[0071] In some examples, the processor may randomize dimensions of the object in the synthetic images. For example, the processor may change the size of the 3D CAD model by random amounts. In some examples, the dimensions of the object may be constrained by a scale of the object within the 3D scene.

[0072] In some examples, the processor may randomize lighting of the 3D scene containing the object for the multiple synthetic images. For example, the processor may randomly select lighting parameters (e.g., a number of light sources, positions of the light sources, light intensities, etc.) for each of the synthetic images.

[0073] At 206, annotations may be generated for the multiple synthetic images using the 3D CAD model. For example, the 3D CAD model may be used as a ground truth in the synthetic images. In some examples, the annotations may include an object type, a bounding box, a segmentation mask, or a combination thereof.

[0074] Fig. 3 is a flow diagram illustrating another method 300 for generating synthetic images and annotations, according to an example. In some examples, the method 300 may be performed by a processor, such as the processor 102 of Fig. 1. In some examples, portions of the method 300 may be performed by different processors.

[0075] At 302, multiple synthetic images of an object may be generated from a 3D CAD model of the object, a 3D scene, and randomized visual parameters. For example, the 3D CAD model of the object may be received. One 3D scene or multiple 3D scenes may also be received. For example, each 3D scene may depict a different 3D environment. A processor may randomize the visual parameters used by a 3D rendering engine to generate the synthetic images. This may be accomplished as described in Fig. 2.

[0076] At 304, annotations may be generated for the multiple synthetic images using the 3D CAD model. In some examples, the annotations may include an object type, a bounding box, a segmentation mask, or a combination thereof.

[0077] At 306, an ML model may be trained to detect the object in the multiple synthetic images based on the annotations. In some examples, the ML may be a neural network. The synthetic images and annotations may be used to train the ML model to detect the object in the synthetic images.

[0078] At 308, the ML model may be run to detect an observed object in an image captured by a camera. For example, in image may be provided to the ML model, which processes the image to detect an object.

[0079] In some examples, the 3D CAD model may be an object that is to be produced in an additive manufacturing process. In this case, the ML model may be trained to detect objects produced by the additive manufacturing process using synthetic images and annotations generated as described above. An image may be captured upon completion of the additive manufacturing process. The image may be fed to the ML model for detection of parts produced by the additive manufacturing process.

[0080] In some examples, the 3D CAD model may be an object that is representative of a type of object offered in a retail environment. For example, the 3D CAD model may be an object that is offered for sale at a store. In this case, the ML model may be trained to detect the objects for sale. The ML model may receive images of items being purchased. The ML model may then detect the purchased items to facilitate the sale of the items.

[0081] Fig. 4 is a flow diagram illustrating yet another method 400 for generating synthetic images and annotations, according to an example. In some examples, the method 400 may be performed by a processor, such as the processor 102 of Fig. 1. In some examples, portions of the method 300 may be performed by different processors.

[0082] At 401, a 3D CAD model, 3D scenes, and textures 420 may be loaded into a 3D rendering engine. In some examples, a user may provide the 3D CAD model that defines the object of interest to be detected. In some examples, the user may provide rendering settings that define how the synthetic image rendering should be implemented by the 3D rendering engine. In case the rendering settings are not specified, the 3D rendering engine may use default settings. The 3D rendering engine then loads the 3D scenes where instances of the 3D CAD model will be spawned, and the textures 420 that could be applied to the 3D CAD model.

[0083] It should be noted that in some examples, multiple 3D CAD models may be loaded at 401. In this case, different 3D CAD models may be loaded to represent different objects of interest for object detection. In yet other examples, different 3D CAD models may be loaded at 401, where a subset of the 3D CAD models are objects of interest for object detection and other 3D CAD models are used to add variety to the synthetic images.

[0084] In some examples, scripts (e.g., Python scripts) may be used to automate the process of adjusting visual parameters, moving a virtual camera, and moving 3D CAD model through the 3D scene. In some examples, a library may be used to communicate the scripts to the 3D rendering engine. In some examples, an API may be used to randomly vary visual parameters in the 3D scene or for the 3D CAD model without any manual intervention.

[0085] At 403, a 3D scene may be selected. At 405, a 3D CAD model may be selected from among the 3D CAD models that were loaded at 401. This selected 3D CAD model may be referred to as the object of interest.

[0086] At 407, N additional instances of 3D CAD models may be selected. In some examples, N is a configurable number. In other examples, N is randomly determined by the processor. In some examples, the N additional instances may include copies of the 3D CAD model selected at 405. In some examples, the N additional instances may include instances of different 3D CAD models than the 3D CAD model selected at 405.

[0087] At 409, a surface location in the 3D scene may be randomly selected. In the case of the *N* additional instances, *N* additional surface locations may be randomly selected for the *N* additional instances.

[0088] At 411, the 3D CAD model instances may be placed at the selected surface locations. It should be noted that in some examples, the 3D CAD model instances may overlap, creating an occlusion of the object of interest. The occlusions may help train the ML model to detect the object of interest in real images where the object is occluded.

[0089] At 413, random textures 420 may be applied to the 3D CAD models. For example, each instance of the 3D CAD models may have a random texture 420 applied to it. Thus, for each 3D scene, a 3D CAD model is placed in a random position, then *N* instance of that 3D CAD model or other 3D CAD models are made, and a random texture is then applied to each instance.

[0090] At 415, the virtual camera is placed in a random position in 3D space. In some examples, the random placement of the virtual camera may be constrained to be within a maximum distance away from the object of interest. At 417, the virtual camera may be oriented to face the object of interest. In other words, after being randomly placed in the 3D scene, the virtual camera may be made to point at the object of interest. At 419, a synthetic image may be rendered.

[0091] At 421, annotations for the rendered synthetic image may be generated. For example, the object type, bounding box, and segmentation map for the 3D CAD model of the object of interest may be generated.

[0092] At 423, a determination may be made whether all synthetic images were generated for the selected 3D CAD model. For example, a configurable number *K* synthetic images may be made for the selected 3D CAD model. If all of the synthetic images have not been generated (423 determination NO), then another synthetic image for the selected 3D CAD model may be generated starting at 405. If all of the synthetic images have not been generated (423 determination YES), then another 3D scene may be selected at 403 and synthetic images may be generated with the selected 3D scene.

[0093] An objective for the randomization (e.g., position, virtual camera, texture, object duplication, etc.) is to generate synthetic images with as much variation as possible. Therefore, during the training phase, the object detection ML model may learn the object itself without overfitting to very specific features.

[0094] Fig. 5 illustrates a parallel synthetic image generation approach, according to an example. At 501, a main process may create a number of subprocesses to generate synthetic images and annotations for training an ML model. In some examples, the main process may create *N* number of subprocesses. In some examples, the subprocesses may include instances of a 3D rendering engine. In some examples, the number *N* subprocesses may be the number of 3D scenes be used to generate the synthetic images. Each subprocess may use a fixed amount of computer resources (e.g., CPU, GPU, RAM).

[0095] At 503a, a first subprocess may generate synthetic images and annotations for a first 3D scene. At 503b, a second subprocess may generate synthetic images and annotations for a second 3D scene, and so forth through the *N*th subprocess that may generate synthetic images and annotations for the *N*th 3D scene, at 503n.

[0096] At 505, the main process may wait for the subprocesses to each generate a subset of synthetic images and annotations. When the subprocesses generate their synthetic images and annotations, the main process may merge each subset of synthetic images and annotations into a combined training dataset of synthetic images and annotations, at 507.

[0097] Fig. 6 illustrates another parallel synthetic image generation approach, according to an example. At 601, a main process may initialize a number of subprocesses to generate synthetic images and annotations for training an ML model. In some examples, the main process may create *M* number of subprocesses. In some examples, the subprocesses may include instances of a 3D rendering engine. As compared to the approach described in Fig. 5, in this example, the *M* subprocesses may generate a subset of synthetic images and annotations for a given 3D scene. This process may be repeated for *N* number of 3D scenes.

[0098] At 603, the main process may start the synthetic image generation and annotation for a first 3D scene. At 605a, a first subprocess may generate a subset of synthetic images and annotations for the first 3D scene. At 605b, a second subprocess may generate a subset of synthetic images and annotations for the first 3D scene, and so forth through the *M*th subprocess that may generate synthetic images and annotations for the first 3D scene, at 605m. At 607, the main process may merge each subset of synthetic images and annotations for the first 3D scene.

[0099] At 609, the main process may start the synthetic image generation and annotation for a second 3D scene. At 611a, a first subprocess may generate a subset of synthetic images and annotations for the second 3D scene. At 611b, a second subprocess may generate a subset of synthetic images and annotations for the second 3D scene, and so forth through the *M*th subprocess that may generate synthetic images and annotations for the second 3D scene, at 611m. At 613, the main process may merge each subset of synthetic images and annotations for the second 3D scene.

[00100] This procedure may be repeated through the *N*th 3D scene. For example, at 615, the main process may start the synthetic image generation and annotation for the *N*th 3D scene. At 617a, a first subprocess may generate a subset of synthetic images and annotations for the *N*th 3D scene. At 617b, a second subprocess may generate a subset of synthetic images and annotations for the *N*th 3D scene, and so forth through the *M*th subprocess that may generate synthetic images and annotations for the *N*th 3D scene, at 617m. At 619, the main process may merge each subset of synthetic images and annotations for the *N*th 3D scene.

[00101] Fig. 7 depicts a non-transitory machine-readable storage medium 730 for generating synthetic images and annotations, according to an example. To achieve its desired functionality, an electronic device 100 includes various hardware components. Specifically, an electronic device includes a processor and a machine-readable storage medium 730. The machine-readable storage medium 730 is communicatively coupled to the processor. The machine-readable storage medium 730 includes a number of instructions 732, 734, 736, 738 for performing a designated function. The machine-readable storage medium 730 causes the processor to execute the designated function of the instructions 732, 734, 736, 738. The machine-readable storage medium 730 can store data, programs, instructions, or any other machine-readable data that can be utilized to operate the electronic device 100. Machine-readable storage medium 730 can store computer readable instructions that the processor of the electronic device 100 can process or execute. The machine-readable storage medium 730 can be an electronic, magnetic, optical, or other physical storage device that contains or stores executable instructions. Machine-readable storage medium 730 may be, for example, Random Access Memory (RAM), an Electrically Erasable Programmable Read-Only Memory (EEPROM), a storage device, an optical disc, etc. The machine-readable storage medium 730 may be a non-transitory machine-readable storage medium 730, where the term “non-transitory” does not encompass transitory propagating signals.

[00102] Referring to Fig. 7, receive instructions 732, when executed by the processor, may cause the processor to receive a 3D CAD model of an object and multiple 3D scenes. Generate synthetic images instructions 734, when executed by the processor, may cause the processor to generate multiple synthetic images of the object from the 3D CAD model and the multiple 3D scenes based on randomized visual parameters. Generate annotations instructions 736, when executed by the processor, may cause the processor to generate annotations for the multiple synthetic images using the 3D CAD model. Training instructions 738 when executed by the processor, may cause the processor to train an ML model for detecting an object in an image captured by a camera based on the multiple synthetic images and annotations.

CLAIMS

What is claimed is:

1. An electronic device, comprising:
a processor; and
a memory communicatively coupled to the processor and storing executable instructions that when executed cause the processor to:
receive a three-dimensional (3D) computer aided design (CAD) model of an object;
generate multiple synthetic images of the object from the 3D CAD model based on a 3D scene and randomized visual parameters; and
generate annotations for the multiple synthetic images using the 3D CAD model.

2. The electronic device of claim 1, wherein the 3D scene comprises a 3D model of an environment.

3. The electronic device of claim 1, wherein the instructions to generate the multiple synthetic images comprise executable instructions that when executed cause the processor to:
randomly select different locations for placing the object in the 3D scene; and
randomly select distances and positions of a virtual camera with respect to the object.

4. The electronic device of claim 1, wherein the instructions to generate the multiple synthetic images comprise executable instructions that when executed cause the processor to:
randomize textures for the object in the 3D scene.

5. The electronic device of claim 1, wherein the instructions to generate the multiple synthetic images comprise executable instructions that when executed cause the processor to:

randomize dimensions of the object in the synthetic images, the dimensions of the object being constrained by a scale of the object within the 3D scene.

6. The electronic device of claim 1, wherein the instructions to generate the multiple synthetic images comprise executable instructions that when executed cause the processor to:

randomize lighting of the 3D scene containing the object for the multiple synthetic images.

7. A non-transitory computer-readable storage medium comprising instructions executable by a processor to:

receive a three-dimensional (3D) computer aided design (CAD) model of an object and multiple 3D scenes;

generate multiple synthetic images of the object from the 3D CAD model and the multiple 3D scenes based on randomized visual parameters;

generate annotations for the multiple synthetic images using the 3D CAD model; and

train a machine-learning (ML) model for detecting an object in an image captured by a camera based on the multiple synthetic images and annotations.

8. The non-transitory computer-readable storage medium of claim 7, wherein the instructions to generate the multiple synthetic images comprise instructions executable by the processor to:

place the 3D CAD model in randomized locations in the multiple 3D scenes.

9. The non-transitory computer-readable storage medium of claim 7, wherein the instructions to generate the multiple synthetic images comprise instructions executable by the processor to:

generate an object grid comprising multiple instances of the object placed in a given 3D scene; and

render a synthetic image to capture the object grid in the given 3D scene.

10. The non-transitory computer-readable storage medium of claim 9, wherein the object grid is to simulate multiple instances of the object being placed in close proximity.

11. The non-transitory computer-readable storage medium of claim 7, wherein the instructions to generate the annotations for the multiple synthetic images comprise instructions executable by the processor to:

determine a bounding box of the object using the 3D CAD model for each of the multiple synthetic images;

save an object type for each of the multiple synthetic images; and

generate a segmentation mask of the object in each of the multiple synthetic images.

12. The non-transitory computer-readable storage medium of claim 8, wherein the instructions further comprise instructions executable by the processor to:

start multiple subprocesses to generate subsets of the multiple synthetic images and annotations; and

merge the subsets of the multiple synthetic images and annotations generated by the multiple subprocesses into a combined dataset.

13. A method comprising:

generating multiple synthetic images of an object based on a three-dimensional (3D) computer aided design (CAD) model of the object, a 3D scene and randomized visual parameters;

generating annotations for the multiple synthetic images using the 3D CAD model;

training a machine-learning (ML) model to detect the object in the multiple synthetic images based on the annotations; and

running the ML model to detect an observed object in an image captured by a camera.

14. The method of claim 13, wherein the 3D CAD model comprises an object that is to be produced in an additive manufacturing process, wherein the ML model is trained to detect objects produced by the additive manufacturing process.

15. The method of claim 13, wherein the 3D CAD model comprises an object that is representative of a type of object offered in a retail environment.

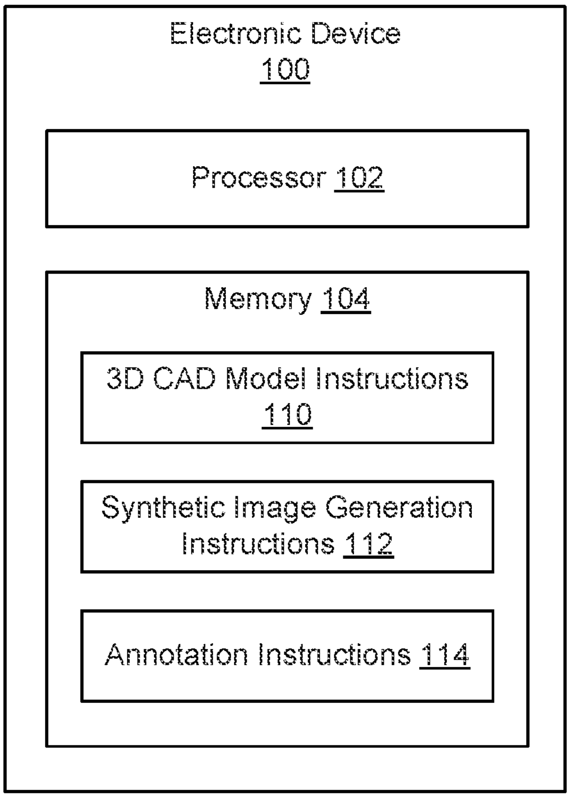
Electronic Device
100
Processor 102
Memory 104
3D CAD Model Instructions
110
Synthetic Image Generation
Instructions 112
Annotation Instructions 114

Fig. 1

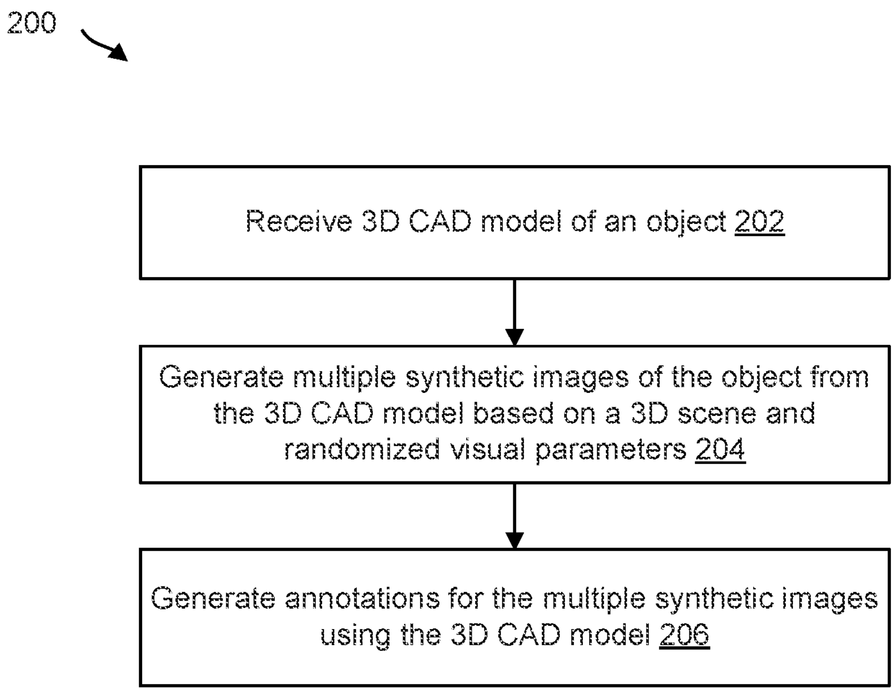
200
Receive 3D CAD model of an object 202
Generate multiple synthetic images of the object from the 3D CAD model based on a 3D scene and randomized visual parameters 204
Generate annotations for the multiple synthetic images using the 3D CAD model 206

Fig. 2

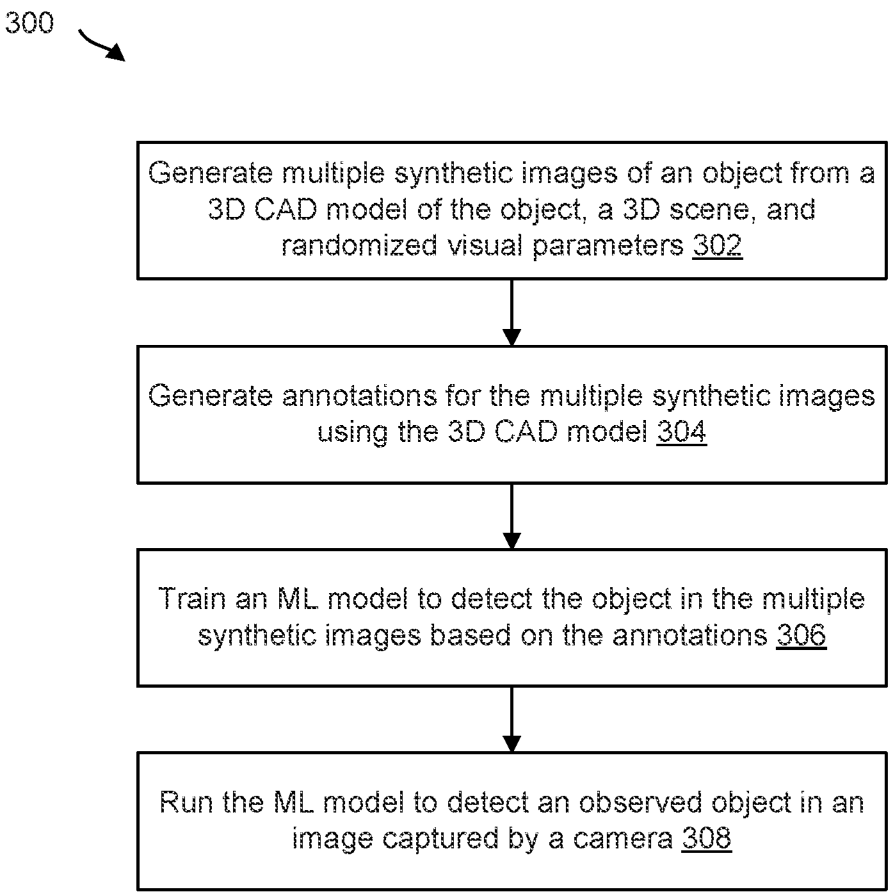
300
Generate multiple synthetic images of an object from a 3D CAD model of the object, a 3D scene, and randomized visual parameters 302
Generate annotations for the multiple synthetic images using the 3D CAD model 304
Train an ML model to detect the object in the multiple synthetic images based on the annotations 306
Run the ML model to detect an observed object in an image captured by a camera 308

Fig. 3

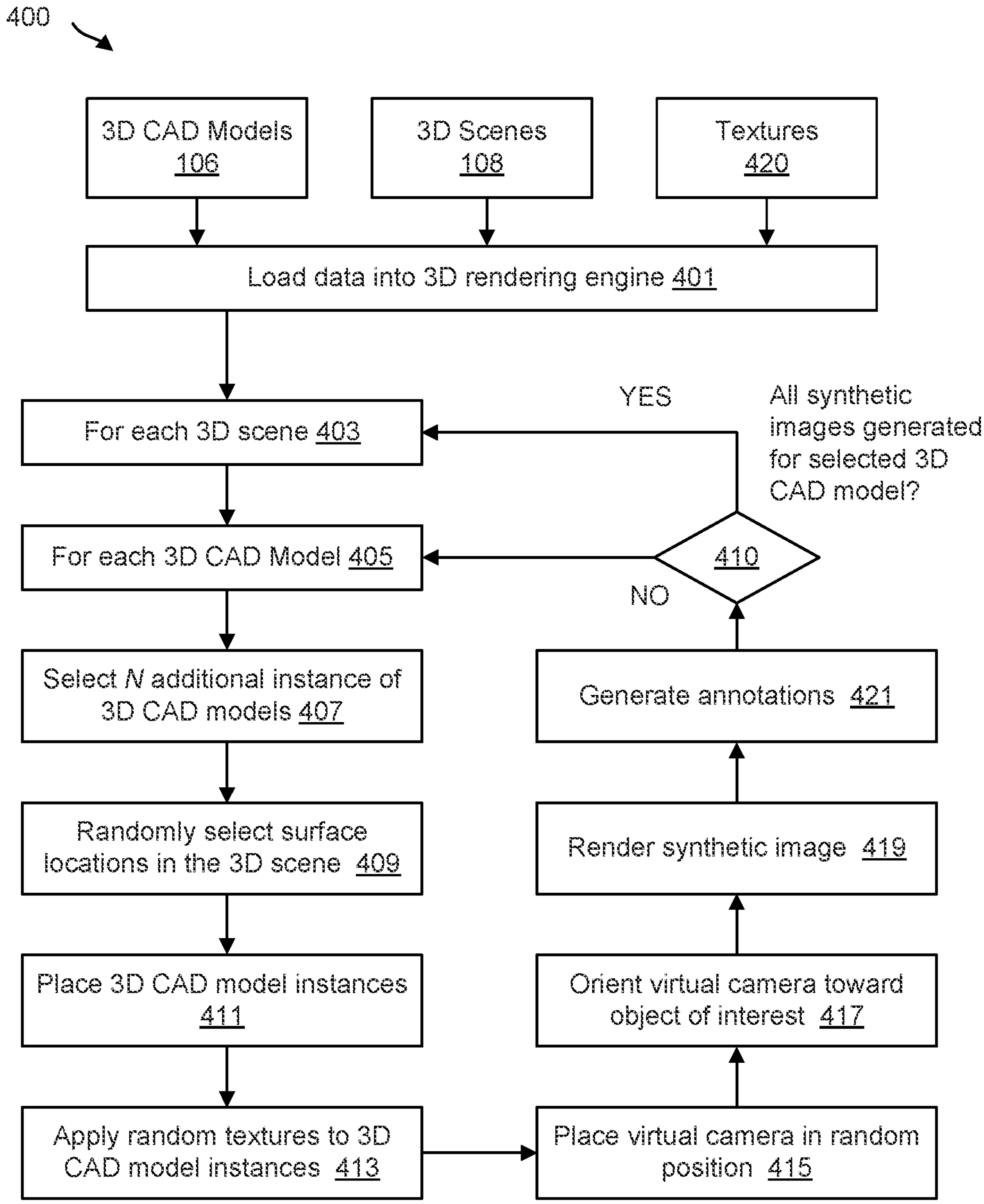
400
3D CAD Models 106
3D Scenes 108
Textures 420
Load data into 3D rendering engine 401
For each 3D scene 403
YES
All synthetic images generated for selected 3D CAD model?
For each 3D CAD Model 405
410
NO
Select N additional instance of 3D CAD models 407
Generate annotations 421
Randomly select surface locations in the 3D scene 409
Render synthetic image 419
Place 3D CAD model instances 411
Orient virtual camera toward object of interest 417
Apply random textures to 3D CAD model instances 413
Place virtual camera in random position 415

Fig. 4

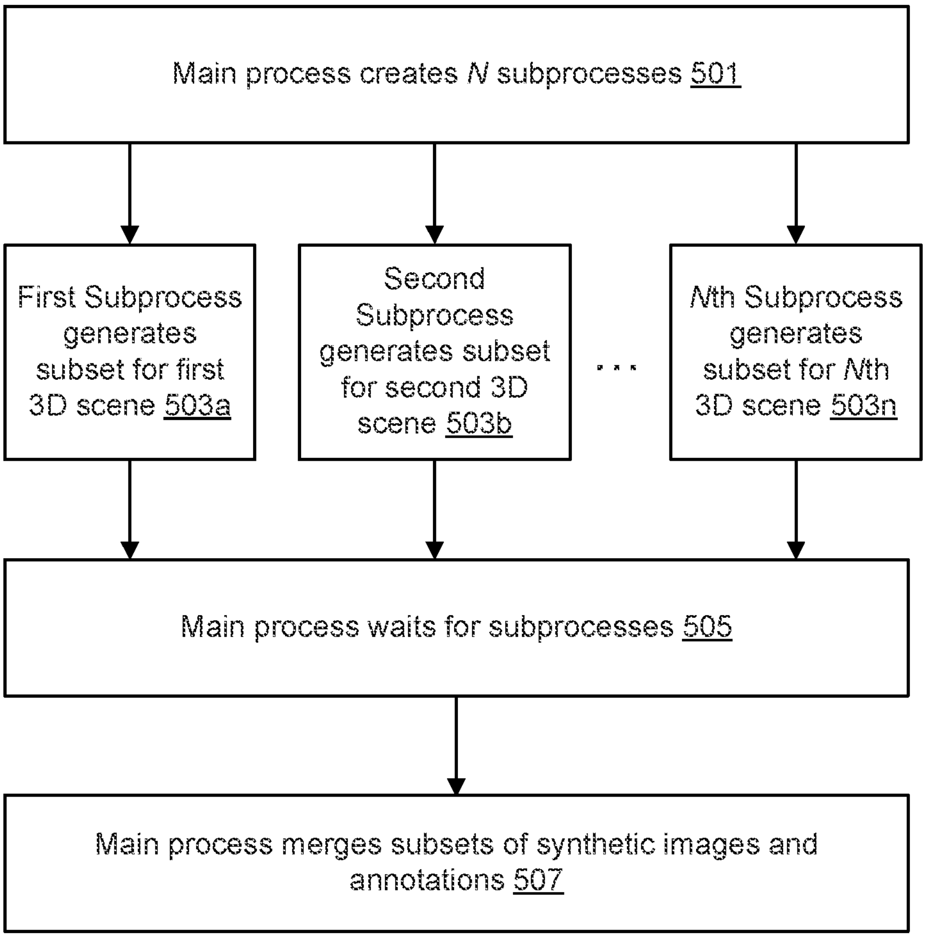
Main process creates N subprocesses 501
First Subprocess generates subset for first 3D scene 503a
Second Subprocess generates subset for second 3D scene 503b
...
Nth Subprocess generates subset for Nth 3D scene 503n
Main process waits for subprocesses 505
Main process merges subsets of synthetic images and annotations 507

Fig. 5

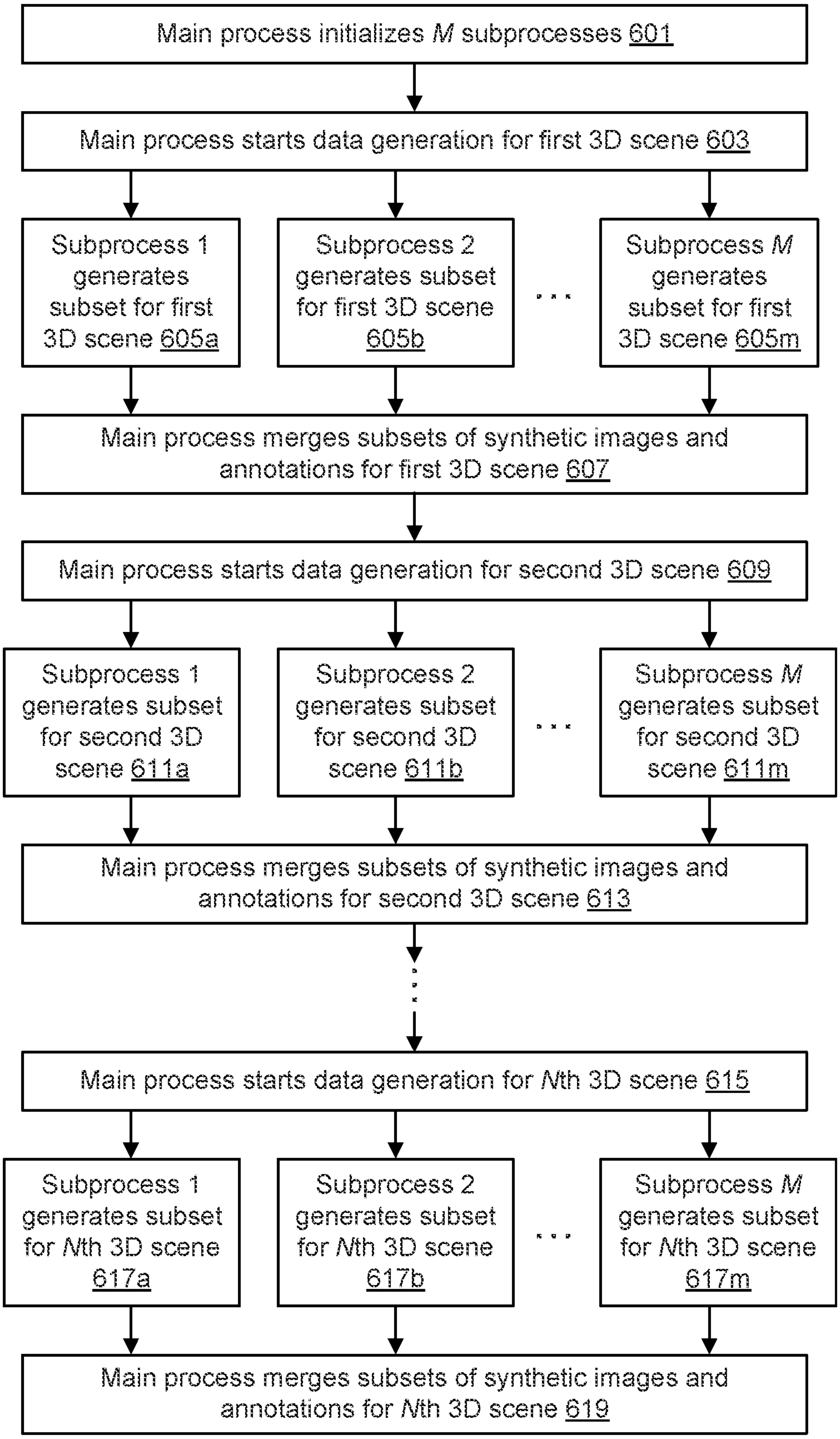
Main process initializes M subprocesses 601
Main process starts data generation for first 3D scene 603
Subprocess 1 generates subset for first 3D scene 605a
Subprocess 2 generates subset for first 3D scene 605b
Subprocess M generates subset for first 3D scene 605m
Main process merges subsets of synthetic images and annotations for first 3D scene 607
Main process starts data generation for second 3D scene 609
Subprocess 1 generates subset for second 3D scene 611a
Subprocess 2 generates subset for second 3D scene 611b
Subprocess M generates subset for second 3D scene 611m
Main process merges subsets of synthetic images and annotations for second 3D scene 613
Main process starts data generation for Nth 3D scene 615
Subprocess 1 generates subset for Nth 3D scene 617a
Subprocess 2 generates subset for Nth 3D scene 617b
Subprocess M generates subset for Nth 3D scene 617m
Main process merges subsets of synthetic images and annotations for Nth 3D scene 619

Fig. 6

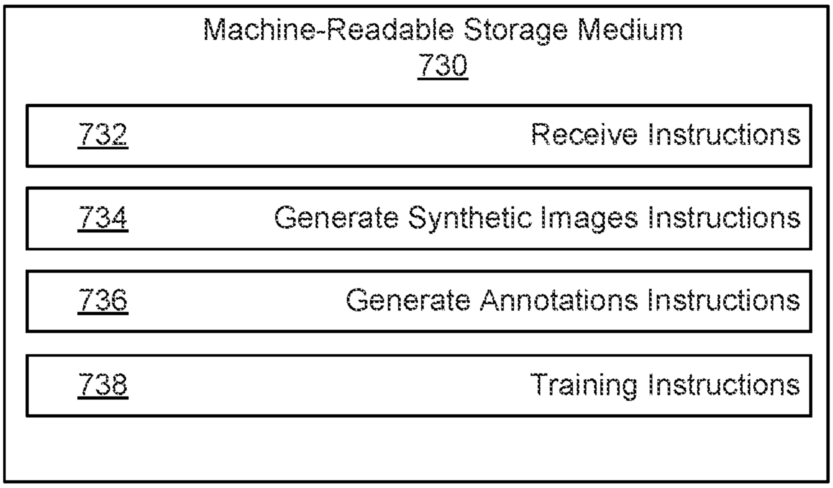
Machine-Readable Storage Medium
730
732
Receive Instructions
734
Generate Synthetic Images Instructions
736
Generate Annotations Instructions
738
Training Instructions

Fig. 7

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US 2021/039871

A. CLASSIFICATION OF SUBJECT MATTER

G06V 10/22 *(2022.01)*
G06T 15/08 *(2011.01)*
G06T 19/20 *(2011.01)*

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06V 10/00-10/22, G06T 15/00 -15/08, 19/00-19/20, 7/521

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

PatSearch (RUPTO internal), USPTO, PAJ, K-PION, Esp@cenet, Information Retrieval System of FIPS

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO 2018/156126 A1 (SIEMENS AKTIENGESELLSCAFT) 30.08.2018	1-15
A	US 2020/0167161 A1 (SIEMENS AKTIENGESELLSCAFT) 28.05.2020	1-15
A	WO 2018/080533 A1 (SIEMENS AKTIENGESELLSCAFT) 03.05.2018	1-15
A	WO 2018/009405 A1 (AVENT, INC.) 11.01.2018	1-15

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"D" document cited by the applicant in the international application

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
02 March 2022 (02.03.2022)	10 March 2022 (10.03.2022)

Name and mailing address of the ISA/RU:	Authorized officer
Federal Institute of Industrial Property, Berezhkovskaya nab., 30-1, Moscow, G-59, GSP-3, Russia, 125993 Facsimile No: (8-495) 531-63-18, (8-499) 243-33-37	O. Kachan Telephone No. 499-240-60-15