

SYSTEM AND METHOD FOR VISUAL INSPECTION

FIELD

5 The present invention relates to a system and method for visual inspection. The present invention further relates to a system and a method applicable for visual inspection required in, for example, housekeeping monitoring, merchandise display/presentation, and more.

BACKGROUND

10

Many advanced countries are facing a shortage in productive manpower, particularly in frontline and supervisory roles in many sectors, due to many factors. Aging population and better education are some of the factors for the reduction in these ranks. Furthermore, the workers in these roles face resource and time constraints on a daily basis. Hence, the job positions of these roles typically have a high labour turnover rate, which means that businesses have to incur higher training costs to train new workers replacing those that have left.

15

20

Specific to the hospitality sector, such as in a hotel, housekeeping work is largely manual and laborious. Manpower crunch has increasingly made it costly to upkeep housekeeping standards. Hotels also rely on supervisors to manually supervise housekeeping work, which involves laborious visual inspection. This poses further challenges on costs, required manpower and time costs spent by a supervisor to check housekeeping work and housekeeper's time to rework on issues identified by the supervisor. If housekeeping standards are not properly maintained, the hotels are likely to face guest complaints and bad ratings which would severely affect their image, reputation and revenue. Due to high manpower turnover faced by hotels, significant time is also incurred on training to maintain the hotel's housekeeping standards. Also, standards enforced by housekeeping supervisors may also vary and worsen with turnover in manpower. If the standards are not met, on top of more rework cleaning, housekeepers also have to attend to more ad-hoc housekeeping requests by hotel guests and these add to the hotel's challenges in a labour constraint industry.

25

30

Furthermore, in the retail sector, merchandise displays are an important part of a company's marketing strategy. It can be quite laborious for retail staff to visually inspect every retail location to ensure that merchandise is displayed according to the required standards.

35

40

Visual inspection tools can be used to assist with the visual inspection work described above. For instance, feature extraction algorithms such as SIFT and many of its derivatives have revolutionized object detection. However, existing visual inspection systems involving such object detection method is typically very specific to an original object's features, require much computing resources, and can have accuracy issues.

SUMMARY

According to examples of the present disclosure, there are provided a system and a method as claimed in the independent claims. Some optional features are defined in the dependent claims.

45

BRIEF DESCRIPTION OF THE DRAWINGS

Examples of the present disclosure are illustrated by the following figures.

50

FIG. 1 is a flowchart illustrating a process flow on a mobile device issued to a housekeeper or cleaner. The device is a component of an example of the present disclosure.

FIG. 2 is a flowchart illustrating a process flow for a supervisor to conduct housekeeping supervision remotely.

55

FIG. 3 is a flowchart illustrating a process flow for a supervisor to conduct housekeeping supervision on-site.

FIG. 4 to 6 is a flowchart illustrating an overall architecture of an example of the present disclosure, in particular involving a mobile application of a mobile device, a backend system and a backend algorithm. Specifically, FIG. 5 is a continuation of the flowchart of FIG.4 and FIG. 6 is a continuation of the flowchart of FIG. 5.

FIG. 7 is an overall architecture of an example of the present disclosure.

FIG. 8 shows two images captured by a mobile device, each image illustrating outlining of objects to guide a housekeeper to take photographs or videos.

FIG. 9 shows a housekeeping performance record of an example of the present disclosure.

5 FIG. 10 shows a yearly staff performance report of an example of the present disclosure.

FIG. 11A shows a room item review report of an example of the present disclosure.

FIG. 11B shows a room item detail review report of an example of the present disclosure.

FIG. 12 shows a schematic diagram of a training server for machine learning.

FIG. 13 shows a flowchart illustrating an image pre-processing workflow.

10 FIG. 14 shows a flowchart illustrating an image processing workflow set to take place after the pre-processing workflow of FIG. 13.

FIG. 15 shows values of parameters of pixels of an image for determining presence of an outline of an object in the image.

15 FIG. 16 shows a schematic diagram illustrating feeding of input reference images to train a neural network to obtain an output trained model of the neural network.

FIG. 17 shows a classification module according to an example of the present disclosure.

FIG. 18 illustrates matched and aligned feature points and outliers between two images that are compared.

FIG. 19 is an overview of a system according to another example of the present disclosure.

20 FIG. 20 shows object relationships in a scene which can be used for evaluation.

FIG. 21 illustrates object overlay used to help a user align objects to a reference viewpoint.

FIG. 22 shows how objects that are moved or missing can be tracked.

FIG. 23 shows how moved or missing objects can be highlighted and communicated to a user.

FIG. 24 shows image segmentation of a bottle out of a scene and checking for portions of the bottle.

25 FIG. 25 illustrates how objects of same type may be counted.

FIG. 26 illustrates how data for scene description may be structured.

FIG. 27 is a diagram showing an example of a scene with three objects.

FIG. 28 shows an example of how scene data may be represented.

FIG. 29 shows an example of how object data may be represented.

30 FIG. 30 contains flow charts illustrating processing of a reference scene.

FIG. 31 is a flow chart for a scene evaluation process.

FIG. 32 is a flow chart showing process details for scene evaluation.

FIG. 33 shows a list of scenes and objects possibly seen in a hotel room.

FIG. 34 is a flow chart relating to a process for scene evaluation in a housekeeping process.

35 FIG. 35 is a flow chart relating to a process for refrigerator item counting.

FIG. 36 shows an implementation of an example of a system of the present disclosure to produce scene description.

FIG. 37 illustrates bounding boxes and masks for detected objects in a scene.

40 DETAILED DESCRIPTION

Examples of the present disclosure relates to the field of machine vision that may involve machine learning, evaluation of a scene based on detection of objects and determination of the intrinsic properties of objects, and the properties of the objects relative to each other. Some methods, systems and/or apparatuses described below are applicable for housekeeping monitoring, merchandise display/presentation inspection, and more.

For example, there can be provided a method and a system for decomposing a scene to object properties defined as a scene description. The scene description can later be used to evaluate a similar scene to determine translational movement and changes in objects. The scene description is a compact representation of the scene by retaining details of objects of interest and their relationships with each other. A more compact representation allows (1) lower storage requirements, (2) evaluation and comparison of scenes to be performed faster and with lower computing resources, and (3) feedback to be provided to a user of the changes in the scene by indicating discrepancies in the evaluated scene.

55 Firstly, a reference image or video of a scene of interest is captured using a camera by a user. This scene is processed by the system to yield a scene description, which is a compact representation of the objects of interest in the scene and their relationships with each other. This scene description may then be used to compare and evaluate against subsequently captured images or videos of the same or

similar scene with the same viewpoint by a camera that is hand-operable by the user. This scene description is a compact representation of the scene and thus saves storage cost and lowers computing cost for image comparison. From the scene description, details which include position, size, colour and other properties may be extracted, evaluated and compared.

To produce a scene description, an object detection and segmentation algorithm is used. In one example, a trained neural network model may be used to perform object detection and segmentation. In the present disclosure, the terms “neural network”, “machine learning”, and “deep learning” are used interchangeably. An alternative method would be to use feature extraction using machine vision algorithms such as SIFT to detect and segment the objects.

Once objects are detected, segmented and identified, they can be described using properties such as position, count, size, colour and their relationships with each other. The segmented images of the objects are also saved and used to perform image comparison.

The aforementioned method and system can be applied to a first application to determine orderliness in a hotel housekeeping scenario. In this case, reference images of various parts of the hotel room are captured. The various parts of the room may include, for instance, bedroom, bathroom, wardrobe, and so on. Taking the bedroom as an example, the reference image may contain a bed, cabinets, floor mats, bottles on cabinets, flippers on floor mats, and so on. A housekeeper, who is responsible for tidying and cleaning hotel rooms daily, may be required by the hotel to maintain standards such as cleanliness, positioning of various objects, and replenishing supplies such as drinks and toiletries. After tidying up, the housekeeper may then capture an image of the room for evaluation. This image is uploaded to a server to be evaluated. A pass or fail status will be transmitted back to the housekeeper. A fail status includes markings showing the locations causing the failure. The housekeeper may then have the option to rework on those locations. This automates most of the housekeeping inspection, and leaves the supervisor to deal with other more pressing issues.

A second application for the method and system is in the counting of objects in various settings, which is useful for sales and accounting purposes, and for checking object positioning. As an example, the method and system may be used for checking stock level for drinks and snacks in a hotel refrigerator, and charge customers for items consumed. In a hotel refrigerator, drinks and snacks of different types are of a specific quantity and are typically placed in a specific order. The method and system perform a dual function including counting of items as well as check their positioning in the refrigerator.

Likewise, the method and system may be applied for counting objects for merchandise display and for checking that object positioning is in order. The method and system can also be applied in a warehousing system for counting objects being stored in an area (e.g. in a warehouse or storage place) and transported retrieved to an area (e.g. in a delivery vehicle). Specifically, objects delivered to each venue in the supply chain can be accounted using the method and system. Missing objects or Objects detected with changes can be automatically alerted by the warehousing system to users.

FIG. 19 shows an overview of the key components of an example of the aforementioned system 1950. Scene 1900 is captured and pre-processed by a mobile device 1910 (e.g. smartphone) which is equipped with a camera. The scene data captured may be in the form of 2D images, 3D images, or videos. Pre-processing of the scene data may include the use of digital image processing algorithms such as image sharpening, white balance, equalization and compression.

The scene data is then transmitted to a server 1920. The transmission is made through a network 1940. Scene data are processed by a computing service 1922 to produce a scene description, which is stored together with the scene data into a storage device 1924. Scene data and description are accessed from a computer 1930, which allows a user to review and edit the scene data and descriptions. The data transfer between the server 1920 and computer 1930 are made through the network 1940.

The network 1940 may be a local area network or the wider Internet, or a combination. The medium of transmission for the network 1940 may be wired or wireless, or a combination of the two. The server 1920 thus may reside in the local area network or in a data centre or typically described as being in the “cloud”.

The computing service 1922 may reside in the server 1920 itself or a virtualized instance in a cloud service platform. The computing service 1922 is used to process the scene data to produce scene descriptions and perform scene evaluations. Scene evaluations include comparisons and counting of objects in the scene. Storage 1924 stores the scene data and its corresponding scene description. The storage 1924 may in the form of files or as data in a database.

A computer 1930 is used to review the scene data, edit the resulting description and evaluation parameters. Modified description and parameters are updated through the server 1920, through the network 1940.

The scene description may contain the following:

- Scene properties such as brightness, white balance, and other intrinsic features, which are calculated from image processing algorithms, and aid in scene evaluation, and
- Object properties such as size, colour, position, bounding box, segmented image and other relevant processed parameters.

The scene description is a structured collection of the above properties and may be stored in structured text format such as JSON (JavaScript Object Notation), XML (Extensible Markup Language), in binary format such as Protobuf (Protocol Buffers) or records in a database.

FIG. 20 illustrates creation of a reference scene using the system of FIG. 19. Specifically, FIG. 20 shows an example of how object relationships may be described in a scene. In this scenario, an objective is to determine whether there is any change in the positions of the objects relative to each other. FIG. 20 contains a scene 2000, which is captured by the mobile device 1910 of FIG. 19. Objects 2010, 2012, 2014 are found in the captured scene using an object detection, identification and segmentation (ODIS) method. This method may be performed using a trained neural network model or specialized machine vision algorithms. Although only three objects are shown, the scenario applies to any number of objects in the scene 2000. In one example, the objects may overlap with each other and cause occlusion of some objects. In this case, the ODIS method will segment out the appropriate visible portions for storage and evaluation, and a segmented image will be created.

A possible implementation of the ODIS method is shown in FIG. 36. This implementation comprises of two modules, namely a Mask RCNN neural network 3602 and a Post-Processor 3610. Mask RCNN is essentially a convolutional neural network (CNN) which is structured and trained to categorize objects in an image and produce masks or regions, which segments out the categorized objects. The neural network 3602 takes an image 3600 as an input and outputs three pieces of information for each of the objects that are found, namely, category 3604, bounding box 3606 and mask 3608. These three pieces of information are then fed to a Post-Processor 3610 to produce a scene description 3612. The conversion process of the Post-Processor is a combination of heuristics and logic operations, and will be explained in later paragraphs.

The weights of the neural network 3602 are specifically trained using a set of curated and annotated images. This set of images is chosen according to the application area in which the neural network 3602 will be used. For example, for use in the area of housekeeping of hotel rooms, images of hotel rooms will be used to train the neural network. The annotation of the images requires relevant objects to be categorized and their outlines to be drawn out. The images and their corresponding annotation data are then presented to the neural network 3602 for training.

The outputs 3604, 3606 and 3608 of the neural network 3602 in FIG. 36 are illustrated in FIG. 37. When fed an input image from the scene 2000, the neural network 3602 produces boundary boxes 3700, 3702, 3704 for the objects 2014, 2010 and 2012 respectively. Each of these boundary boxes 3700, 3702 and 3704 is a rectangle forming a tight fit of the outline of each object 2014, 2010 and 2012 respectively. In addition, the neural network 3602 produces masks 3706, 3708 and 3710 for the objects 2014, 2010 and 2012 respectively. Each of these masks 3706, 3708 and 3710 is a region, which fills the outline of each object 2014, 2010 and 2012 respectively, and each of the masks 3706, 3708 and 3710 represents the segmentation of each object 2014, 2010 and 2012 respectively from the input image. Therefore, in the manner as described with reference to FIG. 36 and 37, the present example may use Artificial Intelligence (A.I.) in the form of a neural network to generate one or more reference images for image comparison later.

In the present example, the top left hand corner of the scene 2000 is considered as the origin, O . A vector 2026 taking reference from the origin O and pointing at the object 2012 thus represents the position of the object 2012. Options for calculating the position of the object 2012 in the scene 2000 include calculating a centre of a bounding box of the object 2012, calculating a centroid of the segmented image (segmented via ODIS method), or any other method that is suitable. Taking a central position of object 2012 as a reference position (e.g. the calculated centre or centroid), vectors 2020 and 2022 taking reference from the reference position and pointing at the objects 2010 and 2014, can be used to position objects 2010 and 2014 respectively. In FIG. 20, the object 2012 is defined as an anchor object. The anchor object is one from which other objects take their relative positions. Typically, a relatively immovable object should be chosen, e.g. a table on which books and a lamp are sitting. An additional vector 2024 linking central reference positions of objects 2010 and 2014 respectively is used to describe relative positions between object 2010 and 2014. Vectors 2020, 2022, 2024 and 2026 thus form a positioning information part of a scene description for the scene 2000. If scene 2000 is taken as a reference scene, the same vectors, which form part of the scene description can be saved into the storage medium 1924 of FIG. 19 for later use. Subsequent scene descriptions from captured scenes similar to 2000 may then be compared with this reference scene description as part of a scene evaluation process. The objects contained in this reference scene are known as reference objects.

If the scene 2000 is a 2D image or a 2D frame from a video, all position vectors will be in two dimensions. However, if the scene 2000 is a 3D image, the position vectors will be in 3-dimensions and the description for FIG. 20 and subsequent figures further illustrating the method may be treated accordingly.

FIG. 21 illustrates features in the form of graphical guidance to facilitate hand-operation of a user of the mobile device 1910 of FIG. 19 to capture images for scene evaluation. Such graphical guidance operates based on features of one or more reference images. In order for scene comparison between a captured image and the reference scene 2000 of FIG. 20 to be meaningful, captured scenes to be subjected to the scene evaluation process should be captured from approximately the same viewpoint as the reference scene 2000. A first outlining feature is proposed to assist a user operating the mobile device 1910 to capture images to achieve this. For example, an overlay of outlines of the reference objects in the reference scene is provided on a display of the mobile device 1910. An example of the overlay is shown in FIG. 21. In a new scene 2150 captured by the camera of the mobile device 1910, there is shown an outline 2160 of the reference object 2010 in FIG. 20, an outline 2162 of the reference object 2012 in FIG. 20 and an outline 2164 of the reference object 2014 in FIG. 20. The outlines 2160, 2162, 2164 of the objects may be obtained from edge detection algorithms such as Canny and Sobel methods. An advantage of having an outline overlay is that the reference scene is made visible for the user to aim and adjust the camera on the mobile device 1910 to capture scenes that are approximately the same viewpoint as the reference scene.

A second feature involving mobile device orientation indicators is proposed for a user to capture good images for scene evaluation. Such mobile device orientation indicators operate based on features of one or more reference images. For example, the angular positions of the mobile device are stored during the capture of the reference scene 2000 of FIG. 20, and these angular positions are used to guide the user to move the mobile device 1910 to the same angular positions to capture good images. The angular positions can be detected using gyro-sensors and acceleration sensors commonly present in mobile devices such as smartphones. A guide is displayed in a Graphical User Interface (GUI) of the mobile device 1910 for guiding a user to adjust the mobile device 1910 to the desired angular positions for capturing images (i.e. graphical guidance). With reference to FIG. 21, this guide can be provided in a form of a bubble level indicator 2170. This bubble level indicator 2170 can work in a similar manner as a known mechanical device with the same name that is used for levelling of horizontal objects such as beams, tables or platforms. In the present example, the bubble level indicator 2170 comprises a bubble or sphere 2172 and a crosshair or reference target 2174 overlaid on the scene 2150 or appear as part of the GUI. The objective of the bubble level indicator 2170 is to guide the user to tilt the mobile device 1910 until the bubble 2172 is centred on the target 2174. The further away the bubble 2172 is from the target 2174, the larger the angle of tilt is required for the user to orientate the mobile device 1910 to the desired orientation to capture images.

FIG. 22 illustrates comparison between a captured image or scene 2200 and the reference scene 2000 of FIG. 20 during scene evaluation. Specifically, FIG. 22 shows an instance of a captured scene 2200, wherein some objects have moved or have gone missing compared to the reference scene 2000 of

FIG. 20. Object 2214 represents the object 2014 in FIG. 20, which has moved. Line 2222 represents a new vector from the object 2012 in FIG. 20 to the moved object 2214 that is calculated. The new vector 2222 is different from the original vector 2022 from the object 2012 to the unmoved object 2014. The difference between the new vector 2222 and the original vector 2022 is shown as vector 2230. A comparison algorithm can be used to conclude that the object 2014 has indeed moved if a value pertaining to this vector 2230 exceeds a predetermined threshold after comparison. Such movement can be flagged as a discrepancy. When applied in different applications, such discrepancy can be indicative of, for example, poor housekeeping or inaccurate arrangement for merchandise display, as the objects are not positioned in a preferred order or manner.

The object 2010 in FIG. 20 is missing in the captured scene 2200. Since the object 2010 has gone missing, it cannot be detected in the captured scene 2200 through, for instance, the ODIS method. Furthermore, vectors 2020 and 2024 in FIG. 20 and new vectors indicative of movement of the object 2010 cannot be calculated. Thus, the object 2010 can be easily evaluated to be missing in the captured scene 2200. A missing object can be flagged as a discrepancy and can also be indicative of, for example, poor housekeeping, incorrect arrangement for merchandise display, or stolen objects.

Object 2216 is an extra object not found in the reference scene 2000. Object 2216 can be found in the captured scene 2200 by using, for instance, the ODIS method. The object 2216 can be flagged as a discrepancy and recorded as a new object not found in the reference scene 2000. Similarly, new objects can be indicative of, for example, poor housekeeping or incorrect arrangement for merchandise display.

FIG. 23 illustrates how the detected discrepancies described above can be communicated to a user using overlays displayed in the captured scene 2200 of FIG. 22 that are shown on the display of the mobile device 1910 of FIG. 19. The scene 2200 shown in FIG. 22 can be displayed on the display or screen of the mobile device 1910 or the computer 1930. Outline 2310 is a highlighted outline of the reference object 2010 in FIG. 20 that is overlaid at the original position of the reference object 2010 in the reference scene 2000 in FIG. 20. The outline 2310 appears to indicate that the object 210 is missing from the scene. Outline 2314 is a highlighted outline of the reference object 2014 in FIG. 20 that is overlaid at the original position of the reference object 2014 in the reference scene 2000. The highlighted outline 2314 allows the user to see that object 2014 has moved to a position indicated by the object 2214 of FIG. 22. The object 2316 is a highlighted outline of the extra object 2216 found in the captured scene 2200 in FIG. 22. In this manner, the user is alerted of all the discrepancies detected in the captured scene 2200. The highlighted outlines may be any colour and a recommendation is to use bold colour like red for the outlines, and the space bounded by the outlines should be transparent to provide good visibility for the discrepancies.

Besides evaluation of objects' presence and position at a scene level, it would also be useful to evaluate each object at a more detailed object/sub-object image level. For example, a segmented image of an object may yield processed parameters such as a colour histogram, area, as well as position. These parameters may be used for further evaluation to determine more precise matching of objects in a captured scene and a reference scene as well as other relevant properties of these objects. These same parameters may be stored in the scene description or calculated as and when required.

In FIG. 24, an object 2400 is used to illustrate how an object may be decomposed into sub-objects of interest to facilitate detailed comparison. The object 2400 is a bottle in this example, but the same concept may be extended to any other object. The bottle 2400 can be an object extracted from a reference scene like the reference scene 2000 in FIG. 20. After it is detected, identified and segmented by, for instance, an ODIS algorithm from the reference scene at a step 2410, a reference segmented image 2402 representative of the object 2400 is obtained. This reference segmented image 2402 may be compared with a subsequently captured and segmented image of a detected object in a captured scene to determine whether there are any differences between the segmented image 2402 and the segmented image of the detected object. For example, a segmented image of the bottle's body may be used to determine whether the bottle is filled or empty, or even detect the colour of the liquid contained in the bottle. With visual inspection performed to such detail, the system is able to detect whether a bottle that is present in a location in a captured scene is a right bottle that should be present at the location in the captured scene.

Furthermore, in this bottle example, there can be other sub-objects that are of interest, namely bottle cap 2404 and label 2406. The presence of the bottle cap 2404 is useful to determine whether the bottle

2400 has been opened. The label 2406 may be used to determine the brand/type of beverage contained in the bottle 2400. The sub-objects 2404 and 2406 may be further segmented and data of the segmented parts form parts of the object description for the bottle 2400. In FIG. 24, the segmentation processes to produce segmented images of sub-objects 2404 and 2406 are represented by arrows 2412 and 2414 respectively. The segmentation processes may be performed manually by a user or automatically by an image processing or ODIS algorithm. Once obtained, the segmented images 2402, 2404 and 2406 can provide and constitute data for object description of the bottle 2400, and can be stored as part of a scene description for a reference scene. The stored data can be used to evaluate a bottle in a captured scene using the same techniques described with reference to FIG. 20 to FIG. 23.

Evaluation of objects at the object/sub-object image level allows the system to perform object accounting more efficiently and accurately. This is illustrated by an example with reference to FIG. 25. FIG. 25 illustrates a scene 2550 comprising of 2 different types of objects. The ODIS method can be used to detect that objects 2560, 2562 and 2564 belong to a "Box" category, and 2570, 2572 and 2574 belong to a "Cylinder" category. The counting of objects at such category level can be done efficiently. Comparison of images at the object/sub-object image level can be performed to distinguish between objects of different categories, for instance, to tell that Box 2564 is different from the objects 2560 and 2562. Image comparison methods, for instance, template matching and SIFT, can be used for the comparison between images of the objects. With such image comparison implemented, the system will be able to distinguish that there are 2 boxes (2560, 2562) of a type 1 with 5-point stars, 1 box (2564) of a type 2 with 4-point star, and 3 cylinders with each having a wing image.

An example of a data structure for scene description (scene description layout) is illustrated in FIG. 26. FIG. 26 shows a multiple layer data hierarchy with each lower layer containing increasing details about each scene and the objects contained in each scene. There is provided a scene layer 2650 comprising data blocks associated with a plurality of scenes that are captured. Each scene comprises one or more object data blocks, *Object1* to *ObjectN*, in a first object layer 2652. The number of object blocks, *N*, corresponds to number of objects detected in each scene. Each object in layer 2652 may comprise one or more sub-objects in a second object layer 2654. A third object layer 2656 shows that sub-objects can have further sub-objects, and so on.

FIGs. 27 to 29 illustrate an example of a scene description of a captured scene and its corresponding data structures.

FIG. 27 shows a scene 2700 (a captured image) containing a lamp 2714 and a bottle 2712 placed on top of a table 2710. For convenience, the scene 2700 is a 2-Dimensional (2D) image. The implementation for a 3-Dimensional (3D) image is the same but with an additional dimension in a *z* direction along a *z* axis representing depth. This depth is measured using the *z* axis perpendicular to the *x* axis (horizontal axis) and *y* axis (vertical axis) in FIG. 27. In the 2D scene 2700 of the present example, the three objects 2714, 2712, and 2710 were detected, identified and segmented by an ODIS method as described earlier. Objects with lesser chance of movement can be selected as reference points. In the scene 2700, the table 2710 is considered to have lesser chance of movement physically. Hence, the bottle 2712 and lamp 2714 positions can be defined with respect to the table 2710. The directional arrows 2740 defines the *x* and *y* directions along the *x* and *y* axes respectively for identification of pixel positions of the objects in the scene 2700. Pixel positions are all normalized to between 0 and 1 in the present example. If for example the image of scene 2700 has a width of 1024 pixels and a height of 768 pixels, the position [0.1, 0.2] denotes the pixel position $0.1 \times 1024 = 102.4$ pixels in the *x*-direction, and $0.2 \times 768 = 153.6$ pixels in the *y*-direction.

FIG. 28 is a diagram depicting a sample high level data structure for storing scene description data for the scene 2700 in FIG. 27. These data structures may be implemented in JSON, XML, Protobuf or records in a database. For this example, Name-Value pair structure as shown in FIG. 28 is used. At a top level (scene layer), there is a data structure 2800 describing a scene. In this example, the data structure 2800 contains the name of the scene (SceneName) with a value containing text of the name (Bedroom). The data structure 2800 also contains an average brightness of the scene (Brightness) and in this case a brightness value of 125. An ellipsis (...) in the data structure 2800 signifies that other properties of the scene are also included as part of the data structure 2800. At this top level, other properties of the scene that can be included may be white balance, RGB histogram values and so on. Any property useful for evaluation of the scene can be included, and this may vary from application to application. In this example, brightness of the scene may be used to determine whether the scene is

adequately lighted. The application may contain a range of valid brightness values. If the brightness of the scene is outside this range, a user can be informed on a mobile device (e.g. 1910 in FIG. 19) used to capture the scene so that the user can make adjustments to capture the scene again with better lighting.

The scene description data contains detected objects defined by data structures 2810, 2812, and 2814. Arrows 2820, 2822, and 2824 pointing from the data structure 2800 of the scene to the object data structures 2810, 2812 and 2814 indicate that the objects are part of the scene.

In this example,

- data structure 2810 defines a lamp
(ObjectName: Lamp1; ObjectType: Lamp; BoundingBox: [0.1,0.3,0.2,0.4]; CentreXY: [0.15,0.35]; IsRelative: Yes; Image: <Bitmap format> ...),
- data structure 2812 defines a table
(ObjectName: Table1; ObjectType: Table; BoundingBox: [0.3,0.1,0.5,0.7]; CentreXY: [0.4,0.4]; IsRelative: Yes; Image: <Bitmap format> ...), and
- data structure 2814 defines a bottle
(ObjectName: Bottle1; ObjectType: Bottle; BoundingBox: [0.15,0.3,0.2,0.4]; CentreXY: [0.175,0.35]; IsRelative: Yes; Image: <Bitmap format> ...).

Lamp 2810 and Bottle 2814 sit on top of a table, and therefore may derive their positions relative to the table data structure 2812, as indicated by arrows 2830 and 2832 in FIG. 28.

Each object structure 2810, 2812 and 2814 comprises the following parameters and fields: the object's name (ObjectName), the object's type (ObjectType), bounding box (BoundingBox), pixel position of the centre of the object (CenterXY), an indicator on whether the position is a relative position (IsRelative), and segmented image (Image) of the object. The bounding box is a rectangular box with the smallest possible dimensions enclosing the object, and it is usually defined by its top-left and bottom-right coordinates. In this example, the bounding box is defined by [x0, y0, x1, y1], where (x0, y0) are the top-left coordinates and (x1, y1) are the bottom-right coordinates.

The arrow 2830 indicates that object structure positions defined for the object of data structure 2810 are relative to positions in the object of data structure 2812. In other words, if the Table1 position (data structure 2812) is at [0.4, 0.4], and Lamp1 (data structure 2810) has a relative position at [0.15, 0.35], then the absolute position of the Lamp1 is at $[0.4+0.15, 0.4+0.35] = [0.55, 0.75]$.

The object data structure 2814 for Bottle1 may also be linked to sub-objects as illustrated in FIG. 29.

FIG. 29 shows the decomposition of the Bottle1 object (i.e. 2712 in FIG. 27), which has the data structure 2814 in FIG. 28, into a bottle cap (e.g. object 2404 in FIG. 24) and a bottle label (e.g. object 2406 in FIG. 24). The bottle cap and label are shown in the bottle 2712 in FIG. 27. FIG. 29 shows an object data structure 2814 for the Bottle1 object that is further linked to sub-object data structures 2900 and 2902 respectively. In the present example, each of the object data structures 2900 and 2902 contains the segmented images of the sub-objects 2404 and 2406 in FIG. 24 respectively. Arrows 2910 and 2912 indicate that the structures 2900 and 2902 are decomposed from the structure 2814.

Specifically, in this example,

- data structure 2900 defines the bottle cap
(ObjectName: BottleCap1; ObjectType: Object; BoundingBox: [0.12,0.31,0.14,0.34]; CentreXY: [0.13,0.325]; IsRelative: Yes; Image: <Bitmap format> ...), and
- data structure 2902 defines a bottle label
(ObjectName: BottleLabel1; ObjectType: Object; BoundingBox: [0.11,0.35,0.18,0.37]; CentreXY: [0.145,0.36]; IsRelative: Yes; Image: <Bitmap format> ...).

FIGs. 30 to 32 illustrate the method performed in the operation of the system 1950 described above with reference to FIG. 19.

FIG. 30 is a flow chart showing method steps involved in capturing a reference scene and generation of a scene description for the reference scene. Reference is made to components in the system 1950 in FIG. 19 for each method step described as follows. Firstly, a scene is grabbed or captured from a

mobile device with camera 1910 in step 900. In a next step 902, scene description (e.g. similar to the scene description described with reference to FIG. 27 to 29) is extracted from the scene by the server 1920 and/or computing service 1922. The extracted scene description is then stored together with the image of the scene (hereinafter “reference image”) in the storage medium 1924 in step 904. Thereafter, in step 910, the reference image and scene description is read by the computer 1930. This reading by the computer 1930 is done for the purposes of reviewing the scene captured and editing of the corresponding descriptions if required at step 912. The reviewing and editing of the scene captured can be done by a user or automated. Editing of the scene and/or corresponding descriptions may be required for additional parameters useful for scene evaluation and comparison. Examples of such parameters include a value indicating whether an object should have its image checked, and adding sub-objects to be evaluated. Once review and the editing are done, the scene description is updated in step 914 through the server 1920 to the storage medium 1924.

FIG. 31 is a flow chart illustrating method steps used for evaluation of captured scenes. Reference is made to components in the system 1950 in FIG. 19 for each method step described as follows. To start, a scene is captured or grabbed from the mobile device with camera 1910 in step 920. In a next step 922, scene description is extracted from the scene by the server 1920 and/or computing service 1922. Thereafter, in step 924, the extracted scene description of the newly captured scene is compared with the stored scene description of the reference scene described with reference to FIG. 30. This comparison action is performed by the server 1920 and/or computing service 1922. A decision has to be taken in a step 926 based on the comparison result from step 924. If the evaluation is deemed successful i.e. the compared images of the captured scene and reference scene matches closely, the process branches to step 928. In step 928, a success indicator may be saved and shown in the display of the mobile device 1910. If it is a failure i.e. the compared images of the captured scene and reference scene do not match, the process goes to step 930, wherein a failure indicator may be saved and shown. Highlighting of discrepancies described with reference to FIG. 23 may be saved and shown in the display of the mobile device 1910.

FIG. 32 is an expansion of the details in step 926 of FIG. 31. Essentially, FIG. 32 shows a process in which each layer in FIG. 26 is evaluated and compared to detect any discrepancies in terms of object presence and positioning. In step 940, the captured scene is compared with a stored reference scene. This involves comparison of a scene object (e.g. 2650 in FIG. 26) of the captured scene with a scene object of the reference scene. Specifically, each scene object has a data structure in a scene layer like the data structure 2800 in the example of FIG. 28. After comparison, in a step 942, the scene comparison results are stored for later evaluation. Thereafter, in step 944, objects within the captured scene are compared with objects in the reference scene. This involves comparison of all the objects in an object layer, for instance, the first object layer 2652 or the second object layer 2654 in FIG. 26. In the present example, top-level objects are compared first in step 944. Each top-level object has a data structure in the first object layer like the object structures 2810, 2812 and 2814 in the example of FIG. 28. In a subsequent step 946, the results for the object comparison are similarly stored for later evaluation. The process then moves to a decision step 948, wherein it is checked whether the objects compared in step 944 contain any sub-objects in a sub-object layer like the second object layer 2654 in FIG. 26. If there are sub-objects, the process returns to step 944 to compare the sub-objects. Specifically, each sub-object has a data structure in a sub-object layer like the data structure 2900 and 2902 in the example of FIG. 29. Steps 944 to 948 is a recursive process, wherein objects or sub-objects in successively lower object layers in the data hierarchy (e.g. FIG.26) are compared until there are no more sub-objects. Once all the objects or sub-objects in all layers have been compared, a next step 950 evaluates whether there are any discrepancies in any of the stored results i.e. results stored at steps 942 and 946. If there are discrepancies, the process branches to step 954, which indicates a failure output. If there are no discrepancies, the process moves to step 952, which produces a success indication. The success/failure indication is an output of step 924 in FIG. 32.

The following paragraphs describe an example of the system (e.g. 1950 of FIG. 19) that is implemented for hotel housekeeping application.

In this application, each housekeeper is given a mobile phone installed with a housekeeping app. This phone app contains a scheduling function to manage housekeepers' cleaning schedules. Each hotel room type in a hotel will have the same layout and amenities provided to the customer. FIG. 33 shows how a hotel room may be into decomposed into scenes and objects. The scenes available in a room type named *Hotel Room Type A* represented by 1000 can be split into many scenes, which include the

Bedroom 1002, Dressing Room 1004, Bathroom 1006, Living Room 1008 and more. These scenes also correspond to the areas, which a housekeeper would have to clean and tidy up in the course of his/her work. Each scene contains many different objects detected by a detection and segmentation algorithm of the system. For example, the Bedroom 1002 contains a bed, pillows, bed cover, side table, lamps among many other objects represented by a group 1012. There are other groups 1014, 1016, and 1018 showing objects that may be found in the Dressing Room 1004, Bathroom 1006 and Living Room 1008 respectively. For example, the Dressing Room 1004 contains a mirror, table, chair, comb, toiletries among many other objects represented by a group 1014. For example, the Bathroom 1006 contains a mirror, shower foam, soap, shampoo, towel among many other objects represented by a group 1016. For example, the Living Room 1008 contains a sofa, lamp, newspaper, writing pad, phone among many other objects represented by a group 1018.

Many hotels have standard housekeeping procedures, which specifies how objects have to be laid out. Taking a picture of a hotel room scene to be evaluated represents a significant productivity boost for the hotel. The housekeeper gets immediate feedback on the quality of work done, and the supervisor saves time on inspection tasks. In the longer term, the hotel owner can collate data on the quality of work done by individual housekeepers.

FIG. 34 is a flow chart of tasks for a housekeeper to perform work verification. Once the housekeeper finishes cleaning and arranging an area in a hotel room at step 1100, he/she proceeds to capture a scene of the area using a mobile device with a camera (e.g. 1910 in FIG. 19) at step 1102. The overlay aids described with reference to FIG. 21 are implemented on the mobile device and are used to guide the housekeeper to take a photograph or video of the scene. The captured scene is automatically transmitted to a server (e.g. 1920 in FIG. 19) and processed according to steps 920 to 930. At step 1104, the housekeeper obtains a result of scene evaluation on the mobile device that is received from the server. A decision step 1106 at the mobile device determines whether the result is a success or not. If the result is a success, the next step 1108 is a verification that housekeeping quality for the area has been met, and the housekeeper's work is done for the area. If the result is a failure, a failed count is updated and checked in a step 1112. The housekeeper is given a number of tries, K, to meet the housekeeping quality for the area. For each failure, the housekeeper has to clean and arrange the area again and steps 1102, 1104 and 1106 are repeated. If the failed count is less than the threshold number of tries, K, the housekeeper may review specific failed areas at a step 1114. At this step, the overlays described with reference in FIG. 21 are used to highlight problem areas (e.g. missing/wrong object, incorrect object arrangement, unclean object, etc.) on the housekeeper's mobile device. This allows the housekeeper to know which are the areas that need to be reworked in a step 1116. After the rework, he/she may capture the scene to be evaluated again at step 1102. If the housekeeper fails to get a successful result for more than K number of tries at step 1112, that area may be marked for a supervisor to conduct a manual check later in step 1110. In this way, the housekeeper can proceed on to work on another area in the room.

The system can be deployed to automate counting of items in a hotel refrigerator. If a guest takes any items from the refrigerator, the items are typically charged to the guest, and a restock order needs to be sent to relevant departments. With reference to FIG. 35, a housekeeper only needs to capture a scene of the refrigerator in a step 3500. At this point, the scene data will be sent to a server (e.g. 1920 in FIG. 19) for processing and the housekeeper may move on to other tasks. After scene evaluation at step 3502, a list of items taken is generated, and the action to be taken is decided by the server at step 3504. If there are items taken, the list of items to be charged is created and accessible for account settlement by the guest, and an order for restocking of the missing items is also created and sent to the devices and/or computer of relevant parties to notify them to take action at a step 3506. If no items are taken, no action needs to be taken is a final step 3508 of the process.

With reference to FIGs. 19 to 32, the detected discrepancies between captured one or more images and one or more reference images at the scene layer 2650 in FIG. 26 can be regarded as a first matching result in the present disclosure. The detected discrepancies between captured one or more images and one or more reference images at the object layer 2652 in FIG. 26 can be regarded as a second matching result in the present disclosure. The detected discrepancies between captured one or more images and one or more reference images at each sub-object layer 2654 and 2656 in FIG. 26 can be regarded as a third matching result in the present disclosure. The assessment to determine presence, movement and changes in appearance of objects relative to the one or more reference

images is determined based on the first and second matching results. The assessment may further take into consideration the third matching result.

Another example of the present disclosure is directed to application of Digital Image Processing and/or Artificial Neural Network (ANN) in the hospitality sector. The example may be in the form of a system, apparatus and/or method for providing an improved and more efficient technique for assessment of quality of housekeeping, including cleanliness, accounting of items (objects), positioning and orientation of items, etc. within an environment such as a room, lobby, lavatory, garden, and similar places that is subject to housekeeping or cleaning.

In a housekeeping setting in a hotel or service apartment, the example provides an inspection process to determine if housekeeping tasks are performed correctly and up to a predetermined standard. Based on results tracked by the supervisory inspection process such as number of times of re-cleaning done for a particular environment, and speed of cleaning the environment, performance metrics of a specific cleaner or cleaning staff or housekeeper can be generated and collated. Such metrics can be used to indicate the performance of the staff. Examples of the metrics that can be tracked include (1) the total time duration of a housekeeping cleaning session (start time and end time) for a particular environment such as a room, and (2) the time duration of each sub-area (for instance, a lavatory in a room) cleaning session. Preferred time duration for each of such housekeeping tasks can be set as a threshold value for comparison to determine if a housekeeping job is acceptable (a "pass job") or unacceptable (a "fail job"). The total number of passes and failures is subsequently collected. In this manner, supervisors and/or the relevant authority responsible for housekeeping monitoring will be able to obtain insights into overall and individual performance of their housekeeping staff.

By applying Digital Image Processing and/or an Artificial Intelligence (A.I.) algorithm (e.g. ANN), a system, apparatus and/or method of the present example can help to identify state of cleanliness of a room, recognise objects in the room and implement counting of the number of objects for accounting purposes. The entire inspection process may be implemented by first storing images of a perfectly clean ideal room. These images are then used for machine training i.e. inputted to a machine for machine learning and can be used as a reference for comparison later. For every image that is taken on for assessing housekeeping of the room subject to housekeeping, there can be trained reference images for actual real-time captured image to trained reference image comparison. The comparison techniques would involve digital image processing techniques, along with an Artificial Neural Network to decide how similar or different the compared images are. A score can be outputted to determine the similarity or difference between the compared images. Such score is indicative of the performance of the housekeeping personnel responsible for the room.

In the present example, after a cleaner performs a housekeeping task, for instance, for a room, the cleaner will be required to take photographs of areas where housekeeping has been performed through a mobile application on a mobile device. The photographs will sent to a backend server for processing via a communications network which may be wireless in nature. The backend server determines whether each photograph meets a criteria indicative of quality of housekeeping. A pass or fail message indicative of the quality of housekeeping will be sent to the mobile application so that the cleaner can take further action. If it is a fail message, the cleaner would need to repeat the housekeeping process for a particular area until a photograph taken of that room area can be given a pass by the backend server. This is equivalent to having a supervisor next to the housekeeper to assess and supervise whether the housekeeper has done the housekeeping in a preferred manner. The pass or fail criteria is determined by server software of the backend server, which may perform digital processing on images (i.e. the photographs) taken and compare the processed images with pre-stored ideal images of the same room. The function of the server software can include detection of foreign objects in the images, detection of dirt in the images, make sure that all objects are in a predetermined location and/or orientation, and identification of missing objects. Hence, detection of foreign objects that should not be present in the room, detection of dirt, detection of objects not placed in a predetermined location and/or orientation, and/or identification of missing objects can cause the system to determine failure in the housekeeping for a room area.

As there can be differences in lighting condition, angle and distance of images taken for the same objects in an environment, the images taken are typically non-ideal and needs to be digitally processed before they can be considered by a machine to determine whether the images should be given a pass or a fail. In the case that ANN is used, perfect room condition images need to be used as part of a

training set for machine learning to obtain a set of trained data models of the ANN. This ensures that determination or classification of clean (pass) or unclear (fail) state of a room by the ANN is accurate and meaningful.

- 5 The same technology can be applied for counting of items within, for instance, a minibar, a refrigerator, and the like in a room, and/or applied for counting of any other items (pillow, quilt, bolster etc.) that may be missing within the hotel room. AI algorithm for Pattern Recognition and ANN can be used to determine type of objects and even brand and model of the objects, for correctness and accounting purposes. Hence, detection of wrong brand and/or wrong model of the objects to be placed in a particular room area can cause the system to determine failure in the housekeeping for a room area.

A system according to the present example can operate as follows. A client device can be issued to each cleaner or housekeeper and a mobile application is installed on the client device to facilitate housekeeping monitoring and supervision. The client device can be a mobile device such as a smartphone, tablet device, mini-computer, and the like. The client device has a camera module for taking photograph and/or video. An example of a process flow 100 at the client device is shown in FIG. 1. At step 102, a cleaner starts the mobile application and upon doing so, the system pushes rooms of various housekeeping statuses into a list displayed on the client device. The list shown on each client device can be different. Each room can be indicated as dirty or reflect a failed quality control (QC) status. The cleaner may be required to login via the mobile application to an individual user account to see a list that is unique to the cleaner. The cleaner selects a room to clean from the list at step 104. Note that the term "cleaning the room" can include accounting of number of objects that should be present in the room in addition to tidying up, and/or cleaning and washing of areas in the room. The system captures a start time and the cleaner starts to clean the room. After the room is cleaned, the cleaner selects a room area for inspection at a step 106. A graphical user interface on the client device may require the cleaner to take a video or a static photograph of the room area at step 108.

In the case of a captured photograph at step 108, the captured photograph is subject to, in the present example, image processing at step 112 to determine whether the contents of the captured photograph would be depicted as a pass or a fail. Fail means a predefined threshold for housekeeping quality is not satisfied and pass means the predefined threshold is satisfied. The image processing will be described in more detail with reference to FIG. 4 to 6 later. In addition to image processing or as an alternative to image processing, step 112 can include a neural network or replace image processing with a neural network. The neural network can involve a classification module applying a trained neural network model to generate a score. The score can determine whether the captured photograph is a fail or a pass. More details on the use of neural network will be described later with reference to FIG. 7, and 12 to 17.

In the case of a failure at step 112, a check is performed at step 114 to determine whether there has been 3 failed attempts at cleaning. Note that other number of failed cleaning attempts can be set as well. A counter can be provided to count the number of failed attempts. If the number of the counter is not 3, the cleaner is given a chance to retake a photograph at step 116 in the case that the cleaner believes that the failure result is erroneous, or to re-clean the room area and submit another captured photograph of the room area. If the cleaner selects "yes" at step 116 to retake a photograph, the process flow goes to step 110.

If asked to do so, or the cleaner thinks that there is a technical problem with the system such as a wrong assessment in determining whether the cleaning attempt passes or fails for the selected room area, the cleaner may capture a video that is associated to the room area at step 118 to provide evidence for a supervisor to consider whether there is a technical problem.

If the image processing at step 112 is a pass, 3 failed attempts are detected at step 114, or the cleaner selects "no" (i.e. not re-taking a photograph) at step 116, or if a video is captured at step 118, step 120 is conducted to determine whether all predefined room areas have been covered. If not all the room areas have been covered, the process flow goes to step 106.

If step 120 determines that all room areas are covered, step 122 checks whether each of all the room areas are given a pass result. If all the room areas are given a pass result, the system captures an end time and generates an electronic record of a cleaned room that is pending supervision at step 124. If

not all the room areas are given a pass result, the system captures an end time and generates an electronic record of a room that requires a supervisor to check at step 126.

FIG. 2 shows an example of a process flow 200 of a supervisor accessing the system described with reference to FIG. 1 remotely via a network (e.g. internet) to review electronic records generated by the process flow 100 of FIG. 1.

The supervisor first logs in to a terminal or client device that can be a personal computer, laptop, smartphone, tablet computer and the like, which is connected to the network, to access the electronic records stored in a database updated by the system. Through a graphical user interface displayed on a screen of the terminal or client device, at step 202, the supervisor selects a “supervisor to check” option to view a list of electronic records for one or more rooms to review.

At step 204, the supervisor selects from the list an electronic record of a room to review. After selecting the electronic record of the room to review, the supervisor can select a room area to review from a list of room areas in the room at step 206. After selecting the room area, the supervisor sees a photograph displayed on the terminal or client device that had been determined by the system as passed or failed at step 208. The supervisor can choose to overwrite the pass or fail result given to the photograph of the room area by the system at step 210. If a video is uploaded as a record for the room area by a housekeeper responsible for the room area, the supervisor can choose to watch the video to determine whether to overwrite the pass or fail result.

After the supervisor overwrites the pass or fail result or decides not to overwrite the result, the supervisor can end the room area review by activating next step 212, for instance, by clicking a button, to prompt the system to check whether all the room areas have been reviewed by the supervisor. If not all the room areas have been reviewed, the process flow 200 goes to step 206. If all the room areas have been reviewed, step 214 is conducted to check whether all the room areas have been given a pass. If not all the room areas are given a pass, a failed quality control (QC) status is given to the electronic record of the room at step 218. If all the room areas are given a pass, a passed quality control (QC) status is given to the electronic record of the room at step 216.

FIG. 3 shows an example of a process flow 300 of a supervisor accessing the system described with reference to FIG. 1 and 2 on-site i.e. at a location close to a room or at the location of the room that the supervisor is reviewing.

The supervisor first logs in to a terminal or client device that can be a personal computer, laptop, smartphone, tablet computer and the like, which is connected to the network, to access the electronic records stored in a database updated by the system. Through a graphical user interface displayed on a screen of the terminal or client device, at step 302, the supervisor selects a Cleaned (i.e. Passed) or Failed list comprising a list of electronic records of one or more rooms to review. The supervisor selects a room or room area to review from the list at step 304. Thereafter, the supervisor has an option to select whether the room or room area is clean (i.e. whether housekeeping is done up to standard) at step 306. If the supervisor selects “yes” at step 306 i.e. the room or room area is clean, a passed quality control (QC) status is given to the electronic record of the room or room area at a step 308. Otherwise, if the supervisor selects “no” at step 310 i.e. the room or room area is not clean, a failed quality control (QC) status is given to the electronic record of the room or room area at step 310.

In another example, after selecting “yes” at step 306, the system can be configured to prompt the supervisor to take a photograph or a video of the room or room area that is regarded by the supervisor to be clean. The photograph or video is evidence and a record of a clean room or clean room area that can be used for machine learning. This photograph or video taken by the supervisor can be subject to image processing and/or processing by a neural network to determine whether the room or room area indeed satisfies predetermined housekeeping quality.

FIG. 4 to 6, when combined, shows an example of an architecture for the system described with reference to FIG. 1 to 3 for determining whether an input image should be given a pass or fail. This architecture is just one image processing implementation that can be used and examples of the present disclosure are not limited to this implementation. FIG. 4 illustrates process flow at a mobile application 400 of a client device (similar to the one described with reference to FIG. 1 to 3) with a camera module. FIG. 5 illustrates process flow at a backend system 500 (also called “assessment server”). FIG. 6

illustrates process flow of a backend algorithm 600 (i.e. algorithm executed by the assessment server) applied by the backend system 500. All the alphabets (A) to (F) in FIG. 4 to 6 are indicative of data flows that are interconnected between the elements of FIG. 4 to 6. (A) is joined to (A), (B) is joined to (B) and so on. The arrows in the FIG. 4 to 6 are indicative of the data flow direction.

With reference to FIG. 4 to 6, after a cleaner accesses or logs in to the mobile application 400, the cleaner selects a room area for inspection at a step 402. Upon selection, the backend system 500 pushes the selected room area to the mobile application 400 together with a pre-stored reference image of the room area at a step 502.

The mobile application 400 works with a camera application on the client device and activates a frontend algorithm 401. Once activated, the frontend algorithm 401 works with the camera module of the client device and enables the cleaner to take photograph and/or video of the room area that has been cleaned. Steps 404 to 410 are steps of this frontend algorithm. An image for assessing pass or fail is captured by the camera in real-time by the cleaner at step 404. After the image is captured, the frontend algorithm 401 extracts features from the image at step 404. The extracted features are matched with the features of the reference image at step 406. The number of matched features is calculated at step 408. At step 410, the number of matched features is checked to see if it passes a predetermined threshold. If the number of matched features do not pass the predetermined threshold, a graphical user interface of the mobile application 400 will prompt the cleaner to take another image (and to perform re-cleaning if necessary) and the process flow goes to step 404 to capture the re-taken image. If the number of matched features pass the predetermined threshold, the image or photograph is deemed to have passed at a first instance. When an image is passed at the first instance, an Application Programming Interface (API) is called to have the image uploaded to the backend system 500 at step 412.

In another example, the frontend algorithm 401 (i.e. steps 404 to 410) can be replaced by steps 920 to 930 described with reference to FIG. 31.

In practice, it is possible that a cleaner takes images and/or videos that are not aligned to the reference image in terms of pixel location of objects in the images. This can cause a lot of images to incorrectly fail to meet the predetermined housekeeping quality. In order not to load the backend server 500, which stores such failed images or videos, the frontend algorithm i.e. steps 404 to 410 are performed to get have the mobile application 400 perform a first level of image comparison. Several techniques can be used for feature extraction from images and for matching of the extracted features in steps 404 to 410. For example, Harris Corner Detector, SIFT (Scale-Invariant Feature Transform), SURF (Speeded-Up Robust Features), FAST (Features from Accelerated Segment Test), BRIEF (Binary Robust Independent Elementary Features) and ORB (Oriented FAST and Rotated BRIEF) are some techniques that can be used. In the present example, ORB is proposed as the technique to use for the frontend algorithm. The frontend algorithm determines how many features are matched between the captured image and the reference image before deciding to pass the captured image to the backend server 500. In the present example, only one reference image is used. However, in another example, more than one reference image may be used. The reference image or images may be products from a machine training stage in which many reference images taken of the same area and/or objects of interest in the room are inputted for the machine training. For process efficiency, the cleaner can be prompted by the graphical user interface a number of times (e.g. 3 times like in the example of FIG. 1) to re-take an image that is able to obtain a pass by the frontend algorithm. The threshold for step 410 can be set according to a user-defined level of tolerance. The threshold will affect the time needed to get a pass result and has to be set carefully. In this way, a quick and efficient assessment to determine whether a captured image is good enough for further processing is performed before the captured image is sent to the backend server 500 for further processing.

The frontend algorithm 401 can be configured to have an auto-capture function once the camera module is activated and the frontend algorithm 401 is running in the background of the client device. This auto-capture function is configured such that, without requiring the cleaner to activate a trigger (e.g. press a button on Graphical User Interface of the client device) to capture an image or images, the frontend algorithm 401 running in the background is configured to automatically perform steps 404 to 410 for each frame that is captured on the screen of the client device by the camera module. Once the frontend algorithm 401 is able to determine from one of the captured frames that the number of matched features between the image of that frame and the reference image are greater than or equal to the predetermined threshold, the frontend algorithm 401 will automatically send the image of that frame to the backend

server 500 for further processing. This feature helps to make the work flow more efficient as the cleaner just need to use the client device to scan the room area that is cleaned without putting in much effort to capture images.

- 5 At the backend system 500, when the uploading of the image that is passed at the first instance is received, the backend system 500 pushes the uploaded image with the reference image of the room area to the backend algorithm 600 for processing at a step 504.

At the backend algorithm 600, calculation is done at step 602 to calculate and find a best match of the image (i.e. captured image that has obtained a pass by the frontend algorithm) that is pushed from the backend system 500 to all relevant stored reference images of the room area. Step 602 comprises essentially steps 404 to 410 of the frontend algorithm. However, in step 602, instead of one reference image, more than one reference image are used for matching. Step 602 can be done, for instance, by using feature extraction algorithms such as AKAZE, SIFT, SURF, etc. to get feature points of the captured image. The feature extraction algorithm is applied on both the captured image as well as the reference image to be matched. In the present example, AKAZE is proposed as the technique to use for the backend algorithm 600. If one of the reference image has highest number of the same feature points as that of the captured image, this reference image is deemed as the best match and will be used for further processing and comparison. This reference image is subsequently referred as the "reference image (best match)".

At step 604, perspective transformation is performed on the captured image to adjust the image to a preferred perspective for processing. The preferred perspective may or may not be that of the reference image. If the preferred perspective is not that of the reference image, perspective transformation is also performed on the reference image (best match), to adjust the reference image (best match) to the preferred perspective for processing. Perspective transformation ensures that both the captured image and the reference image (best match) are aligned to a same orientation.

At step 606, a search is conducted to find aligned feature points between the captured image and the reference image (best match) and an alignment matrix is calculated. Thereafter, step 608 is conducted to find misaligned features between the captured image and the reference image (best match) and exclude any Outliers that are found. Feature extraction techniques such as Akaze, Sift and the like can be used. Using these techniques, all the aligned feature points and those that have the same feature points but are misaligned can be found. Misaligned feature points refer to feature points have the same features but may be found at different locations. Misaligned feature points are also called the Outliers. FIG. 18 illustrates matched and aligned feature points XX between the captured image 1802 and the reference image (best match) 1804. Outliers YY are feature points that are matched but at different locations in the captured image 1802 and the reference image (best match) 1804.

With reference to FIG. 6, with the number of matched feature points and the number of outliers detected and recorded, step 610 is conducted to calculate color clustering for the captured image (i.e. uploaded photograph) and the reference image (best match). Another image processing technique such as DB Scan, K-Means, and the like, can be applied. In step 610, the color clusters of the captured image and the reference image (best match) are grouped by colour i.e. according to Red, Green, and Blue (RGB) and/or Hue, Saturation and Value (HSV). The color clusters between the captured image and the reference image (best match) are then matched or compared according to the RGB or HSV values at step 612.

Step 614 checks whether the color clusters between the captured image and the reference image (with best match to the captured image) matches. If the color clusters do not match, a step 616 checks whether there are matched feature points (i.e. aligned feature points) between the captured image and the reference image (best match). If there are no matched color clusters and matched feature points between the captured image and the reference image (best match), step 618 checks whether misaligned feature points (i.e. outliers) are matched between the captured image and the reference image (best match). If even the misaligned feature points between the captured image and the reference image (best match) fail to match, then a failed state or status is issued along with a highlighting image superimposed on the captured image to highlight the part or parts in the image that failed to have any matches at step 620. If there is a "yes" to matching for any one of the color cluster matching done at step 614, the matched feature points matching done at step 616 and the matched misaligned feature points at step 618, a passed state or status will be issued at step 622. In the case

that stricter standards are preferred, the system can be set to treat presence of a misaligned feature as a fail.

In summary, the decision to pass the captured image can be made as follows:

- 5 1) Matched Features and/or Location Matched (without comparing colour clusters) = Pass
- 2) Matched Features, Location not matched (i.e. misaligned), but matched colour clusters = Pass
(This "Pass" condition can be set as "Fail" if stricter standards are required)
- 3) All feature points not matched and color clusters not matched = Fail

10 Hence, the matching results obtained from the checks described above can be used for assessment to determine presence, movement and/or changes in appearance of objects in the captured images relative to the reference image.

15 With reference to FIG. 4 to 6, the passed or failed result issued at step 622 or 620 is then obtained at the backend server 500 and sent to the mobile application 400 at step 506. The passed or failed result is displayed on a screen of the client device at a step 414.

In another example, the backend algorithm 600 (i.e. steps 602 to 622) can be replaced by steps 920 to 930 described with reference to FIG. 31.

20 When a camera module working with mobile application 400 is in use, a scene captured by the camera module shows up on screen. A cleaner moves the client device into a desired position and/or orientation to point the camera module at the desired scene to take a photograph. A button provided by the client device can be pressed to take the photograph. To facilitate accurate and efficient image matching, the image to be captured by the cleaner should be as consistent as possible with the pre-stored reference image (discussed with reference to FIG. 4 to 6). This means that, for instance, preferably each object that should be present in the images are at similar location and/or orientation. The frontend algorithm 401 can be configured to detect each object for each frame of the scene captured by the camera module on the screen, generate guiding outlines for each object, and display the guiding outlines for each object in each frame of the scene captured by the camera module on the screen. The displayed guiding outlines help to guide the cleaner to align actual outlines of the detected object with the displayed guiding outlines. Once aligned, the cleaner can proceed to take a photograph or video. In this way, the cleaner will be guided by the guiding outlines to take useful photograph or video for an area of interest. An example of this feature will be discussed in more detail with reference to FIG. 8.

35 FIG. 8 shows an image 800 of a room and first and second images 802 and 804 showing different views on a screen of a client device, which is a smartphone in this example, held by a cleaner responsible for cleaning the room. The client device is running a mobile application similar to the mobile application 400 of FIG. 4. As the cleaner directs a camera of the smartphone to take the first image 802, the cleaner may click on an area of interest such as a position 1 in the first image 802. After the cleaner selects the area of interest at position 1, a guiding outline 806 of a detected object, which is a bedframe in this case, appears. Likewise, as the cleaner moves the camera and the view of the second image 804 shows up on screen, the cleaner may click on an area of interest such as a location of a position 2 in the second image 804. After the cleaner selects the area of interest at position 2, the guiding outline 808 of a detected object, which is a round table in this case, appears. To take a good image that is useful and efficient for processing, the cleaner should try to move the camera position such that the guiding outline 806 or 808 matches the actual outline of the bedframe or the round table respectively. To improve efficiency and reduce time taken by the cleaner to take images, video mode can be implemented. A video is essentially made up of multiple frames of images captured per second and the multiple images can be sent to the backend system 500 of FIG. 5 for real-time matching with reference images. When a match is done and a pass or fail is determined, the pass or fail result is sent to the mobile application for display to the cleaner.

55 FIG. 9, 10, 11A and 11B show examples of records and reports that can be generated by the system described with reference to FIG. 1 to 6 to show housekeeper performance, yearly staff performance, room item review and room item detail review respectively. FIG. 9, 10, 11A and 11B also show parameters, values, and items that can be tracked.

60 Specifically, FIG. 9 can be said to be a report or dashboard 900A summarising parameters or items that are tracked for a room subject to housekeeping in each day of a selected month (e.g. September in year 2019). FIG. 9 is indication that each individual cleaner's work on all the rooms could be tracked.

Each room is further subdivided into specific areas (e.g. Area 2 - Bathroom) where one or more images are to be captured by a housekeeper to kept as a cleanliness record. Objects in the room (e.g. Bed, Furniture, etc.) and objects in each room area (e.g. in bathroom: wash basin(s), toiletries, etc.) are tracked. From FIG. 9, a supervisor can have an overview of how well each housekeeper performs housekeeping for all the areas within the room. For instance, whether housekeeping for specific object and/or room area was performed within allocated time (e.g. less than 30 minutes or within 31 to 60 minutes for some objects/area) or has exceeded the allocated time (e.g. exceed 60 minutes). In this example, housekeeping within or outside an allocated time range is also an indication of performance level, wherein exceeding the time range is regarded as poorer performance. Also, from FIG. 9, the supervisor can decide whether the housekeeper needs further training or help. FIG. 9 also provides information relating to key performance indices that a housekeeping management can use to decide on reward to be given to each housekeeper. FIG. 10 is a summary of "Yearly staff performance" and through this summary, the supervisor can make an assessment on how good his or her team of housekeepers are, and what could be done to perform better. In the example of FIG. 10, a supervisor can view, for a specified time and date, or specified date and/or time range, 2 staff with poor performance (housekeeping done greater than 30 minutes), 8 staff with medium performance (housekeeping done within 11 to 30 minutes) and 11 staff with good performance (housekeeping done less than 10 minutes). The supervisor is also able to refer to and zoom in on one or more images of objects or areas in the room through records 1100A and 1100B generated as shown in FIG. 11A and 11B respectively, especially those that are determined as "fail" and from there supervise and teach a housekeeper how to improve. Override button can be provided for the supervisor to override a decision made by the system. In the present disclosure, a room area can refer to an object. For instance, a bed, floor/carpet, etc. in the room can each be regarded as an area of the room.

As an alternative or as an addition to the architecture illustrated by FIG. 4 to 6, Artificial Intelligence (A.I.) technology such as Artificial Neural Network (ANN) and/or Machine Learning can be used instead or incorporated into the architecture of FIG. 4 to 6.

The system 1950 illustrated by FIG. 19 that is described earlier operates based on non-A.I. Image Processing technologies, in particular, reference image comparison technique. However, A.I. technology can also be used to improve the system 1950 to provide accurate pass or fail results. For example, A.I. Technology may be applied to enhance accuracy of pass/fail results by having inputs for machine learning structured based on the data structures described with reference to FIG. 26. The inputs for the trained A.I. system for determining pass/fail results can also be structured based on the data structures described with reference to FIG. 26. In other words, the data structures can be used as inputs for machine learning, and as inputs for pass/fail predictions when the A.I. system is live. If used as an addition to the system 1950, pass and fail results of the system 1950 determined by reference image comparison technique can be considered in addition to the pass and fail results produced through use of A.I. technology to provide more accurate pass and fail results. For example, appropriate weightages can be given to pass and fail results of the system 1950 determined by reference image comparison technique, and pass and fail results produced through use of A.I. technology to determine final pass and fail results (i.e. final visual assessment results).

An A.I. system may also be used for generating features of the reference images for image comparison performed by the system 1950 illustrated by FIG. 19 and pass/fail results are determined based on the image comparison techniques described with reference to FIG. 19 to 35. An example of such A.I. system is described earlier with reference to FIG. 36 and 37.

Furthermore, A.I. can be applied to produce one or more matching results for assessment to determine presence, movement and/or changes in appearance of objects in the captured images relative to the reference image.

Specifically, A.I. can be applied for image similarities/differences detection, object recognition, and object counting. FIG. 7 shows an example of an overall architecture of a system 700 involving use of A.I. technology. For Machine learning using ANN methodology, the system (such as 700 of FIG. 7) can be trained to recognise objects within a room by first inputting many images of different objects under different circumstances such as different lighting, different orientation, etc. By training the system to recognise the objects, a clearer or more precise knowledge of placement of wrong objects and/or missing can be detected in inputted images to be subject to housekeeping quality assessment.

The system 700 comprises a plurality of client devices 702 with a camera module for taking photographs and/or videos. The client devices 702 are mobile devices such as a smartphone, tablet device, mini-computer, and the like. Each of the client devices 702 is configured for a housekeeper to acquire one or more target images 706 (photograph and/or video) of an environment subject to housekeeping using the camera module. The one or more target images 706 are sent to an assessment server 724 via a network 708 (e.g. internet). The assessment server 724 processes the one or more target images 706 and returns a score for the one or more target images 706 acquired using the camera module.

The assessment server 724 comprises a scoring system 722. The scoring system 722 takes in received image or images 718 i.e. the one or more target images 706 as input images. The received image or images 718 are each subject to an image processing step 716 to process the image to a form for extracting parameters and/or features to input to a classification module 714 of a neural network. The classification module 714 uses trained data models 710 from a training server 704 and/or reference images in a reference images database 712 for comparison with the received image or images 718 to generate a score. The generated score is transmitted at a score transmission step 720 to the one or more client device 702. The score can be a value that can be further processed to a user-friendly format such as "pass" or "fail" that can be represented graphically on each client device 702 to indicate quality of housekeeping assessed for the one or more target images 706.

FIG. 12 shows an example of the training server 704 of FIG. 7. Images of areas of interest (e.g. specific areas in hotel rooms) need to be taken. These images are called reference images 1202. The reference images 1202 are inputted to the training server 704. A training process 1204 takes place based on inputs from a Neural Network and/or Deep Learning module 1210 to produce the trained data models 710 of FIG. 7. After receiving the reference images 1202, the training process 1204 involves sending them to a pre-processing module 1206 to convert them into a form that is suitable for image processing. After pre-processing, the pre-processed images are sent to an image processing module 1208 for image processing to detect and extract features from the pre-processed images. The processed images along with the extracted features are then sent to the Neural Network and/or Deep Learning module 1210 for machine learning. After machine learning, the trained data models 710 are produced and stored in a database accessible to the assessment server 724 of FIG. 7.

FIG. 13 shows details of the pre-processing module 1206 of FIG. 12. For the input reference images 1202 sent to the pre-processing module 1206 to be compared meaningfully in later stages, the images to be compared must be of the same type and format. Image pre-processing performed by the pre-processing module 1206 first performs a step 1304 to standardise image attributes. This involves normalisation of the input reference images 1202 into standard image attributes in terms of the image size, resolution, colour and shape. With all input images 1202 standardised to the same image size, resolution, colour (RGB) and shape (no. of RGB bits horizontal by vertical resolution), further processing for comparison is then possible. In another words, all images that are to be subject to machine training or later used for comparison are converted into the same resolution and format. If a decision is made to have all images in High Definition (HD), then any images that are taken that do not have that resolution will have to go through the pre-processing to convert the image into HD resolution. Image pre-processing can also optionally involve step 1306 to remove noise (if any) by a technique called Gaussian blur. A pre-processed output image 1308 is generated after step 1304 and optionally step 1306.

FIG. 14 shows details of the image processing module 1208 of FIG. 12. After Pre-processing, the pre-processed output image 1308 is sent to the image processing module 1208. Step 1404 is first performed to normalize the image 1308 to required attributes. Step 1404 is similar to step 1304 of FIG. 13. Step 1404 can be skipped if the normalization is identical as the normalization of step 1304. Step 1404 is only necessary if there are more or other parameters to normalize. After normalization, features detection at a step 1406 is performed, followed by features extraction at step 1408. At steps 1406 and 1408, the unique properties of objects in the image 1308 will be captured, such as lines of each object, corners, and/or any special patches that can describe the object in the image. After the features or feature points are extracted, the image 1308 along with the extracted features or feature points are sent to the Neural Network and/or Deep Learning module 1210 of FIG. 12 for machine training.

To illustrate detection of lines, corners and/or special patches done in steps 1406 and 1408, an example is described as follows with reference to FIG. 15. FIG. 15 shows a digital image 1500 that is divided into many small sectors, wherein each sector can comprise one or more pixels. Weightage values can

be given based on, for instance, the brilliance of the one or more pixels found in each sector in the image 1500. The weightage values are shown in each sector in FIG. 15.

In the example of FIG. 15, one can see that there is a significant jump in brilliance weightage values of some adjacent sectors. In this case, the sectors with values between 0.78 and 0.82 in the image 1500 can be grouped as a patch representing an object and the sectors with values between 0.12 to 0.18 can be grouped as a patch representing another object. The disruption in the pattern of the values between adjacent sectors can determine that there is present an outline of an object. In a similar manner, a corner can be detected if a sharp edge is present in a patch representing an object. Special patches that describe objects in the image can be detected by looking at the values of the attributes in the pixel or pixels in the sectors of the image as well. For instance, a table with round edges may be detected if patches resembling its shape is detected. Likewise, objects and features of the objects can be identified by looking at other attributes such as variation in colour values.

FIG. 16 shows details of the Neural Network and/or Deep Learning module 1210 of FIG. 12 for machine training. The processed image 1602 from the image processing module 1208 is sent to the Neural network (or Artificial Neural Network) and/or Deep Learning module 1210 for producing the final trained Data model or models 710 of FIG. 7. Hundreds (or a number that is practical) or thousands of images such as the processed image 1602 can be taken for various areas of interest within a room and sent to the Neural network (or Artificial Neural Network) and/or Deep Learning module 1210 for training. Note that in the present disclosure, an area of interest can be an object of interest, such as a bed, a table, etc. in the room. Each trained model can be for a specific area or object.

Artificial Intelligence in the form of Artificial Neural Network (ANN) and/or machine deep learning can be used for the classification of images. By introducing ANN, performance of the system is enhanced. Accuracy in detection of objects and finding of fault between compared images is improved. Errors and noises due to different lighting conditions, tolerances in distance to objects within images ... etc are eliminated or reduced.

Specifically, the processed image 1602 from the image processing module 1208 and object-specific training parameters 1610, which may include the extracted features or feature points of the processed image 1602, are sent to a machine learning classifier 1604. The machine learning classifier 1604 classifies the input images 1602 and outputs classified images 1606. The classified images 1606 are then evaluated and sorted at step 1608. During classification by the machine learning classifier 1604, all processed images such as the processed image 1602 (on objects of interest as well as objects of non-interest) are processed using image classification algorithms such as bag-of-words, support vector machines (SVM), K-nearest neighbors (KNN), logistic regression and the like. During evaluation and sorting of step 1608, the classified images 1606 may be classified into categories of pass, fail, unknown, or reject. Pass means that areas or objects of interest are recognised and/or at correct position and/or orientation and/or correct state (cleanliness state). Fail means that the areas or objects of interest cannot be found. Unknown means that the system is unable to determine if the areas or objects of interest are at correct position and/or orientation and/or correct state (cleanliness state). Reject means that there is an error in the determination of the areas or objects. After the images are categorised, the algorithm i.e. the model of the neural network and/or Deep Learning is modified accordingly and a trained data model 710 is output for use by the classification module 714 of FIG. 7. The classified images 1606 that have been evaluated and sorted can be stored in the reference images database 712 of FIG. 7 for use by the classification module 714.

Once the trained data models 710 are completed, the system is ready to accept actual live images for classification and scoring.

In one example, the classification module 714 of FIG. 7 comprises a neural network that is trained according to the machine training example of FIG. 16. For this example, the classification module 714 will be used for detecting similarity of objects or areas in a room and there will be one score output from the classification module 714.

However, in another example, the classification module 714 of FIG. 7 can be described as follows with reference to FIG. 17. This example involves score averaging or aggregation based on a first score from image matching involving comparison of parameters in room images and a second score from the use of a neural network that is trained according to the machine learning example of FIG. 16. The

classification module 714 of FIG. 17 is used for classification and scoring of the live images 718 of FIG. 7. The classification module 714 takes in as input the reference images from the reference images database 712 of FIG. 7, the trained data models 710 of FIG. 7 for neural network processing, and the input image 718 of FIG. 7 that has been captured by a housekeeper or cleaner for housekeeping quality assessment.

In one aspect of the classification module 714, the input image 718 is subject to an image matching step 1704 conducted between the reference images from the reference images database 712 and the input image 718. The image matching step 1704 can be conducted according to the backend algorithm 600 of FIG. 6 i.e. step 602 to step 620 or step 622. A first score is calculated based on the image matching results of the image matching step 1704.

In another aspect of the classification module 714, the input image 718 is sent for neural network processing 1710 using the trained data models 710 of FIG. 7. Foreign objects in the input image 718 that should not be present are identified and/or objects in the input image 718 that should be present are identified at a step 1711 and the neural network calculates a second score indicative of level of cleanliness at a step 1712 based on the object identification results of step 1711.

The first score and the second score are then averaged or aggregated at a score aggregation step 1714 to obtain a final score 1716, which will be transmitted to the respective client device 702 of FIG. 7 at a score transmission step 720 of FIG. 7. The final score 1716 is indicative of a pass or a fail in housekeeping quality (cleanliness) for the input image 718.

Scoring accuracy is enhanced when two aspects are considered in this example.

In yet another example, the classification module 714 of FIG. 7 can be configured to making a decision based on comparison between stored reference images from the reference images database 712 and the input images 718 taken by a housekeeper or cleaner. Objects within the images are compared according to all features that are extracted. Weightages can be assigned to different feature differences (e.g. color difference, brilliance difference, etc.) between the compared images, and a decision can be made by considering a score associated with a level of cleanliness (tidiness) of, for instance, a room, between a stored trained image (i.e. reference image from the reference images database 712) and a current input image 718. The score, if it is above or below a certain threshold, can be translated into a Pass or Fail that will be sent to the client device 702. The threshold is a parameter that can be increased or decreased to loosen or tighten the strictness of the standards, and also to cater to tolerances of errors in the image processing. For example, if there is a similarity between outline (features) of objects between a stored trained image (i.e. reference image from the reference images database 712) and a current input image 718 with a percentage difference error of 5%, and a colour and/or brilliance difference of 2%, and without any detection of foreign objects in the image, the score indicative of a pass decision can be generated.

With regard to overall performance metrics for a housekeeper or cleaner, that can include an aggregation of all passes or failures within a period of time over a number of rooms that the housekeeper or cleaner is involved in tidying. Such data can become a key performance index for a supervisor to evaluate efficiency and capability of the housekeeper or cleaner.

The neural network comprised in the classification module 714 of FIG. 7 can continue to be developed and constantly trained using the training server 704 for pattern recognition of objects within the room such as furniture, including table, chair, wardrobe, towel, the minibar items etc. using images of the respective objects. Furthermore, the images 706 and/or videos taken by the housekeeper or cleaner when the system is in operation can be subject to pre-processing of FIG. 13 and image processing of FIG. 14 and sent to the Neural Network and/or Deep Learning module 1210 of FIG. 12 for machine learning. The overriding inputs from supervisors in process flows of FIG. 2 and/or FIG. 3 are applicable to the present example involving Artificial Intelligence and can be taken into consideration for machine learning to improve accuracy as well.

To speed up machine training, in addition to supplying the machine with many manually taken images for training, images may be subject to automated modifications to produce more images for training purposes. For instance, the colors and lighting conditions of images that will obtain a pass result may

be modified slightly to produce images that would train the machine to recognise all of such modified images as a pass result.

Administrative users of the system may reload new reference images for the system to use due to changes in housekeeping standards when necessary.

There may also be a messaging interface that can be in a form of an instant messaging function provided in a mobile application that users may download and install to, for instance, their mobile device such as a smartphone, to enable the users to send housekeeping requests to housekeepers. Such messaging interface can be used by users including a guest staying in a room requiring housekeeping and/or a supervisor overlooking housekeeping work. The mobile application that the users downloaded and installed is capable of communicating with the mobile application (e.g. 400 in FIG. 4) accessed by the housekeepers on their client devices (e.g. 702 in FIG. 7). The system may be configured to provide a record of time of last done housekeeping job, and/or current status (e.g. busy, available, location, etc.) of a housekeeper to be displayed through the mobile application of the users on their client devices. From the displayed record, the user intending to send a housekeeping request through the messaging interface would be able to identify a preferred housekeeper to handle the request based on their location or availability.

The "assessment server" described in the present disclosure can comprise one or more processors executing instructions to control the assessment server to perform the intended tasks of the assessment server. Each of the one or more processor can be a semiconductor chip that typically resides in a computer and other electronic devices. Its basic job is to receive input and provide an appropriate output and it operates based on instructions that may include source code and the like. Each of the one or more processor can also refer to a central processing unit that handles system instructions, including processing user input/output (I/O) devices such as mouse and keyboard, and for running applications. Software involved in examples of the present disclosure may be stored and distributed in a non-transitory computer or machine readable storage medium such as a flash memory storage device, hard disk drive, and the like.

Examples of the present disclosure may have the following advantages.

- 1) Reduction of workload of supervisors to manually inspect areas in a building that are subject to housekeeping.
- 2) Supervisor productivity is increased as supervisors can now spend more time to train housekeepers or perform other duties.
- 3) Housekeeper productivity is increased as instantaneous feedback is provided on quality of housekeeping.
- 4) Instant feedback can be provided during housekeeping for missing items in a room or minibar in a room, which can be chargeable to guests staying in the room.

Examples of the present disclosure may have the following features.

A system (e.g. 1950 in FIG. 19 and 700 in FIG. 7) for visual inspection, wherein the system comprises:

a client device (e.g. 1910 in FIG. 19 and 702 in FIG. 7) with a camera module, wherein the client device is hand-operable by a user to capture one or more images (2150 in FIG. 21, 2200 in FIG. 22, 2700 in FIG. 17, and 800 in FIG. 8) of an area and is configured to send the one or more images to an assessment server (500 in FIG. 5, 724 in FIG. 7 and 1920 in FIG. 19); and

the assessment server,

wherein the client device or assessment server comprises one or more processors configured to execute instructions to control the client device or the assessment server respectively to:

match features of the captured one or more images with features of one or more reference images of the area to produce one or more matching results; and

assess the one or more matching results to determine presence, movement, and/or changes in appearance of one or more objects in the area,

wherein the client device is configured to provide graphical guidance (e.g. 806 and 808 in FIG. 8, 2160, 2162 and 2164 in FIG. 21, and 2170 in FIG. 21) to the user in a display of the client device to capture the one or more images of the area and the graphical guidance is provided based on features of the one or more reference images of the area.

The client device or the assessment server may be controllable to:

generate a data structure for the area captured in the captured one or more images (e.g. 2650 in FIG. 26 and 2800 in FIG. 28);

5 generate a data structure for each object of one or more objects in the area captured in the captured one or more images (e.g. 2652 in FIG. 26, and 2810, 2812 and 2814 in FIG. 28);

produce a first matching result between the data structure for the area captured in the captured one or more images, and one or more predetermined data structures of one or more reference images of the area;

10 produce a second matching result between the data structure for each object of the one or more objects in the area captured in the captured one or more images, and one or more predetermined data structures of each corresponding object of the one or more corresponding objects in the one or more reference images of the area; and

15 assess the generated data structures to determine presence, movement, and/or changes in appearance of one or more objects in the area based on the first matching result and the second matching result.

A portion of the one or more objects in the area captured in the captured one or more images may be defined as a sub-object of the one or more objects in the area, and a portion of each corresponding object of the one or more corresponding objects in the one or more reference images of the area may be defined as a sub-object of the corresponding object, wherein the client device or the assessment server is further controllable to:

20 generate a data structure for each sub-object of the one or more sub-objects in the area captured in the captured one or more images (e.g. 2654 and 2656 in FIG. 26, and 2900 and 2902 in FIG. 29),

25 wherein the assessment of the generated data structures is further based on the following matching result:

30 a third matching result between the data structure of each sub-object of the one or more sub-objects in the area captured in the captured one or more images, and one or more predetermined data structures of each corresponding sub-object of the one or more corresponding objects in the one or more reference images of the area.

35 The assessment of the generated data structures may be done using reference image comparison technique, and the results after applying the reference image comparison technique are considered to determine a final visual inspection assessment result for the captured one or more images of the area.

40 The assessment of the generated data structures may be done using neural network and/or deep learning technology (e.g. 722 in FIG. 7, and 1210 in FIG. 12), wherein the neural network and/or deep learning technology is used to predict one of the matching results, and predictions made by the neural network are considered to determine a final visual inspection assessment result for the captured one or more images of the area.

45 The assessment of the generated data structures may be done using both neural network and/or deep learning technology (e.g. 1718 in FIG. 17), and reference image comparison technique, and the predictions made by the neural network and/or deep learning technology and the results after applying the reference image comparison technique are considered to determine a final visual inspection assessment result for the captured one or more images of the area by the system.

50 The features of the one or more reference images of the area may be generated by neural network and/or deep learning technology.

55 The client device may be configured to enable a user to review an electronic record (e.g. 1100A in FIG. 11A and 1100B in FIG. 11B) comprising one or more images of the area and override the final visual inspection assessment result by the system.

60 The final visual inspection assessment result may be recorded and retrievable to be presented in an electronic report (e.g. 900A in FIG. 9) that tracks visual inspection assessment results for the area over a period of time.

The client device may be configured to prompt a user to capture one or more images as evidence to override the final visual inspection assessment result before the final visual inspection assessment result is overridden.

- 5 The final visual inspection assessment result may be used to account for one or more objects missing in the area.

An electronic restock request may be generated and notified to a user when missing object or objects are detected.

10

An electronic invoice may be generated based on the missing object or objects.

The client device may be configured to enable a user to select the area from a list of areas of an environment to be subject to visual inspection of the one or more images captured for the area.

15

The client device may be configured to enable a user to select the environment from a list of environments that is pushed to the client device.

20

The client device may be configured to receive a visual inspection request through a messaging interface.

The one or more captured images may be automatically captured by the client device when the camera module is activated without requiring user input to trigger capturing of one or more images.

25

The client device may be configured to:
display on the display guiding outlines (e.g. 806 and 808 in FIG. 8 and 2160, 2162 and 2164 in FIG. 21) in each image frame of the area captured by the camera module to guide a user to align outlines of an object in the image frame with the displayed guiding outlines and capture the one or more images of the area.

30

The client device may be configured to:
display on the display an indicator (e.g. 2170 in FIG. 21) of angular positioning of the client device, wherein the indicator is configured to indicate a preferred orientation of the client device to guide a user to orientate the client device to the preferred orientation and capture the one or more images of the area.

35

The client device may be configured to highlight, on the display, non-presence, movement, and/or changes in appearance of an object (e.g. 2314 and 2310 in FIG. 23) in the area captured in the captured one or more images as compared to the corresponding object in the one or more reference images of the area.

40

The client device may be configured to highlight, on the display, additional object (e.g. 2316 in FIG. 23) in the area captured in the captured one or more images that is not present in the one or more reference images of the area.

45

The system may be applied to housekeeping monitoring, wherein the assessment server is configured to determine whether a first predetermined level of housekeeping quality is satisfied for the area based on the matching results, and to provide an output indicative of whether the first predetermined level of housekeeping quality is satisfied.

50

The client device may be configured to determine whether a second predetermined level of housekeeping quality is satisfied for the area, wherein the second predetermined level of housekeeping quality is less stringent than the first predetermined level of housekeeping quality, and the second predetermined level of housekeeping quality has to be satisfied in order for the client device to send the one or more captured images to the assessment server for determine whether the first predetermined level of housekeeping quality is satisfied.

55

The system may be applied to check retail merchandising display of one or more objects in the area.

60

The system may be a warehousing system to monitor one or more objects being stored in the area.

A method for visual inspection, the method comprising:

- 5 a capturing step to capture one or more images of an area using a camera hand-operable by a user to capture the one or more images (e.g. 920 in FIG. 31, and 404 in FIG. 4);
 - a matching step to match features of the captured one or more images with features of one or more reference images of the area to produce one or more matching results (e.g. 922 and 924 in FIG. 31, and 404 in FIG. 4); an assessment step to assess the one or more matching results to determine presence, movement, and/or changes in appearance of one or more objects in the area (e.g. 406 and 408 in FIG. 4, 602 to 618 in FIG. 6, and 926 to 930 in FIG. 31); and
 - 10 a guiding step to provide graphical guidance to the user to capture the one or more images of the area, wherein the graphical guidance is provided based on features of the one or more reference images of the area.
- 15 The method may further comprise:
- in the matching step,
 - generating a data structure for the area captured in the captured one or more images,
 - generating a data structure for each object of one or more objects in the area captured in the captured one or more images,
 - 20 producing a first matching result between the data structure for the area captured in the captured one or more images, and one or more predetermined data structures of one or more reference images of the area, and
 - producing a second matching result between the data structure for each object of the one or more objects in the area captured in the captured one or more images, and one or more predetermined data structures of each corresponding object of the one or more corresponding objects in the one or more reference images of the area (e.g. 922 and 924 in FIG. 31); and
 - 25 in the assessment step,
 - 30 assessing the generated data structures to determine presence, movement and/or changes in appearance of one or more objects in the area based on the first matching result and the second matching result (e.g. 926 to 930 in FIG. 31).

- 35 The method above, wherein a portion of the one or more objects in the area captured in the captured one or more images may be defined as a sub-object of the one or more objects in the area, and a portion of each corresponding object of the one or more corresponding objects in the one or more reference images of the area may be defined as a sub-object of the corresponding object, wherein the method further comprises:
- in the matching step,
 - 40 generating a data structure for each sub-object of the one or more sub-objects in the area captured in the captured one or more images, and
 - producing a third matching result between the data structure of each sub-object of the one or more sub-objects in the area captured in the captured one or more images, and one or more predetermined data structures of each corresponding sub-object of the one or more corresponding objects in the one or more reference images of the area; and
 - 45 in the assessment step (e.g. 944 and 948 in FIG. 32),
 - 50 assessing the generated data structures to determine presence, movement and/or changes in appearance of one or more objects in the area based on the third matching result in addition to the first matching result and the second matching result (e.g. 950 to 954 in FIG. 32).

- Examples of the present disclosure can help to automate some tasks, which are visual in nature but are mundane and time-consuming. Examples of these tasks include visual inspection and stock counting. Examples of the present disclosure helps to alleviate visual overload when a task requires checking of a large number of objects. Examples of the present disclosure also eliminates subjective human biases by providing an objective visual assessment of a scene comprising a large number of objects to be checked.

More specific instances of the automated tasks include inspection of hotel rooms after housekeepers have done their job, determining whether arrangement of items of shelves are correct, and counting the number of items in a refrigerator or shelf to determine billing and whether restocking is required.

5 In the retail business sector, merchandising displays are an important part of the company's marketing strategy. Hence, there is strict control on how and where products are positioned. Examples of the present disclosure will help to check that the company's standards are being followed by providing automated visual checks at each of the retail locations.

10 The warehousing (including Fast-Moving Consumer Goods (FMCG)) business is another area where examples of the present disclosure may be put into use. The method and system deployed according to the examples of the present disclosure are able to track items/objects moved to each venue in the supply chain.

15 In the specification and claims, unless the context clearly indicates otherwise, the term "comprising" has the non-exclusive meaning of the word, in the sense of "including at least" rather than the exclusive meaning in the sense of "consisting only of". The same applies with corresponding grammatical changes to other forms of the word such as "comprise", "comprises" and so on.

20 While the invention has been described in the present disclosure in connection with a number of examples and implementations, the invention is not so limited but covers various obvious modifications and equivalent arrangements, which fall within the purview of the appended claims. Although features of the invention are expressed in certain combinations, it is contemplated that these features can be arranged in any combination and order.

CLAIMS

1. A system for visual inspection, wherein the system comprises:
 - a client device with a camera module, wherein the client device is hand-operable by a user to capture one or more images of an area and is configured to send the one or more images to an assessment server; and
 - the assessment server,
 wherein the client device or assessment server comprises one or more processors configured to execute instructions to control the client device or the assessment server respectively to:
 - match features of the captured one or more images with features of one or more reference images of the area to produce one or more matching results; and
 - assess the one or more matching results to determine presence, movement, and/or changes in appearance of one or more objects in the area,
 wherein the client device is configured to provide graphical guidance to the user in a display of the client device to capture the one or more images of the area and the graphical guidance is provided based on features of the one or more reference images of the area,
 - wherein the system is applied to housekeeping monitoring to determine whether predetermined level of housekeeping quality is satisfied, or the system is applied to check retail merchandising display of one or more objects in the area, or the system is a warehousing system to monitor one or more objects being stored in the area.
2. The system as claimed in claim 1, wherein the client device or the assessment server is controllable to:
 - generate a data structure for the area captured in the captured one or more images;
 - generate a data structure for each object of one or more objects in the area captured in the captured one or more images;
 - produce a first matching result between the data structure for the area captured in the captured one or more images, and one or more predetermined data structures of one or more reference images of the area;
 - produce a second matching result between the data structure for each object of the one or more objects in the area captured in the captured one or more images, and one or more predetermined data structures of each corresponding object of the one or more corresponding objects in the one or more reference images of the area; and
 - assess the first matching result and the second matching result to determine the presence, movement, and/or changes in appearance of the one or more objects in the area.
 3. The system as claimed in claim 2, wherein a portion of the one or more objects in the area captured in the captured one or more images is defined as a sub-object of the one or more objects in the area, and a portion of each corresponding object of the one or more corresponding objects in the one or more reference images of the area is defined as a sub-object of the corresponding object,
 wherein the client device or the assessment server is further controllable to:
 - generate a data structure for each sub-object of the one or more sub-objects in the area captured in the captured one or more images;
 - produce a third matching result between the data structure of each sub-object of the one or more sub-objects in the area captured in the captured one or more images, and one or more predetermined data structures of each corresponding sub-object of the one or more corresponding objects in the one or more reference images of the area; and
 - assess the third matching result in addition to the first matching result and the second matching result to determine the presence, movement, and/or changes in appearance of the one or more objects in the area.
 4. The system as claimed in claim 2 or 3, wherein the assessment of the generated data structures is done using reference image comparison technique, and the results after applying the reference image comparison technique are considered to determine a final visual inspection assessment result for the captured one or more images of the area.

5. The system as claimed in claim 2 or 3, wherein the assessment of the generated data structures is done using neural network and/or deep learning technology, wherein the neural network and/or deep learning technology is used to predict one of the matching results, and predictions made by the neural network are considered to determine a final visual inspection assessment result for the captured one or more images of the area.
5
6. The system as claimed in claim 2 or 3, wherein the assessment of the generated data structures is done using both neural network and/or deep learning technology, and reference image comparison technique, and the predictions made by the neural network and/or deep learning technology and the results after applying the reference image comparison technique are considered to determine a final visual inspection assessment result for the captured one or more images of the area by the system.
10
7. The system as claimed in any one of the preceding claims, wherein the features of the one or more reference images of the area are generated by neural network and/or deep learning technology.
15
8. The system as claimed in any one of claims 4 to 7, wherein the client device is configured to enable a user to review an electronic record comprising one or more images of the area and override the final visual inspection assessment result by the system.
20
9. The system as claimed in any one of claims 4 to 8, wherein the final visual inspection assessment result is recorded and retrievable to be presented in an electronic report that tracks visual inspection assessment results for the area over a period of time.
25
10. The system as claimed in claim 9, wherein the client device is configured to prompt a user to capture one or more images as evidence to override the final visual inspection assessment result before the final visual inspection assessment result is overridden.
30
11. The system as claimed in any one of claims 4 to 10, wherein the final visual inspection assessment result is used to account for one or more objects missing in the area.
12. The system as claimed in claim 11, wherein an electronic restock request is generated and notified to a user when missing object or objects are detected.
35
13. The system as claimed in claim 11 or 12, wherein an electronic invoice is generated based on the missing object or objects.
40
14. The system as claimed in any one of the preceding claims, wherein the client device is configured to enable a user to select the area from a list of areas of an environment to be subject to visual inspection of the one or more images captured for the area.
15. The system as claimed in claim 14, wherein the client device is configured to enable a user to select the environment from a list of environments that is pushed to the client device.
45
16. The system as claimed in any one of the preceding claims, wherein the client device is configured to receive a visual inspection request through a messaging interface.
50
17. The system as claimed in any one of the preceding claims, wherein the one or more captured images is automatically captured by the client device when the camera module is activated without requiring user input to trigger capturing of one or more images.
55
18. The system as claimed in any one of the preceding claims, wherein the client device is configured to:

display on the display guiding outlines in each image frame of the area captured by the camera module to guide a user to align outlines of an object in the image frame with the displayed guiding outlines and capture the one or more images of the area.

- 5 19. The system as claimed in any one of the preceding claims, wherein the client device is configured to:
display on the display an indicator of angular positioning of the client device, wherein the indicator is configured to indicate a preferred orientation of the client device to guide a user to orientate the client device to the preferred orientation and capture the one or more images of the area.
10
20. The system as claimed in any one of the preceding claims, wherein the client device is configured to highlight, on the display, non-presence, movement, and/or changes in appearance of an object in the area captured in the captured one or more images as compared to the corresponding object in the one or more reference images of the area.
15
21. The system as claimed in any one of the preceding claims, wherein the client device is configured to highlight, on the display, additional object in the area captured in the captured one or more images that is not present in the one or more reference images of the area.
20
22. The system as claimed in any one of the preceding claims, wherein in the case the system is applied to housekeeping monitoring, the assessment server is configured to determine whether a first predetermined level of housekeeping quality is satisfied for the area based on the matching results, and to provide an output indicative of whether the first predetermined level of housekeeping quality is satisfied.
25
23. The system as claimed in claim 22, wherein the client device is configured to determine whether a second predetermined level of housekeeping quality is satisfied for the area, wherein the second predetermined level of housekeeping quality is less stringent than the first predetermined level of housekeeping quality, and the second predetermined level of housekeeping quality has to be satisfied in order for the client device to send the one or more captured images to the assessment server for determine whether the first predetermined level of housekeeping quality is satisfied.
30
35
24. A method for visual inspection, the method comprising:
a capturing step to capture one or more images of an area using a camera hand-operable by a user to capture the one or more images;
a matching step to match features of the captured one or more images with features of one or more reference images of the area to produce one or more matching results;
an assessment step to assess the one or more matching results to determine presence, movement, and/or changes in appearance of one or more objects in the area; and
45 a guiding step to provide graphical guidance to the user to capture the one or more images of the area, wherein the graphical guidance is provided based on features of the one or more reference images of the area,
wherein the method is applied to housekeeping monitoring to determine whether predetermined level of housekeeping quality is satisfied, or the method is applied to check retail merchandising display of one or more objects in the area, or the method is a warehousing system to monitor one or more objects being stored in the area.
50
25. The method as claimed in claim 24, wherein the method further comprises:
in the matching step,
55 generating a data structure for the area captured in the captured one or more images,
generating a data structure for each object of one or more objects in the area captured in the captured one or more images,

producing a first matching result between the data structure for the area captured in the captured one or more images, and one or more predetermined data structures of one or more reference images of the area, and

5 producing a second matching result between the data structure for each object of the one or more objects in the area captured in the captured one or more images, and one or more predetermined data structures of each corresponding object of the one or more corresponding objects in the one or more reference images of the area; and in the assessment step,

10 assessing the first matching result and the second matching result to determine presence, movement and/or changes in appearance of one or more objects in the area.

26. The method as claimed in claim 25, wherein a portion of the one or more objects in the area captured in the captured one or more images is defined as a sub-object of the one or more objects in the area, and a portion of each corresponding object of the one or more corresponding objects in the one or more reference images of the area is defined as a sub-object of the corresponding object,
15 wherein the method further comprises:
in the matching step,

20 generating a data structure for each sub-object of the one or more sub-objects in the area captured in the captured one or more images, and

25 producing a third matching result between the data structure of each sub-object of the one or more sub-objects in the area captured in the captured one or more images, and one or more predetermined data structures of each corresponding sub-object of the one or more corresponding objects in the one or more reference images of the area; and in the assessment step,

assessing the third matching result in addition to the first matching result and the second matching result to determine the presence, movement and/or changes in appearance of one or more objects in the area.

30