



(51) International Patent Classification:

G06T 19/00 (2011.01) G06K 9/46 (2006.01)  
G06K 9/00 (2006.01)

(21) International Application Number:

PCT/US2020/054939

(22) International Filing Date:

09 October 2020 (09.10.2020)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

62/914,087 11 October 2019 (11.10.2019) US

(71) Applicant: SWIMC LLC [US/US]; 1100 Midland - Legal Department, 101 W. Prospect Ave., Cleveland, OH 44115 (US).

(72) Inventors: WILLIAMS, Preston; 406 Elmwood Road, Bay Village, OH 44140 (US). DO, Brendan; 36616

Rummel Mill Drive, North Ridgeville, OH 44039 (US). RICHMOND, Daniel Cody; 35064 Sullivan Drive, North Ridgeville, OH 44039 (US). DOWELL, Michael; 605 Joseph Drive, Aurora, OH 44202 (US).

(74) Agent: SCOTT, James C.; The Sherwin-Williams Company, 1100 Midland - Legal Department, 101 W. Prospect Ave., Cleveland, OH 44115 (US).

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

(54) Title: AUGMENTATION OF DIGITAL IMAGES WITH SIMULATED SURFACE COATINGS

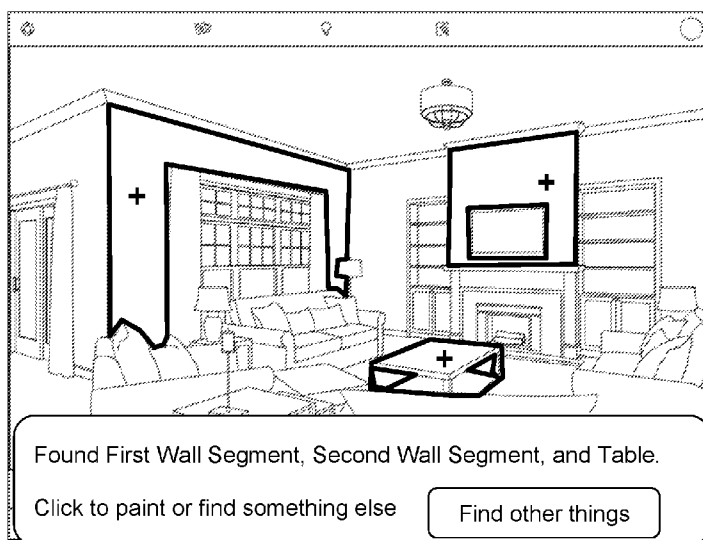


FIG. 2C

(57) Abstract: An image augmentation system and method for simulating the application of a coating to a surface of an original image. A mask generator can perform a search to detect a recognized object depicted in the original image, and generate a object mask for the recognized object. A user interface can to receive an identification of a selected object and an identification of a selected coating to be simulated on the selected object, and display a painted image. A Tenderer can receive the object mask for the selected object and the selected coating and generate the painted image, each pixel of the painted image having a painted color determined based on the object mask, the coating, and the original image.



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

**Published:**

- *with international search report (Art. 21(3))*
- *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

**AUGMENTATION OF DIGITAL IMAGES WITH SIMULATED SURFACE COATINGS**

5

**TECHNICAL FIELD**

Embodiments of the present disclosure relate generally to the field of computer graphics, and more particularly to augmenting digital images to depict the simulated application of coatings.

10

**BACKGROUND**

Visual aids can assist purchasers in the selection of paints, stains, or other coatings for application to a surface. One form of visual aid are paint chips, or color swatches, in which the coating has been applied to a physical surface, that can provide a truer sense of what a coating looks like when applied. Such swatches are limited in that they are often small, and aren't applied to the entirety of a surface that the purchaser may wish to cover.

Computers and mobile devices provide the ability to modify digital images. For example, photo editing tools enable regions of pixels can be selected to be recolored, either by filling with a solid, opaque color, or by using one or more translucent overlays. In order to realistically portray how a coating might look on a given surface using a photo editing tool, a number of manual techniques are required to select the appropriate pixels and apply coloration while maintaining areas of shadow, highlight, or other factors that affect how a pixel of a surface should be colored.

The field of augmented reality has developed a number of techniques for automatically augmenting a displayed image to, for example, depict objects that don't exist in the physical scene,

or to alter the appearance of existing objects with one or more overlays. Such systems often rely on edge detection or other techniques known in the art to identify regions of similar colors that can be assumed to make up a continuous surface. These systems often struggle to determine the pixel boundaries of objects or surfaces when one or more intervening objects partially block the view.

5           A need exists, therefore, for improved systems and methods to modify images to simulate the application of a coating to a surface depicted in a digital image.

### SUMMARY

Embodiments of the present disclosure address the need for improved systems and methods  
10 to modify images to simulate the application of a coatings to a surface depicted in a digital image.

In an embodiment, an image augmentation system for simulating the application of a coating to a surface of an original image comprising a plurality of original pixels can comprise at least one memory and at least one processor configured to implement a mask generator configured to receive the original image, perform a search to detect a recognized object depicted in the original  
15 image, and generate a object mask for the recognized object comprising a mask pixel corresponding to each of the plurality of original pixels, at least one parameter of each mask pixel having a first value if the corresponding original pixel is part of the recognized object and a second value if the corresponding original pixel is not part of the recognized object, a user interface configured to receive an identification of a selected object and an identification of a selected  
20 coating to be simulated on the selected object, and display a painted image, and a renderer configured to receive the object mask for the selected object and the selected coating and to

generate the painted image, each pixel of the painted image having a painted color, the painted color determined to be the same color as the corresponding pixel of the original image if the corresponding mask pixel has the second value, and determined based on the coating and the corresponding original pixel if the corresponding mask pixel has the first value.

5           In embodiments, the mask generator comprises a class recognizer configured to identify a class of object present in an area of the original image. In embodiments, the mask generator further comprises an image segmenter for each of a plurality of classes configured to detect a recognized object by identifying the pixels of the original image that include the recognized object. The class recognizer and the image segmenter can each comprise a neural network.

10           In embodiments, the painted color is further determined based on at least one tint plane adjustment map defining a difference between a base color and the color of the corresponding original pixel. The difference between the base color and the color of the corresponding original pixel can be based on at least one of the hue, the saturation, and/or the luminosity of the base color and the color of the corresponding original pixel.

15           In embodiments the coating and the at least one tint plane adjustment map can comprise filter primitives, and the renderer can be configured to generate the painted image by applying the filter primitives to the original image.

20           In embodiments, an method for simulating the application of a coating to a surface of an original image comprising a plurality of original pixels can include receiving an original image file, performing a search to detect a recognized object depicted in the original image, generating

an object mask for the recognized object comprising a mask pixel corresponding to each of the plurality of original pixels, at least one parameter of each mask pixel having a first value if the corresponding original pixel is part of the recognized object and a second value if the corresponding original pixel is not part of the recognized object, displaying, on a user interface  
5 device, a first painted image depicting the original image and a selectable area corresponding to each recognized object, receiving, from the user interface device, a user selection of a selected object and a coating, rendering a painted image, each pixel of the painted image having a painted color determined to be the same color as the corresponding pixel of the original image if the corresponding mask pixel has the second value, and determined based on the coating and the  
10 corresponding original pixel if the corresponding mask pixel has the first value.

While various embodiments are amenable to various modifications and alternative forms, specifics thereof have been shown by way of example in the drawings and will be described in detail. It should be understood, however, that the intention is not to limit the claimed inventions to the particular embodiments described. On the contrary, the intention is to cover all modifications,  
15 equivalents, and alternatives falling within the spirit and scope of the subject matter as defined by the claims.

#### BRIEF DESCRIPTION OF THE DRAWINGS

Subject matter hereof may be more completely understood in consideration of the  
20 following detailed description of various embodiments in connection with the accompanying figures.

FIG. 1 is a schematic diagram depicting an image augmentation system, according to an embodiment.

FIGS. 2A-2F are mockups depicting example user interface screens of an image augmentation application, according to an embodiment.

5 FIG. 3 is a schematic diagram depicting the data structure of a coating, according to an embodiment.

FIG. 4 is a schematic diagram depicting object masks, according to an embodiment.

FIG. 5 is a schematic diagram depicting components of a mask generator, according to an embodiment.

10 FIG. 6 is a mockup depicting an example user interface screen, according to an embodiment.

FIG. 7 is a mockup depicting tint plane mask images, according to an embodiment.

FIG. 8 is a flowchart depicting a method for generating object masks, according to an embodiment.

15 FIG. 9 is a flowchart depicting a method for generating a probability map, according to an embodiment.

FIG. 10 is a schematic diagram depicting components of a renderer, according to an embodiment.

FIG. 11 is a mockup depicting an example painted image, according to an embodiment.

20 FIG. 12A is a code listing depicting a partial HTML document, according to an embodiment.

FIG. 12B is a code listing depicting a partial HTML document, according to an embodiment.

FIG. 13 is a flowchart depicting a method for rendering a painted image, according to an embodiment.

5 FIG. 14 is a schematic diagram depicting architectural elements of an image augmentation system, according to an embodiment.

While various embodiments are amenable to various modifications and alternative forms, specifics thereof have been shown by way of example in the drawings and will be described in detail. It should be understood, however, that the intention is not to limit the claimed inventions to  
10 the particular embodiments described. On the contrary, the intention is to cover all modifications, equivalents, and alternatives falling within the spirit and scope of the subject matter as defined by the claims.

#### DETAILED DESCRIPTION

15 FIG. 1 depicts a system 100 for augmenting an input image to depict a simulated coating applied to a surface in the input image. System 100 can comprise user interface 200, mask generator 300, and renderer 400.

User interface 200 can comprise a mobile application, web-based application, or any other executable application framework. User interface 200 can reside on, be presented on, or be  
20 accessed by any computing devices capable of communicating with the various components of system 100, receiving user input, and presenting output to the user. In embodiments, user interface

200 can reside or be presented on a smartphone, a tablet computer, laptop computer, or desktop computer.

FIGS. 2A-2F depict example screens that can be provided by user interface 200 to present output to, and receive input from, a user according to an embodiment. Each screen of user interface  
5 200 can present controls or other interfaces enabling the user to request one or more actions from system 100. The operation of system 100 with respect to the screens depicted in FIGS 2A-2F is provided in further detail herein. While FIGS. 2A-2F are depicted, embodiments of system 100 can provide more, fewer, or alternative screens.

As depicted in FIG. 2A, the user can be prompted to select or upload an original, or input,  
10 image 202 from a memory. User interface 200 can further enable the user to control an image capture device such as a camera connected or integrated into a computing device to capture an image on demand. As depicted in FIG. 2B, input image 202 can be displayed, and the user can be prompted to request that objects be automatically recognized. In alternative embodiments, object recognition can begin without waiting for user input.

As depicted in FIG. 2C, a screen can be provided that identifies objects recognized in input  
15 image 202 and requests user feedback. If the user does not want to click one of the recognized objects, a screen such as that depicted in FIG. 2D can request confirmation that depicted scene is correctly identified. FIG. 2E depicts an example screen enabling the user to choose a selected coating 204 for application on a recognized object. FIG. 2F depicts an example screen depicting  
20 painted image 402, which has been modified to depict the selected coating.

Referring again to FIG. 1, input image 202 can comprise a digital image in any image file

format known in the art, such as bitmap (BMP), raw, graphics interchange format (GIF), joint photographic experts group (JPEG, or JPG), portable network graphics (PNG), or the like. Input image 202 can further comprise a single frame or image from a sequences of images such as animated GIF files, or video formats such as moving picture experts group (MPEG, or MP4), audio  
5 video interleave (AVI), or the like. While input image 202 and other image data described herein can include vector graphics data, the data structure of input image 202 and other images will be described, conceptually, herein as a two-dimensional array of pixels.

10 Pixels within an image file can each comprise one or more parameters defining the visual appearance of the pixel when output to a display, a printer, or other image viewer. Pixel parameters can comprise color information in red/green/blue (RGB), hue/saturation/value (HSV), cyan/magenta/yellow/black (CMYK), or other computer-comprisable color definition formats, as well as transparency parameters such as an alpha value.

Referring now to FIG. 3, selected coating 204 can be one of a plurality of coatings 206 provided by system 100. Each coating 206 can represent a paint, stain, texture, or other item or  
15 surface treatment, for simulated application by embodiments of the present disclosure. According to an embodiment, each coating 206 can be presented by a data structure comprising an identifier 208, a type 210, a principal color 212, opacity information 214, and a texture map 216. In embodiments, identifier 208 can be a model or product number or name, or can be associated with a standard color scheme such as the PANTONE color scheme. Type 210 can indicate a category  
20 or style of coating represented. For example, types can include “paint,” “stain,” or “texture.” Embodiments can further support subtypes such as “gloss paint,” “matte paint,” “opaque stain,”

“translucent stain,” or the like.

Principal color 212 can be the main color of a coating that may have multiple pigment colors, or may take on multiple colors based on the surface substrate. For example, for coatings of type “paint,” the principal may comprise color information that is based on the color of the main pigment in the coating.

Opacity information 214 of a coating can be a single value, for example, 100% for opaque paints and stains, or can comprise image data with each pixel having varying alpha or other values to indicate uneven transparency. Texture map 216 can comprise image data with each pixel having a parameter value determined based on high or low points in the texture of the coating 206. The image data for opacity 214, and texture map 216, if present, can comprise tileable images, such that the image data can be repeated across a region of pixels in the painted image 402.

The data representing coatings 206 can be stored in a memory or data store of system 100. In embodiments, coatings 206 can correspond with one or more existing or proposed products, and the data regarding coatings 206 can be retrieved from a product information system included within, or remote from, system 100. Coatings can further comprise, or be convertible to, scalable vector graphics (SVG) filter primitives.

Mask generator 300 can receive input image 202 and generate one or more object masks 302. As depicted in FIG. 4, each object mask 302 can be associated with a recognized object 304, and include a tint plane mask image 306 (or simply, tint plane 306) and a probability map 308. Objects masks 302 can define the pixels of input image 202 that depict a paintable, or tintable surface within input image 202. Each recognized object 304 can be an entire object (such as a

table), or an individual surface of an object (such as the top of a table).

FIG. 5 is a schematic view depicting components of mask generator 300. Class recognizer 310 can comprise a classifier that has been trained to identify areas of an image that depict an object in one or more classes of common objects. In embodiments, class recognizer 310 can be trained to identify classes of objects such as interior and exterior architectural surfaces, including walls, floors, ceilings, and wall/ceiling trim and molding; architectural scenes, such as living room, kitchen, bedroom, bathroom, entryway, office, recreation room, garage, front exterior, and rear exterior; kitchen and bathroom cabinets of all type; common forms of furniture, including tables of all type, chairs of all type, desks, armoires, wardrobes, sideboards, bookcases, benches, entertainment centers, consoles, and chests of drawers, and the like.

Class recognizer 310 can return a recognized class 312 data structure for each object class detected in the original image. Each recognized class 312 can comprise an identifier of the class, and, in embodiments, coordinates locating where the class was identified within the image. The coordinates can be pixel coordinates defining a bounding box. For example, a tuple of (x, y, width, height) can represent the x and y coordinates for the top left corner of a bounding box, and the width and height can define the extent of the bounding box. The coordinates can also be a more detailed representation of the perimeter of the area where the object class was recognized. In other embodiments, the identifier of the class can be provided without additional coordinate or location information.

In embodiments, class recognizer 310 can also deduce one or more insights regarding the depicted scene, for example, class recognizer 310 may determine that an input image that includes

objects of certain classes (for example a sink, a refrigerator, and a dishwasher), that the scene depicts a kitchen. In another example, if a driveway object, a front door object, and one or more window objects are detected, class recognizer 310 may determine that an outdoor view of the front of a house is depicted. In embodiments, the type of scene can be used to narrow the potential classes searched for recognized objects. For example, if a region of the image is recognized as both a “bed” and a “sofa,” but other classes within the image indicate that the scene depicts a bedroom, the recognized class 312 indicating that a region depicts a bed may be given additional weight.

Image segmenter 314 can comprise a plurality of object classifiers 316 each trained to identify the boundaries of objects of a particular class. For example, image segmenter 314 can comprise an interior wall recognizer, a sofa recognizer, an entertainment center recognizer, and/or the like. Image segmenter 314 can receive each recognized class 312 and generate a tint plane mask image 306 for each recognized object 304. If more than one recognized object 304 is found, the tint plane mask image 306 can be created in a predetermined order of priority (for examples, walls, followed by ceilings, followed by trim, followed by furniture, etc.). Each object classifier 316 can search the coordinate space identified by recognized class 312 for an object of the associated type without having to search the entire image. Because class recognizer 310 can operate at a less detailed level before object classifiers 316 perform more precise identification and location of objects, mask generator 300 can have improved efficiency and accuracy compared to systems that are trained only to recognize specific objects across an entire image.

Each of class recognizer 310 and the plurality of recognizers of image segmenter 314 can

comprise a neural network, or other deep learning network such as a based on a deep learning network such as a convolutional neural network (CNN). Each neural network can be implemented as a TensorFlow model, and can comprise a neural network with an architecture based on or similar to ResNet, AlexNet, GoogLeNet, DeepLab, VGG, or other neural network architecture.

5 TensorFlow is an open source machine learning library published by Google, LLC. ResNet is a residual learning framework described by Kaiming He et al., *Deep Residual Learning for Image Recognition*, ARXIV:1512.03385 (2015). AlexNet is a convolutional neural network framework described by Alex Krizhevsky et al., *ImageNet Classification with Deep Convolutional Neural Networks*, 25 ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS 1097 (2012).

10 GoogLeNet is a convolutional neural network architecture described by Szegedy et al., *Going Deeper with Convolutions*, ARXIV:1409.4842 (2015). DeepLab is a convolutional neural network framework described by Liang-Chieh Chen et al., *DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs*, ARXIV:1606.00915 (2016). VGG is a convolutional neural network framework described by Karen Simonyan et al.,

15 *Very Deep Convolutional Networks for Large-Scale Image Recognition*, ARXIV:1409.1556 (2014).

Each neural network be trained using a plurality of labeled training images, and supervised and unsupervised neural network training techniques can be used. For example, deeply annotated training images from the publicly available ADE20K dataset can be used in addition to proprietary training data. The ADE20K dataset is published by the Massachusetts Institute of Technology, and

20 described by Bolei Zhou, et al., *Scene Parsing through ADE20K Dataset*, PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2017. Training images can

be annotated to include labels identifying of parts of objects as well as whole objects.

Probability mapper 318 can receive each tint plane mask image 306 and a generate probability map 308 for each recognized object 304. Probability map 308 can comprise image data and each pixel in the probability map 308 can correspond to a pixel in input image 202. Probability  
5 map 308 can be represented in memory as a one-dimensional array, with every pixel of the source image represented with a single integer value 0-99. Pixels in probability map 308 can be ordered in the same way that image data is returned from an image processing API, such as the hypertext markup language 5 (HTML5) Canvas API's getImageData method – right to left and top to bottom, wrapping from the end of each image pixel row to the beginning of the next image pixel row.  
10 Probability map 308 can be defined by probability mapper 318, and updated and refined by other components of system 100.

At least one parameter of each probability map pixel (for example, red, green, blue, alpha, or other parameters) can have a probability value based on the confidence that the pixel in the input image is part of the associated recognized object 304. For example, the alpha (or transparency)  
15 value for each pixel can have a value of one hundred for pixels that are part of the interior of a recognized object, 1-99 for pixels that are part of an anti-aliased edge of the recognized object, and zero for pixels that are not part of the recognized object.

Embodiments of mask generator 300 can correctly identify up to 95% of the pixels that comprise a paintable surface of a depicted object that is represented by at least 1,000 pixels in the  
20 input image 202. The number of false positives (pixels incorrectly identified as being part of a paintable surface) can be fewer than 5% of the total number of pixels that actually comprise the

surface.

In embodiments, mask generator 300 can receive confirmation from the user at one or more phases of the mask generation process. For example, in one embodiment, mask generator 300 can present the tint planes 306 of the recognized objects 304 to the user by updating the display provided by user interface 200 to display the input image overlaid with graphical elements  
5 indicating recognized objects. The boundaries of one or more recognized objects 304 can be presented in a bold or highlighted manner, as depicted in FIG. 5. Further, the user can be presented with object selection controls 320 enabling the user to highlight or identify one or more recognized objects of interest.

10 As depicted in FIG. 6, object selection control 320-1 is associated with a first wall segment, object selection control 320-2 is associated with a second wall segment, and 320-3 is associated with a coffee table. Each object selection control 320 can enable the use to select the associated recognized object. While mask generator 300 can recognize any number of objects that are depicted in input image 202, only three recognized objects are highlighted in the depiction of FIG.  
15 6 for clarity. User interface 200 can highlight all recognized objects within input image 202, and/or can enable the user to filter based on one or more criteria, such as the size of the recognized object, the type of the recognized object, or whether the recognized object has had a simulated coating applied during the current (or previous) session. Mask generator 300 can wait until receiving user confirmation before the execution of probability mapper 318.

20 FIG. 7 depicts a representation of tint plane mask images 306-1, 306-2, and 306-3 that are associated with the recognized objects depicted in FIG. 6, with input image 202 depicted in

juxtaposition with tint plane mask image 306-1.

FIG. 8 is a flowchart depicting a method 1000 that can be executed to create object masks 302 based on input image 202. At 1002, the input image can be received, and a search for recognized objects can be initiated at 1004. In embodiments, at 1004, recognized objects can be searched for directly, or a first classifier can determine locations of recognized classes, before a series of second classifiers determines locations of the recognized objects.

At 1006, tint planes can be generated for each recognized object. At 1008, the user can, in embodiments, be queried to confirm that the appropriate objects were recognized. If the user indicates that one or more objects were not correctly identified, control can return to 1004 for updated search results. In embodiments, the user can also be presented with one or more deduced insights regarding the depicted scene. For example, the user can be asked to confirm that the input image depicts a specific interior or exterior architectural scene (for example, a kitchen). If the user indicates that the deduced scene was incorrectly identified, that information can be used as input to recognize the classes of objects within a scene.

At 2000, the probability map for each tint plane mask image can be determined, as depicted and described below with respect to FIG. 9. At 1012, object masks 302 representing each recognized object 304 can be output.

FIG. 9 is a flowchart depicting a method 2000 for generating a probability map 308 for each recognized object 304, according to an embodiment. As discussed above, method 2000 can be executed or implemented as part of method 1000 for generating object masks 302.

At 2010, the contiguous outside perimeter boundary, or fence, of the tint plane 306 for a

recognized object 304 can be determined by executing at least one edge detection algorithm and comparing the results at 2012, finding long straight edges at the perimeters at 2014, finding near-vertical and near-horizontal longest edges and corners at 2016, and filling missing edge pixels at 2018. More, fewer, or alternative algorithms can be executed to define the outside perimeter by  
5 embodiments.

At 2012, the edge detection algorithms can comprise edge detection algorithms known in the art such as canny edge detection, holistically-nested edge detection, and/or the like. The perimeter pixels determined by each edge detection algorithm can be compared to determine pixels that are assumed to be near the perimeter of the mask image. Each edge detection algorithm can  
10 be configured based on one or more thresholds. The edge detection thresholds can be predetermined and stored in a memory for use during execution of method 2000, or they can be updated based on feedback received during execution. Long straight lines that form the perimeters of polygonal recognized objects such as architectural surfaces and manufactured objects can also be detected.

15 At 2014, a feature extraction technique such as the Hough transform can be applied to selected sets of edge pixels in order to find long lines near the perimeter of the mask image. These lines can be used at 2014 to find near-vertical and near-horizontal edges and the corners of the perimeters at 2016. At 2018, the perimeter boundary of each tint plane can be completed by best-guess replacement of missing edge pixels.

20 At 2020, the median hue, saturation, and luminosity (HSL) values for each pixel in input image 202 that has a value greater than zero in the probability map for the recognized object 304

can be determined.

At 2030, the median HSL values can be used to fill areas of the probability map 308 using a flood-fill algorithm. The flood-fill algorithm can be a conventional algorithm, or a smart fill algorithm. The perimeter boundary of the tint plane 306 can be used as a fence, such that flood-  
5 fill operations will terminate when they reach a pixel on the perimeter boundary. Flood-fill operations can begin from pixels with a high score in the probability map 308, and the area that is filled can be capped by the strength of each pixel in the probability map 308 as well as other factors. Additional factors that can be used to determine whether to fill a location include the delta  
10 between the location's HSL value and the median HSL value for the tint plane. Further, instead of simply returning a binary value for whether a pixel is or isn't part of a tint plane 306, a weighted value for confidence can be returned. These weighted values can be used to improve the accuracy and fidelity of the rendering operations.

FIG. 10 is a schematic view depicting components of renderer 400, according to an embodiment. Renderer 400 can receive input image 202, object masks 302, and a selected coating  
15 204 to apply to produce painted image, 402. FIG. 11 depicts an example painted image 402, in which regions corresponding to the recognized objects of the example depicted in FIG. 4 have been painted with a first selected coating 204-1, second selected coating 204-2, and third selected coating 204-3.

Returning to FIG. 10, image color mapper 404 can perform a first normalization of the  
20 pixels within the probability map 308 of each object mask 302, to determine a first luminosity adjustment to be made to each pixel of the probability map to normalize the lightness within a

given recognized object. In a second normalization, image color mapper 404 can further determine a second luminosity adjustment to be made between the probability maps 308 of all of the object masks 302.

Image color rotator 406 can use the first and second luminosity adjustments to produce tint  
5 plane adjustment maps 408 for each tint plane 306. Each tint plane 306 can have one or more tint plane adjustment maps 408 defined. A hue map can identify regions of the object that contain the object's primary color (as identified by analysis of each pixel within the tint plane) and/or near-matches of the primary color. A highlight map can identify regions of the object that contain unusually luminous areas relative to the tint plane as a whole. Objects that are identified as being  
10 bright overall (such as an illuminated lamp) may not have a highlight map. A lightness normalization map can contain a lightness-adjusted version of the object bounded by the tint plane that is used as a foundation for applying a selected coating 204. An analysis performed on all tint planes can determine the target lightness value for the normalization of the object in relation to its peers. Each tint plane adjustment map 408 can be Base64 image data such as a binary image file  
15 comprising an scalable vector graphics (SVG) filter primitive, as defined by any version of the W3C Scalable Vector Graphics Specification, though other filter formats can be used.

The tint plane adjustment maps 408 can be combined with the selected coating 204 to create render filter 410, which can be a SVG filter. Those of ordinary skill in the arts will appreciate that multiple SVG filter primitives can be combined to define a composite SVG filter that can be  
20 applied to an input image. In an embodiment, render filter 410 can comprise a masked image, applying the same variations of hue, saturation, and luminosity that were present in input image

202, but applied to the selected coating 204, in place of each object's primary color.

Painter 412 can be a client application configured to receive render filter 410, input image 202 and selected coating 204 to produce painted image 402 by overlaying each pixel of input image 202 with a painted pixel with values determined based on render filter 410 and selected coating  
5 204. Painter 412 can comprise a web browser or other application configured to display on the screen of a user device and one or more client-side elements for interacting with and updating the document object model (DOM) as interpreted by the browser.

In embodiments, painter 412 can utilize the Canvas application programming interface (Canvas API). Canvas elements in HTML documents can provide instructions to a web browser  
10 or other HTML client-side interface to render an image with modifiers, such as render filter 410 and/or selected coating 204 applied. In embodiments, input image 202 can be provided as an ImageBitmap object within a Canvas element, and render filter 410 can be provided as one or more CanvasFilter objects within the canvas element.

FIGS. 12A and 12B are code listings depicting an example HTML markup that can be  
15 produced by painter 412 in one embodiment. As depicted, input image 202 is depicted as an image (img) element at line 1. Render filter 410 is defined at lines 4-15 as a filter element "filter\_0." As depicted, render filter 410 includes selected coating 204 defined as a flat color feFlood element (line 5) and an feBlend element at line 14. Tint plane adjustment maps 408 are depicted as feImage  
20 elements (lines 6 and 7). Painted image 402 is depicted at lines 21-22 as an SVG element filtered by render filter 410, and masked by a mask image "mask\_0" defined based on probability mask 408 such that only the tint plane pixels that meet a threshold confidence value are displayed over

input image 202.

In one embodiment, tint planes 306, probability maps 308, and tint plane adjustment maps 408 can be generated on a server computing system. These can be generated a single time when input image 202 is first uploaded. These artifacts can be stored in a networked data store (such as a database or file system), or in embodiments can be stored on a client in HTML5 web storage. Painter 412 can comprise client-side components (such as JavaScript (React) code) that can generate HTML representing render filter 410 each time the user selects a different coating. The browser or other client-side applicant can then natively repaint the DOM to reflect the updated filters.

10 FIG. 13 depicts a method 3000 for rendering a painted image according to an embodiment. At 3002 the object masks 302, including tint planes 306 and probability maps 308 can be received. At 3004, lightness can be corrected and normalized per object. At 3006, binary images can be created to represent the per-pixel difference from the median HSL for each tint plane. At 3008, the binary images can be added to SVG filter primitives, which can be used at 3010 to render the  
15 painted image 402.

System 100 can be implemented with, and hosted by, a client-server architecture. One such architecture that can be used according to an embodiment is in depicted FIG. 14. Client 102 can comprise a web application, or other application for execution on a computing device such as a person computer, mobile phone, tablet, or other device, and can implement or user interface 200  
20 and house renderer 400. In embodiments, various components of client 102 can be generated or executed on a computing device associated with a user, while other components can be executed

by an application server or other service providing computing system that is remote or separate from the user device. Embodiments of client 102 can be implemented in a version of JavaScript such as ECMAScript 6 (or ES6), as defined by the European Computer Manufacturers Association (Ecma) in the ECMA-262 and ISO/IEC 16262 standards, though other languages can be used.

5 Client 102 can comprise a Flow static type checker, and execute within contexts provided by a Redux.js application container. User interface 200 can be provided, at least in part, via a React.js user interface library using Syntactically Awesome Style Sheets (Sass), with rendering provided through HTML5 Canvas with SVG Compound filters and BabylonJS and WebGL for two and three-dimensional rendering.

10 Server 104 can comprise one or more applications providing data and/or processing services to client 102 for execution on one or more computing systems. Server 104 can implement or house mask generator 300 in embodiments. Server 104 can be implemented using Node.js using a Fastify framework. Mask generator 300 can be implemented using DeepLab to define and execute Tensorflow models. Methods provided by the OpenCV framework can be used to improve  
15 and refine the image mask returned by machine learning models. Server 104 can further comprise a data store, such as a MongoDB, or other database.

While components or the entireties of client 102 and server 104 can be implemented and hosted across separate computing systems, any number of computing systems can be used in embodiments.

20 In one embodiment, the system 100 and/or its components or subsystems can include computing devices, microprocessors, modules and other computer or computing devices, which

can be any programmable device that accepts digital data as input, is configured to process the input according to instructions or algorithms, and provides results as outputs. In one embodiment, computing and other such devices discussed herein can be, comprise, contain or be coupled to a central processing unit (CPU) configured to carry out the instructions of a computer program.

5 Computing and other such devices discussed herein are therefore configured to perform basic arithmetical, logical, and input/output operations.

Computing and other devices discussed herein can include memory. Memory can comprise volatile or non-volatile memory as required by the coupled computing device or processor to not only provide space to execute the instructions or algorithms, but to provide the space to store the  
10 instructions themselves. In one embodiment, volatile memory can include random access memory (RAM), dynamic random access memory (DRAM), or static random access memory (SRAM), for example. In one embodiment, non-volatile memory can include read-only memory, flash memory, ferroelectric RAM, hard disk, floppy disk, magnetic tape, or optical disc storage, for example. The foregoing lists in no way limit the type of memory that can be used, as these embodiments are  
15 given only by way of example and are not intended to limit the scope of the disclosure.

In one embodiment, the system or components thereof can comprise or include various modules or engines, each of which is constructed, programmed, configured, or otherwise adapted to autonomously carry out a function or set of functions. The term “engine” as used herein is defined as a real-world device, component, or arrangement of components implemented using  
20 hardware, such as by an application specific integrated circuit (ASIC) or field programmable gate array (FPGA), for example, or as a combination of hardware and software, such as by a

microprocessor system and a set of program instructions that adapt the engine to implement the particular functionality, which (while being executed) transform the microprocessor system into a special-purpose device. An engine can also be implemented as a combination of the two, with certain functions facilitated by hardware alone, and other functions facilitated by a combination of hardware and software. In certain implementations, at least a portion, and in some cases, all, of an engine can be executed on the processor(s) of one or more computing platforms that are made up of hardware (e.g., one or more processors, data storage devices such as memory or drive storage, input/output facilities such as network interface devices, video devices, keyboard, mouse or touchscreen devices, etc.) that execute an operating system, system programs, and application programs, while also implementing the engine using multitasking, multithreading, distributed (e.g., cluster, peer-peer, cloud, etc.) processing where appropriate, or other such techniques. Accordingly, each engine can be realized in a variety of physically realizable configurations, and should generally not be limited to any particular implementation exemplified herein, unless such limitations are expressly called out. In addition, an engine can itself be composed of more than one sub-engines, each of which can be regarded as an engine in its own right. Moreover, in the embodiments described herein, each of the various engines corresponds to a defined autonomous functionality; however, it should be understood that in other contemplated embodiments, each functionality can be distributed to more than one engine. Likewise, in other contemplated embodiments, multiple defined functionalities may be implemented by a single engine that performs those multiple functions, possibly alongside other functions, or distributed differently among a set of engines than specifically illustrated in the examples herein.

It should be understood that the individual steps used in the methods of the present teachings may be performed in any order and/or simultaneously, as long as the teaching remains operable. Furthermore, it should be understood that the apparatus and methods of the present teachings can include any number, or all, of the described embodiments, as long as the teaching  
5 remains operable.

Various embodiments of systems, devices, and methods have been described herein. These embodiments are given only by way of example and are not intended to limit the scope of the claimed inventions. It should be appreciated, moreover, that the various features of the embodiments that have been described may be combined in various ways to produce numerous  
10 additional embodiments. Moreover, while various materials, dimensions, shapes, configurations and locations, etc. have been described for use with disclosed embodiments, others besides those disclosed may be utilized without exceeding the scope of the claimed inventions.

Persons of ordinary skill in the relevant arts will recognize that embodiments may comprise fewer features than illustrated in any individual embodiment described above. The embodiments  
15 described herein are not meant to be an exhaustive presentation of the ways in which the various features may be combined. Accordingly, the embodiments are not mutually exclusive combinations of features; rather, embodiments can comprise a combination of different individual features selected from different individual embodiments, as understood by persons of ordinary skill in the art. Moreover, elements described with respect to one embodiment can be implemented  
20 in other embodiments even when not described in such embodiments unless otherwise noted. Although a dependent claim may refer in the claims to a specific combination with one or more

other claims, other embodiments can also include a combination of the dependent claim with the subject matter of each other dependent claim or a combination of one or more features with other dependent or independent claims. Such combinations are proposed herein unless it is stated that a specific combination is not intended. Furthermore, it is intended also to include features of a claim  
5 in any other independent claim even if this claim is not directly made dependent to the independent claim.

Moreover, reference in the specification to “one embodiment,” “an embodiment,” or “some embodiments” means that a particular feature, structure, or characteristic, described in connection with the embodiment, is included in at least one embodiment of the teaching. The appearances of  
10 the phrase “in one embodiment” in various places in the specification are not necessarily all referring to the same embodiment.

Any incorporation by reference of documents above is limited such that no subject matter is incorporated that is contrary to the explicit disclosure herein. Any incorporation by reference of documents above is further limited such that no claims included in the documents are  
15 incorporated by reference herein. Any incorporation by reference of documents above is yet further limited such that any definitions provided in the documents are not incorporated by reference herein unless expressly included herein.

For purposes of interpreting the claims, it is expressly intended that the provisions of Section 112, sixth paragraph of 35 U.S.C. are not to be invoked unless the specific terms “means  
20 for” or “step for” are recited in a claim.

## CLAIMS

What is claimed is:

1. An image augmentation system for simulating the appearance of a coating to a surface of an original image comprising a plurality of original pixels, the system comprising:

5 at least one memory and at least one processor configured to implement:

a mask generator configured to:

receive the original image, perform a search to detect a recognized object depicted in the original image, and

generate a object mask for the recognized object comprising a mask pixel

10 corresponding to each of the plurality of original pixels, at least one parameter of each mask pixel having a first value if the corresponding original pixel is part of the recognized object and a second value if the corresponding original pixel is not part of the recognized object;

15 a user interface configured to receive an identification of a selected object and an identification of a selected coating to be simulated on the selected object, and display a painted image; and

a renderer configured to receive the object mask for the selected object and the selected coating and to generate the painted image, each pixel of the painted  
20 image having a painted color, the painted color determined to be the same color as the corresponding pixel of the original image if the corresponding

mask pixel has the second value, and determined based on the coating and the corresponding original pixel if the corresponding mask pixel has the first value.

- 5 2. The system of claim 1, wherein the mask generator comprises a class recognizer configured to identify a class of object present in an area of the original image.
3. The system of claim 2, wherein the mask generator further comprises an image segmenter for each of a plurality of classes, the image segmenter configured to detect a recognized object by  
10 identifying the pixels of the original image that include the recognized object.
4. The system of claim 3, wherein the class recognizer and the image segmenter each comprise a neural network.
- 15 5. The system of claim 1, wherein the painted color is further determined based on at least one tint plane adjustment map, each tint plane adjustment map defining a difference between a base color and the color of the corresponding original pixel.
6. The system of claim 5, wherein the difference between the base color and the color of the  
20 corresponding original pixel is based on at least one of a hue, a saturation, and/or a luminosity of the base color and the color of the corresponding original pixel.

7. The system of claim 5, wherein the coating and the at least one tint plane adjustment map comprise filter primitives, and wherein the renderer is configured to generate the painted image by applying the filter primitives to the original image.

5

8. A method for simulating the appearance of a coating to a surface of an original image comprising a plurality of original pixels, the method comprising:

receiving an original image file;

performing a search to detect a recognized object depicted in the original image;

10 generating an object mask for the recognized object comprising a mask pixel corresponding to each of the plurality of original pixels, at least one parameter of each mask pixel having a first value if the corresponding original pixel is part of the recognized object and a second value if the corresponding original pixel is not part of the recognized object;

15 displaying, on a user interface device, a first painted image depicting the original image and a selectable area corresponding to each recognized object;

receiving, from the user interface device, a user selection of a selected object and a coating;  
and

rendering a painted image, each pixel of the painted image having a painted color:

20 determined to be the same color as the corresponding pixel of the original image if the corresponding mask pixel has the second value, and

determined based on the coating and the corresponding original pixel if the corresponding mask pixel has the first value.

9. The method of claim 8, wherein performing the search to detect a recognized object in the original image comprises executing a classifier configured identify a class of object present in an area of the original image.

10. The method of claim 9, wherein performing the search to detect a recognized object in the original image further comprises executing a classifier configured to detect a recognized object of an associated class by identifying the pixels of original image that include the recognized object.

11. The method of claim 10, where wherein the classifier configured to identify a class of object present in an area of the original image and the classifier configured to detect a recognized object of an associated class each comprise a neural network.

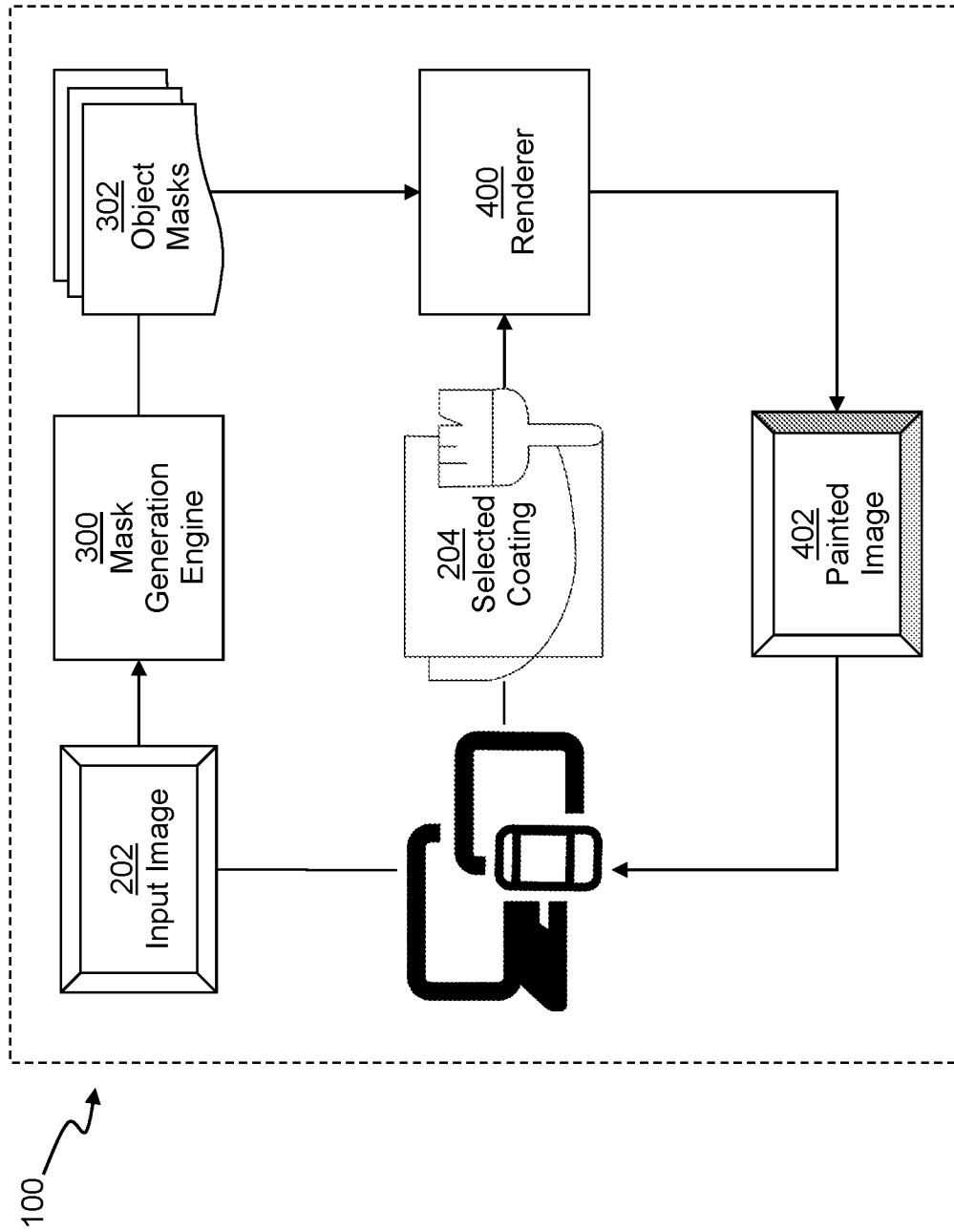
15

12. The method of claim 8, wherein the painted color is further determined based on at least one tint plane adjustment map, each tint plane adjustment map defining a difference between a base color and the color of the corresponding original pixel.

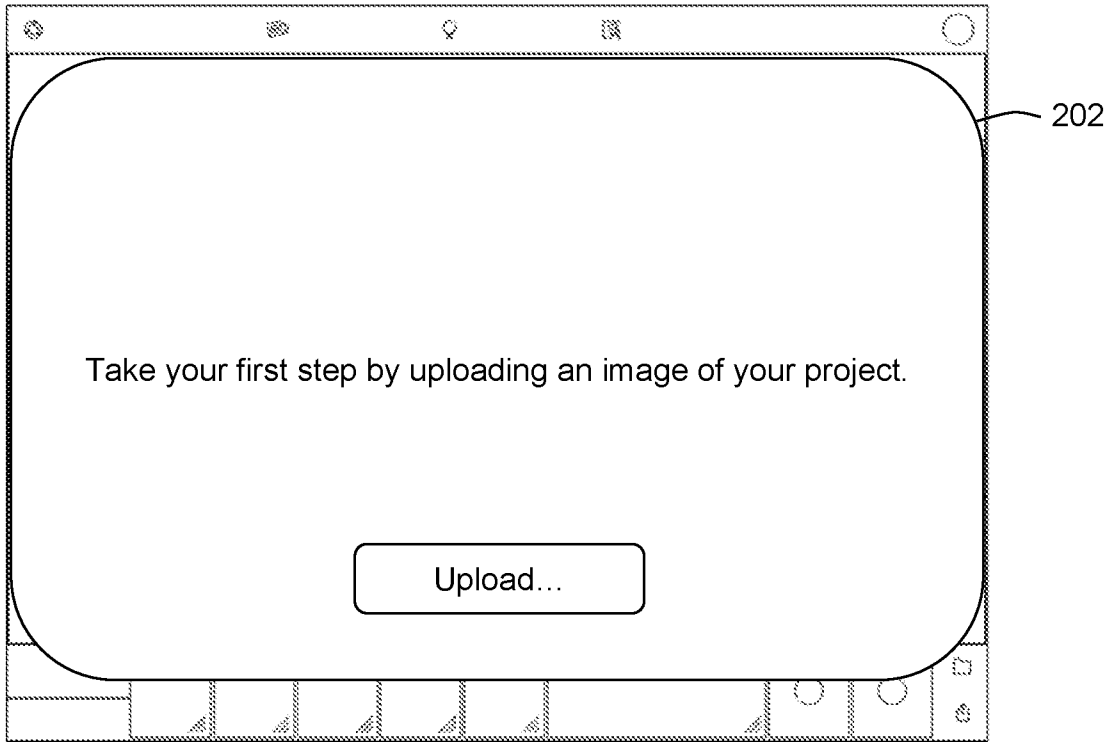
20 13. The method of claim 12, wherein the difference between the base color and the color of the corresponding original pixel is based on at least one of a hue, a saturation, and/or a luminosity

of the base color and the color of the corresponding original pixel.

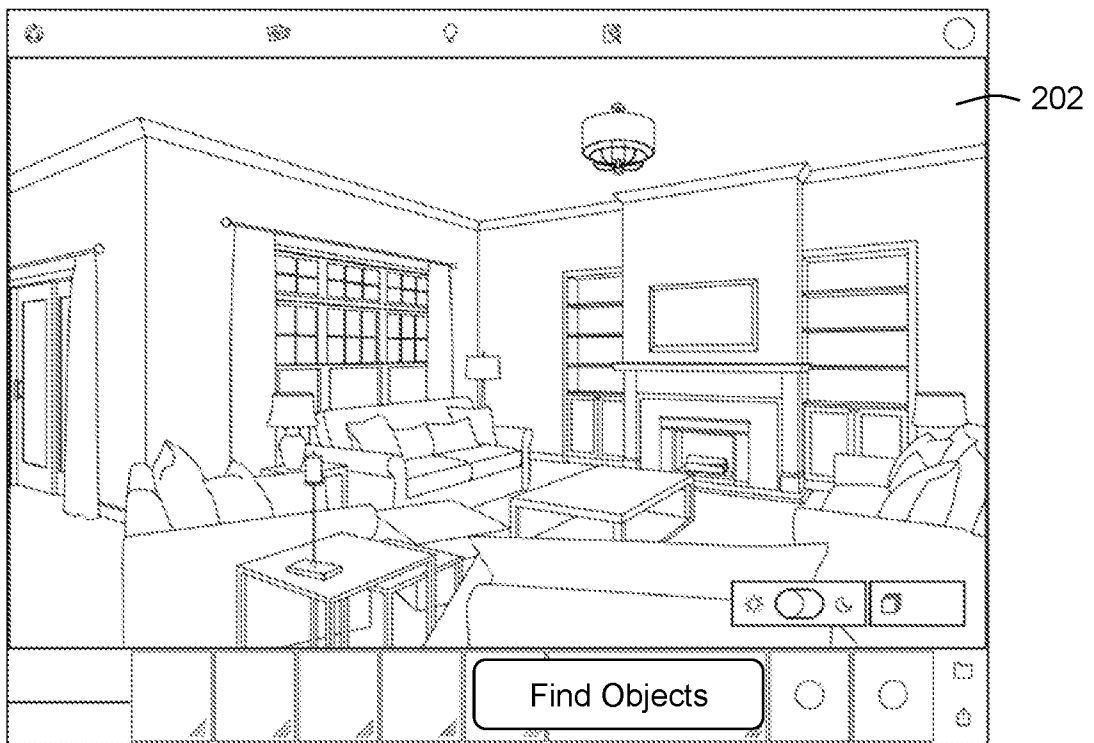
14. The method of claim 13 wherein the coating and the tint plane adjustment map comprise filter primitives, and wherein rendering the painted image comprises by applying the filter  
5 primitives to the original image.



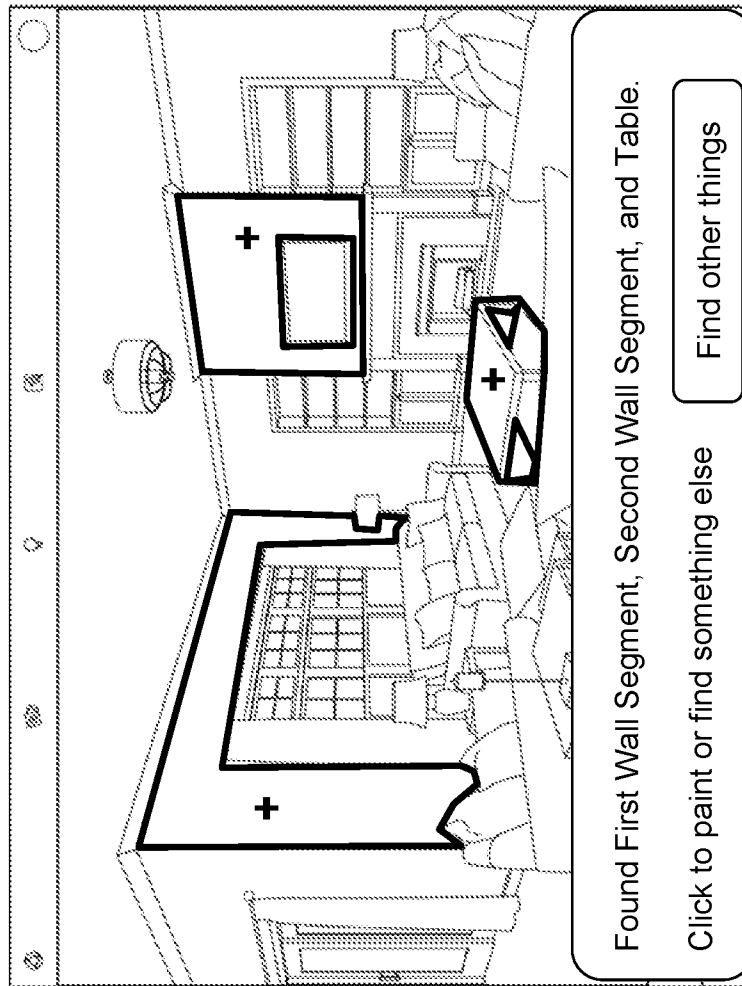
**FIG. 1**



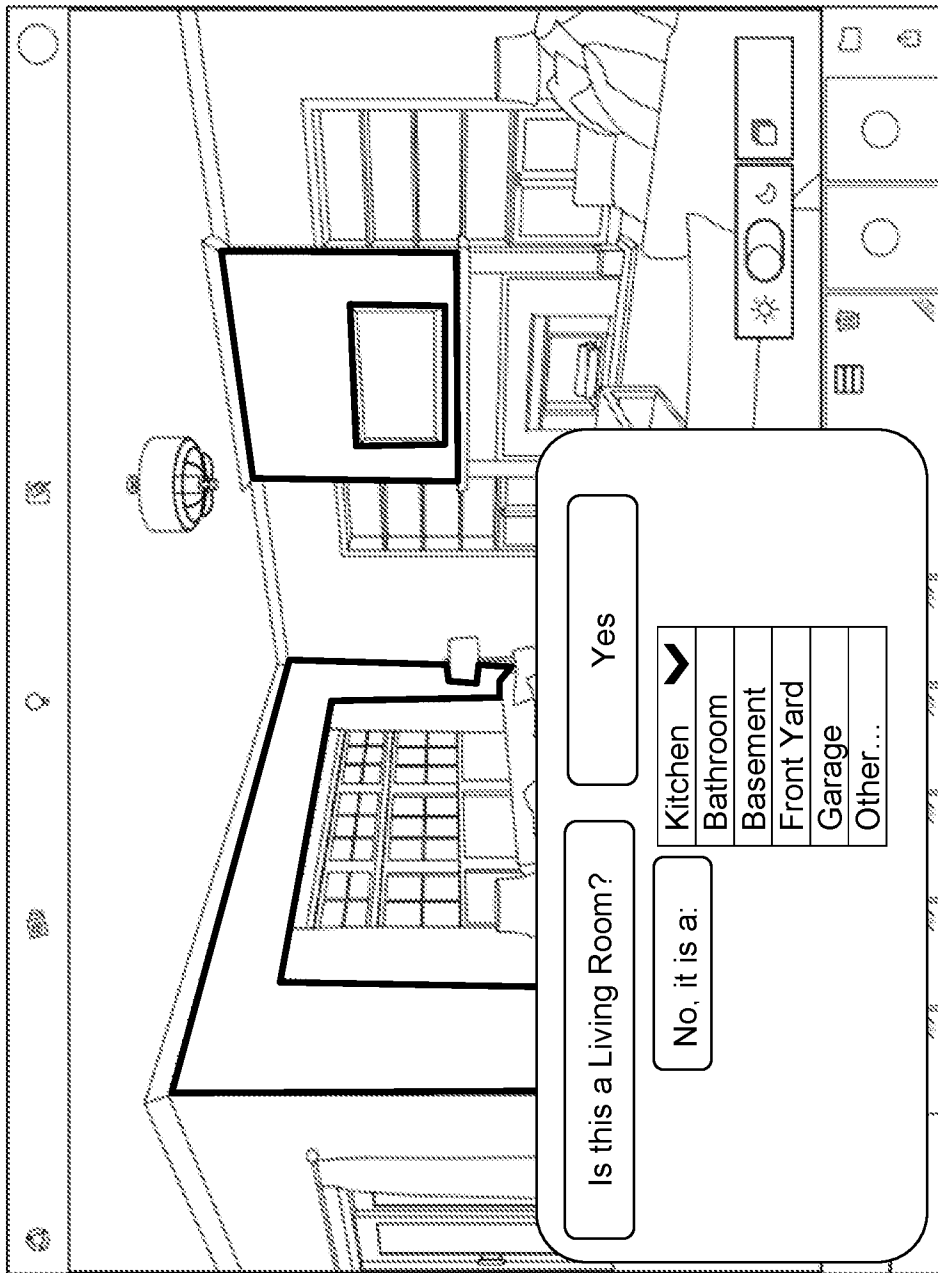
**FIG. 2A**



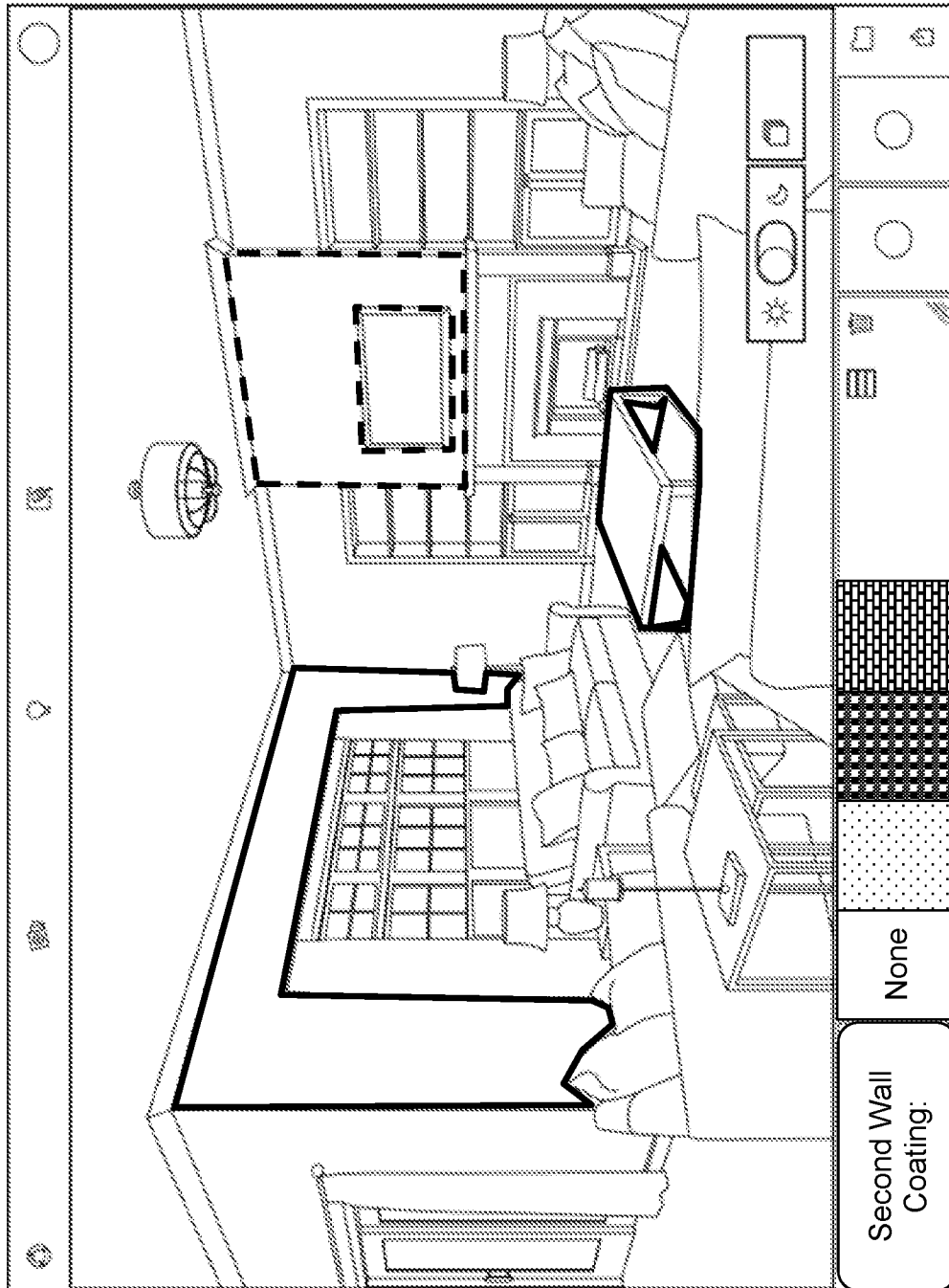
**FIG. 2B**



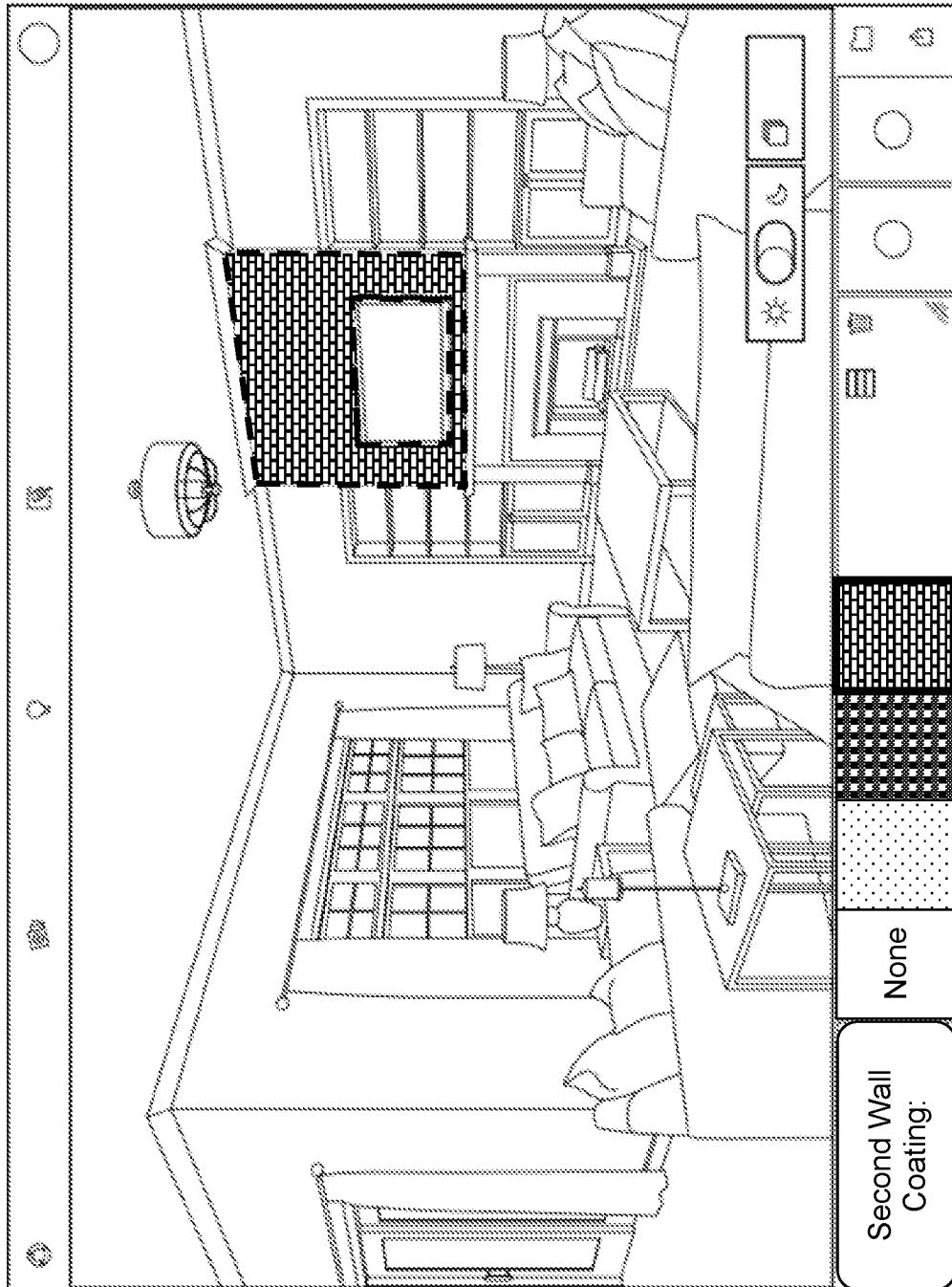
**FIG. 2C**



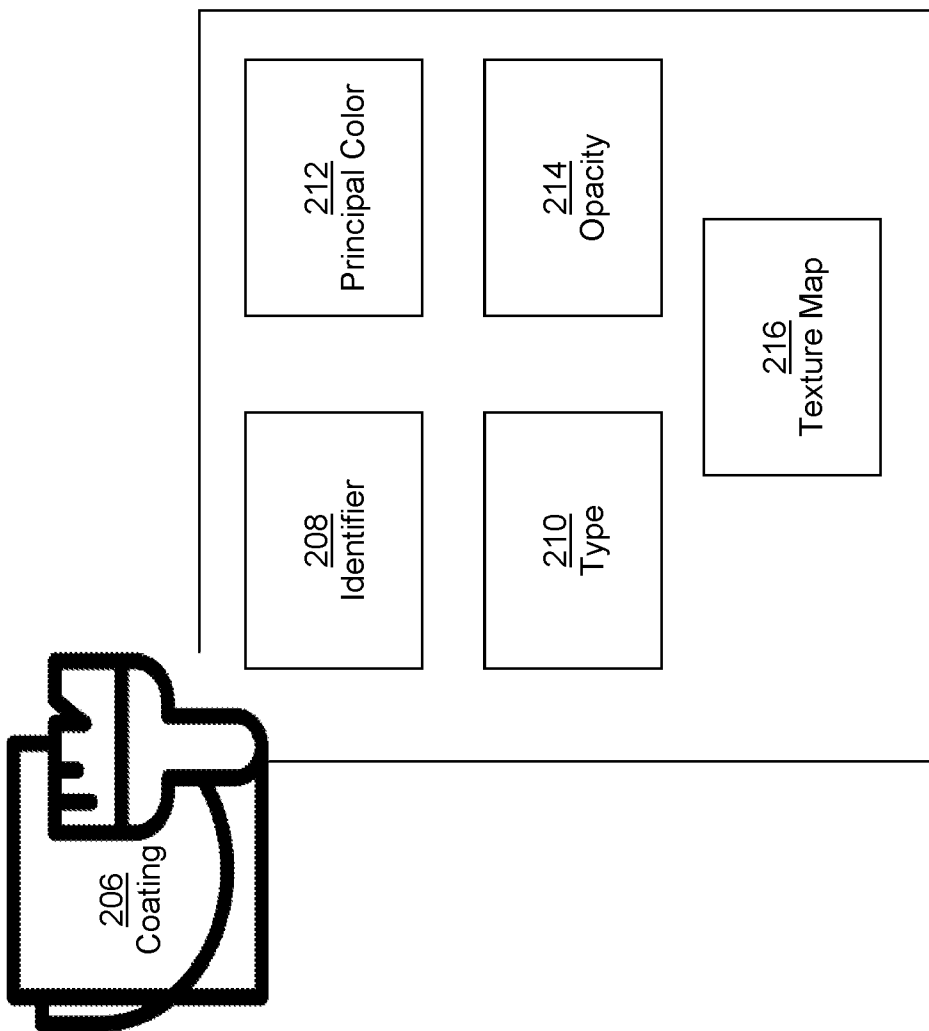
**FIG. 2D**



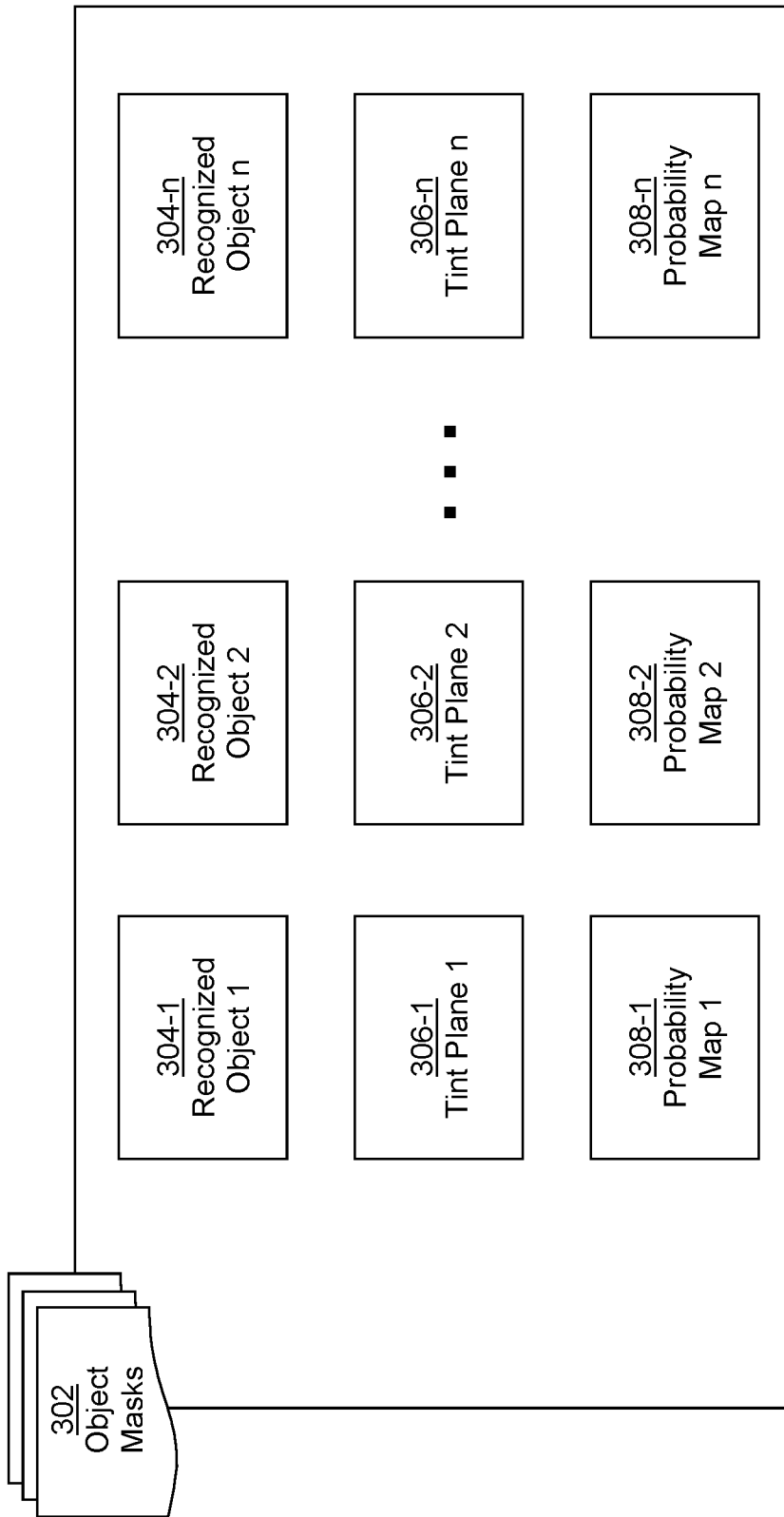
**FIG. 2E**



**FIG. 2F**



**FIG. 3**



**FIG. 4**

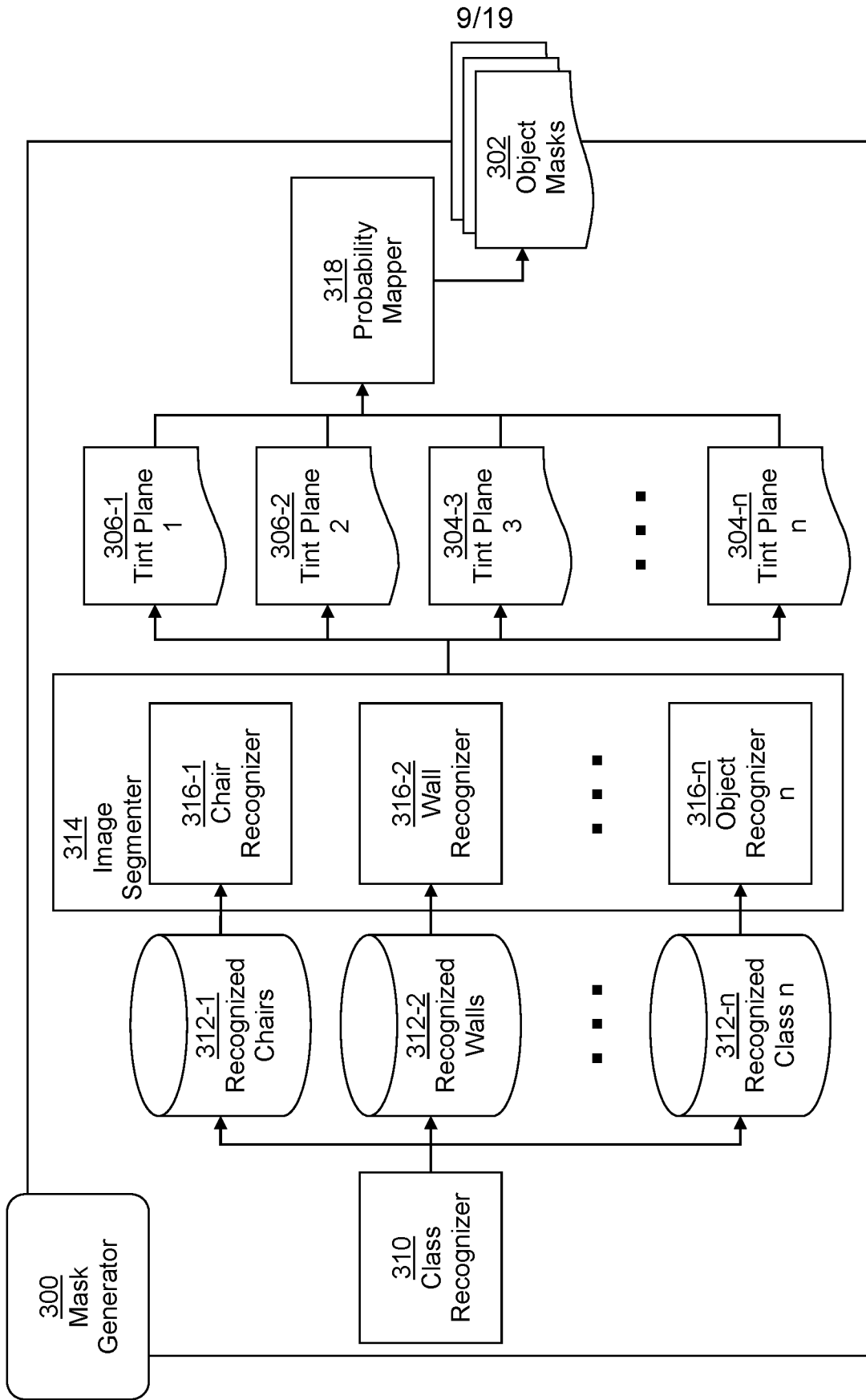
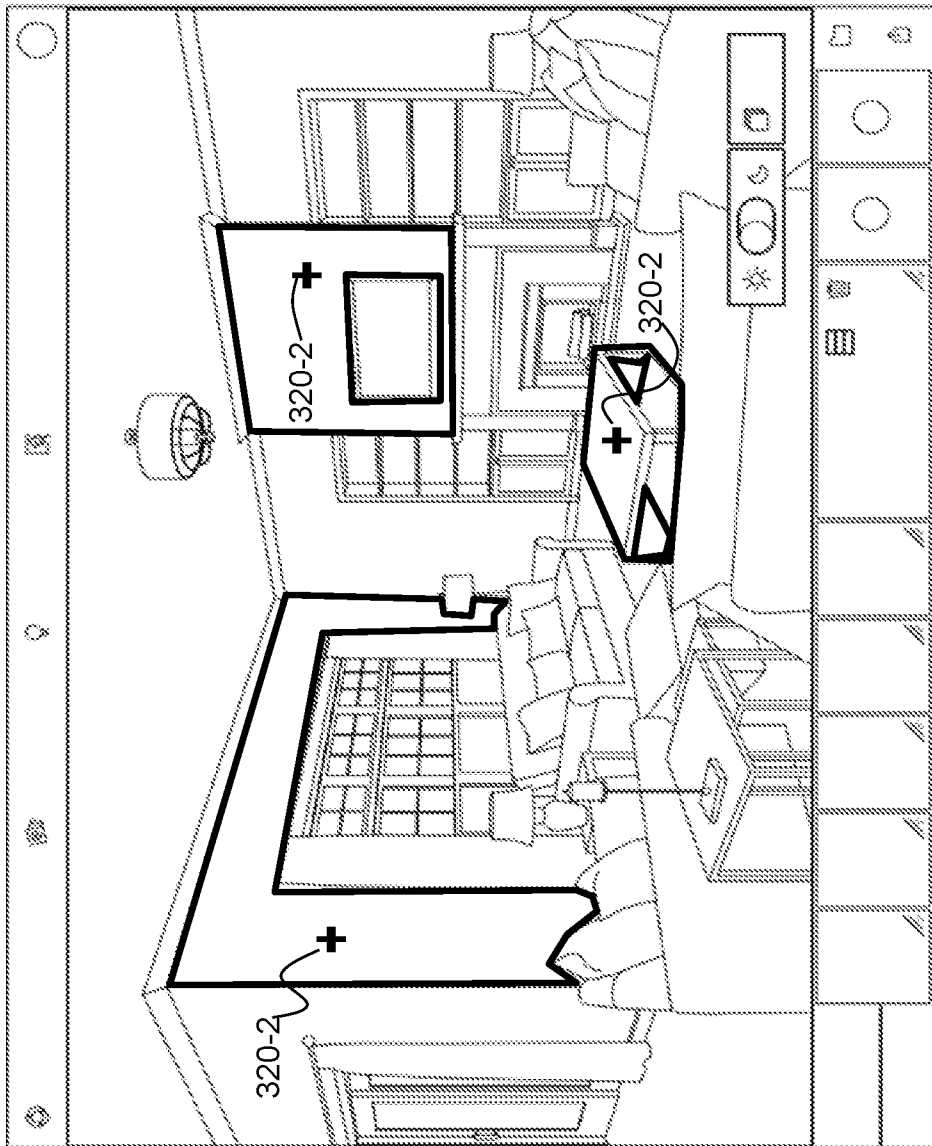
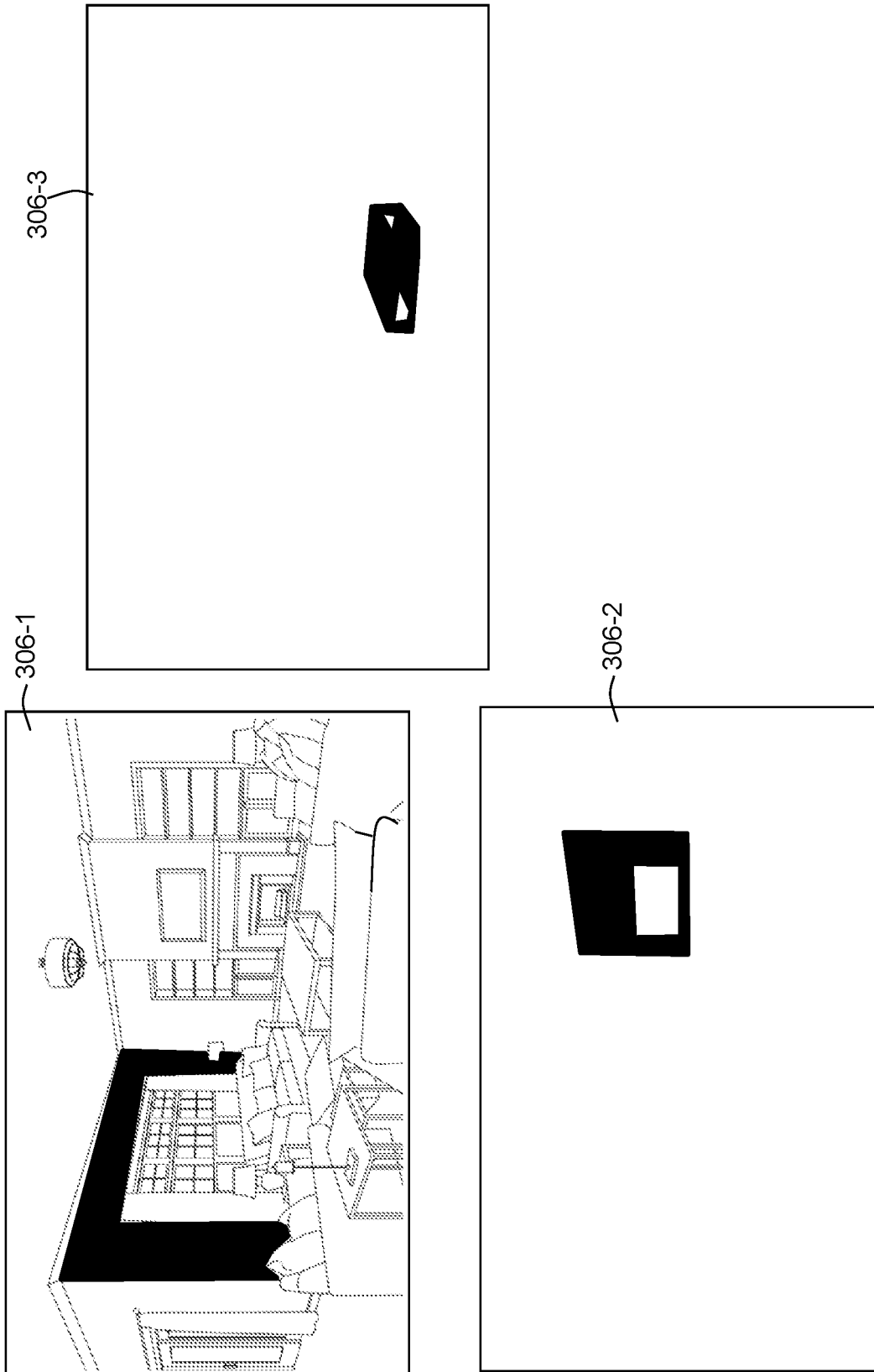


FIG. 5

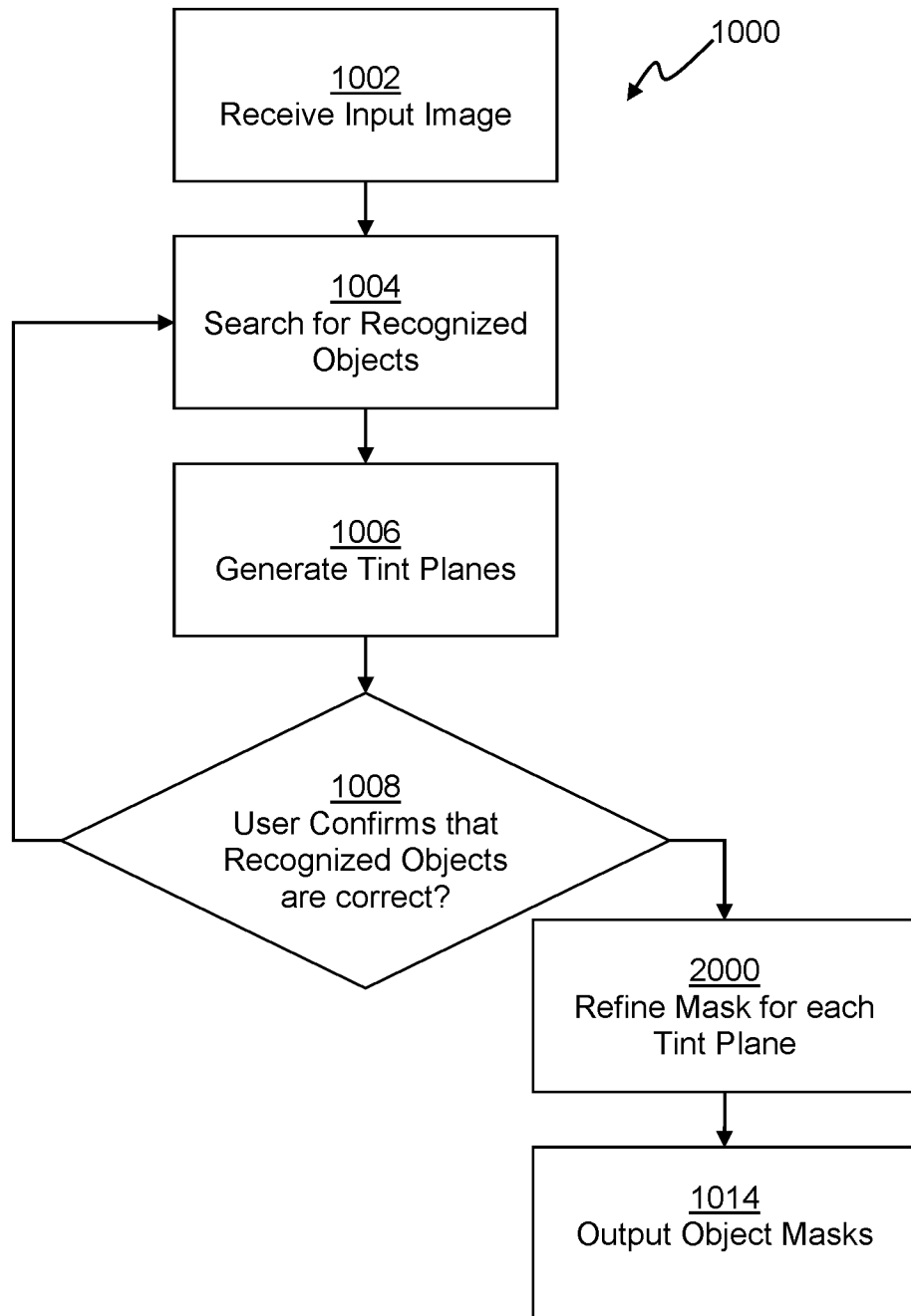


**FIG. 6**



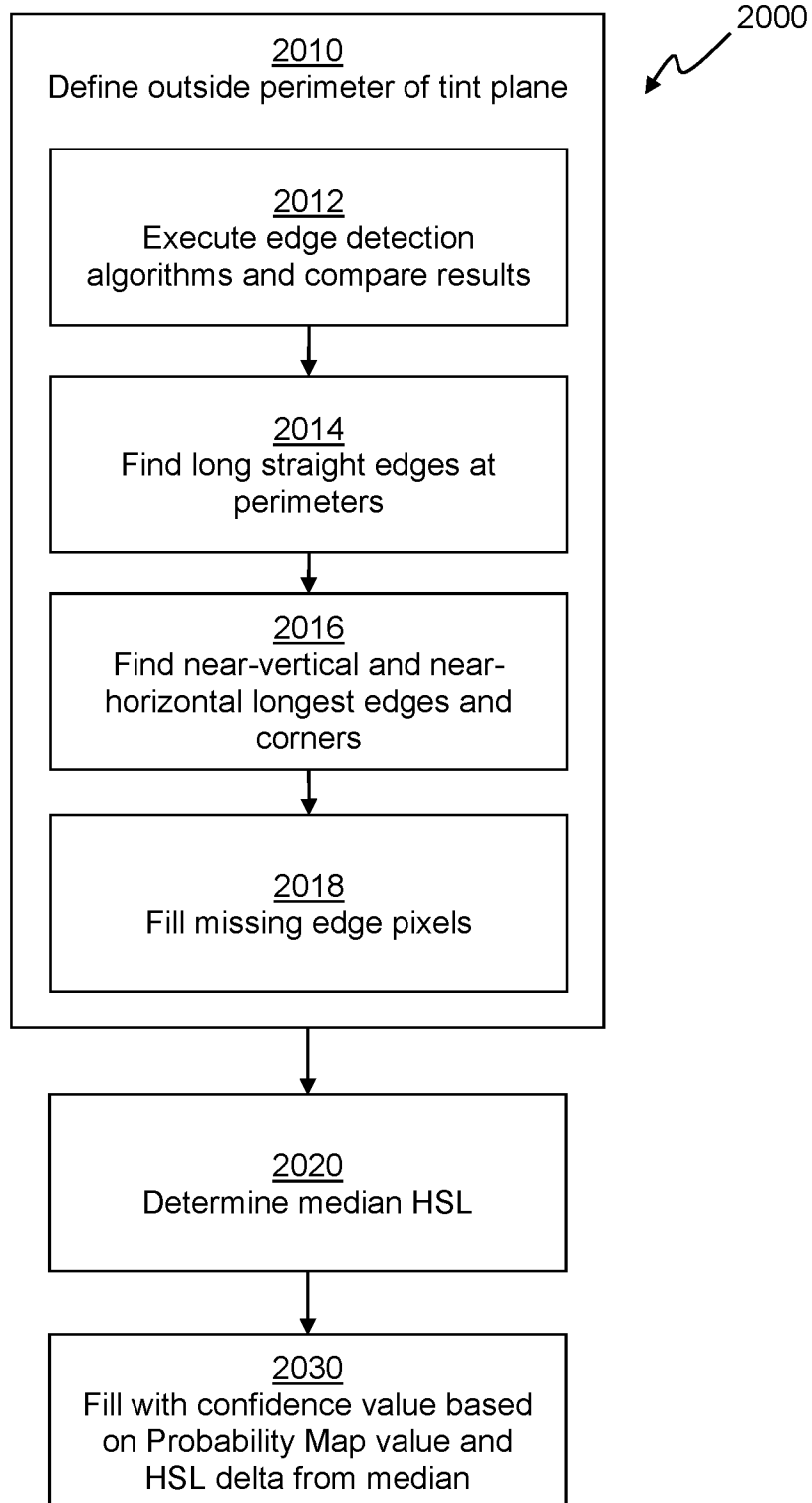
**FIG. 7**

12/19

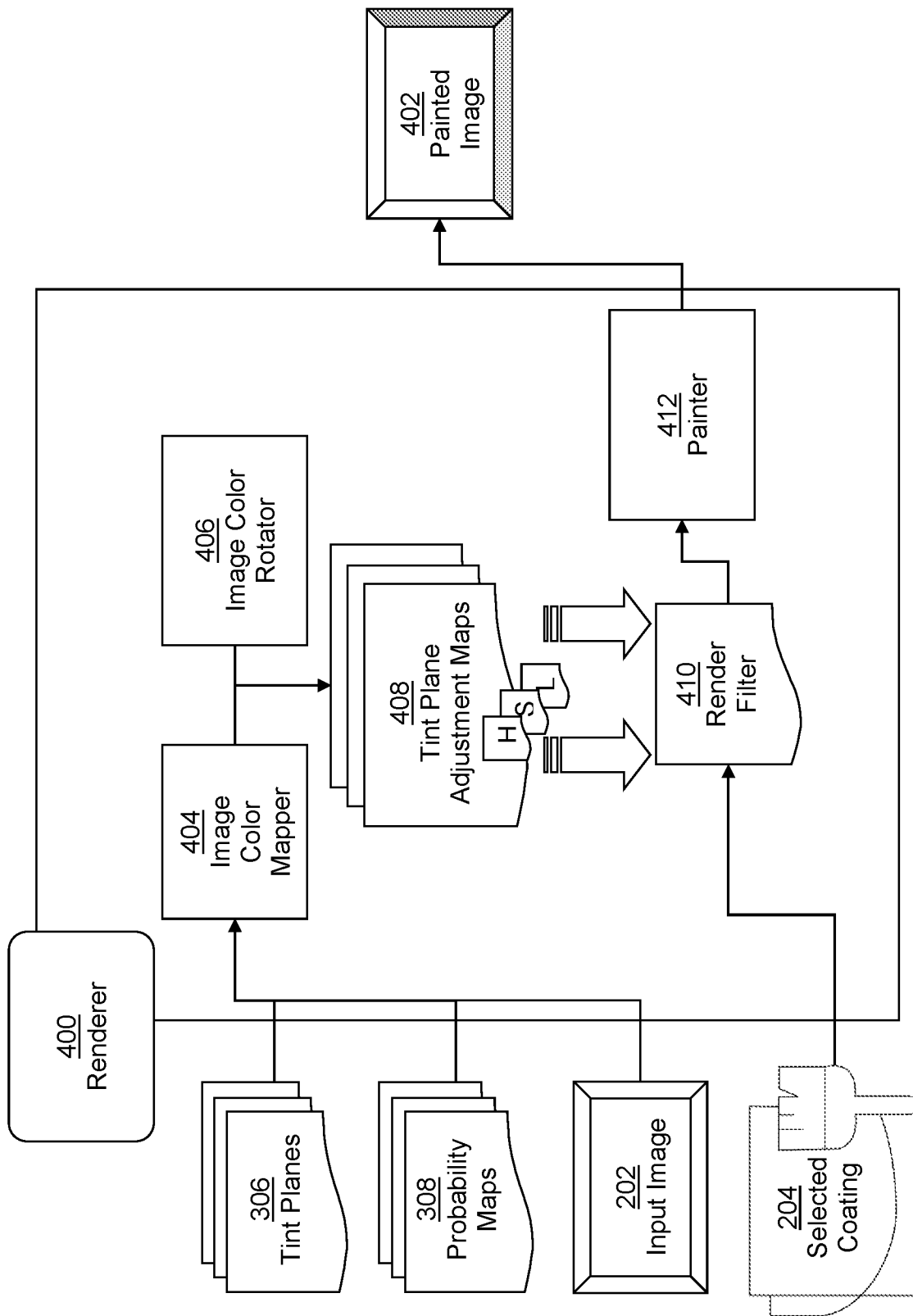


**FIG. 8**

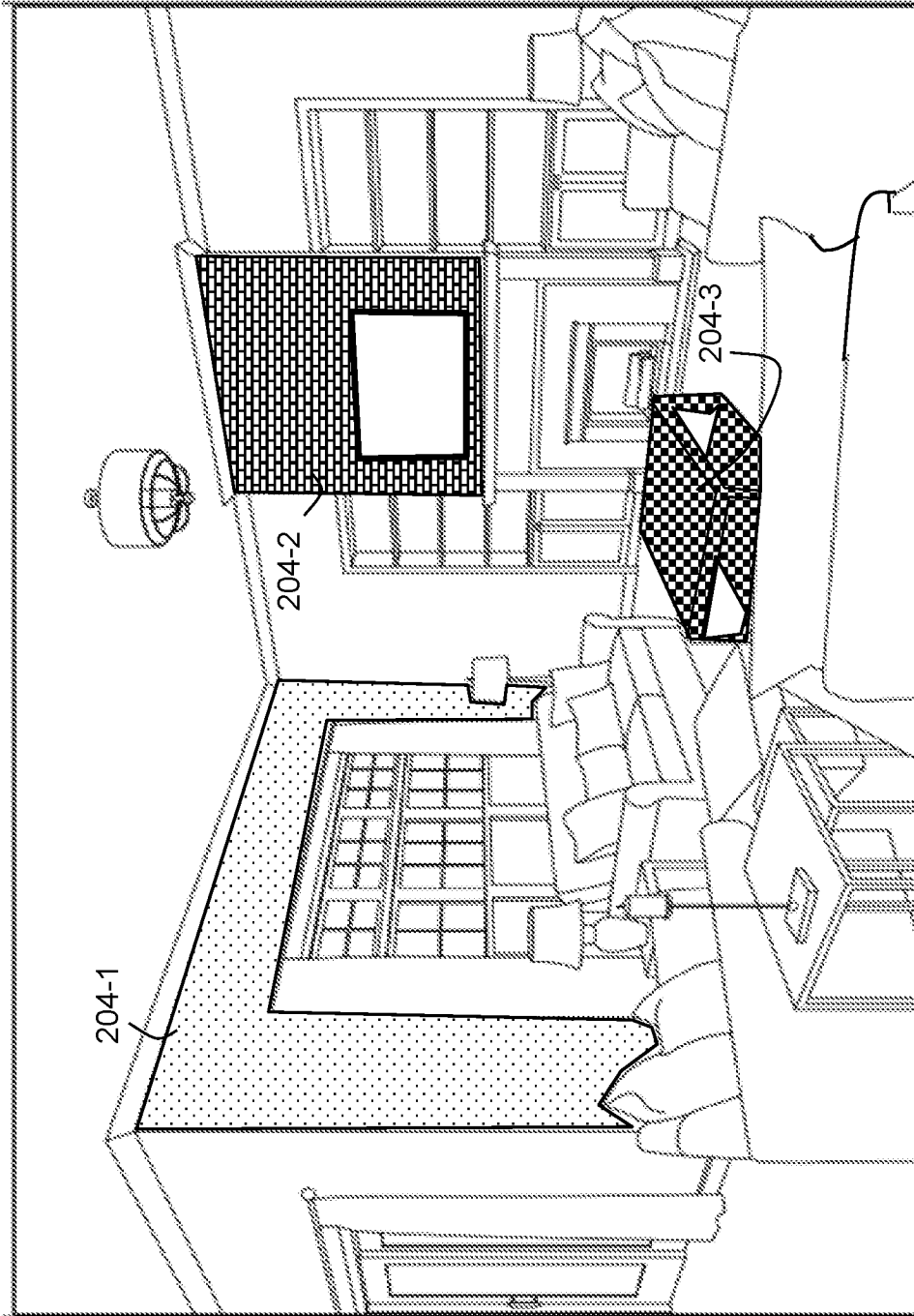
13/19



**FIG. 9**



**FIG. 10**



**FIG. 11**

```

1: 
2: <svg class="svg-defs-wrapper" x="0" y="0" width="100%" height="100%" version="1.1"
  xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink viewBox="0
  0 2400 1600">
3: <defs>
4:   <filter id="filter_0" x="0" y="0" width="100%" height="100%"
  filterUnits="objectBoundingBox" primitiveUnits="objectBoundingBox" color-
  interpolation-filters="sRGB">
5:     <feFlood flood-color="#7dc1cb" result="selectedCoating"></feFlood>
6:     <feImage xlink:href="data:image/png;base64,iVB... " x="0" y="0" width="100%"
  height="100%" result="lightenedImg"></feImage>
7:     <feImage xlink:href="blob:http://localhost:8080/inputimage" x="0" y="0"
  width="100%" height="100%" result="roomImageZero"></feImage>
8:     <feComposite in="lightenedImage" in2="roomImageZero" operator="over" x="0%" y="0%"
  width="100%" height="100%" result="roomImageZero"></feComposite>
9:     <feComposite in="roomImageZero" in2="roomImageZero" operator="over" x="0%" y="0%"
  width="100%" height="100%" result="roomImage"></feComposite>
10:    <feImage xlink:href="data:image/png;base64,iVBORw0KNSUhEJgAABL..." x="0" y="0"
  width="100%" height="100%" result="desaturationMask"></feImage>
11:    <feColorMatrix in="roomImageZero" type="saturate" values="0"
  result="roomImage"></feColorMatrix>
12:    <feComposite in="roomImage" in2="desaturationMask" operator="in" x="0%" y="0%"
  width="100%" height="100%" result="desatRoomParts"></feComposite>
13:    <feComposite in="desatRoomParts" in2="roomImageZero" operator="over" x="0%" y="0%"
  width="100%" height="100%" result="roomImage"></feComposite>

```

cont'd in FIG. 12B...

**FIG. 12A**

17/19

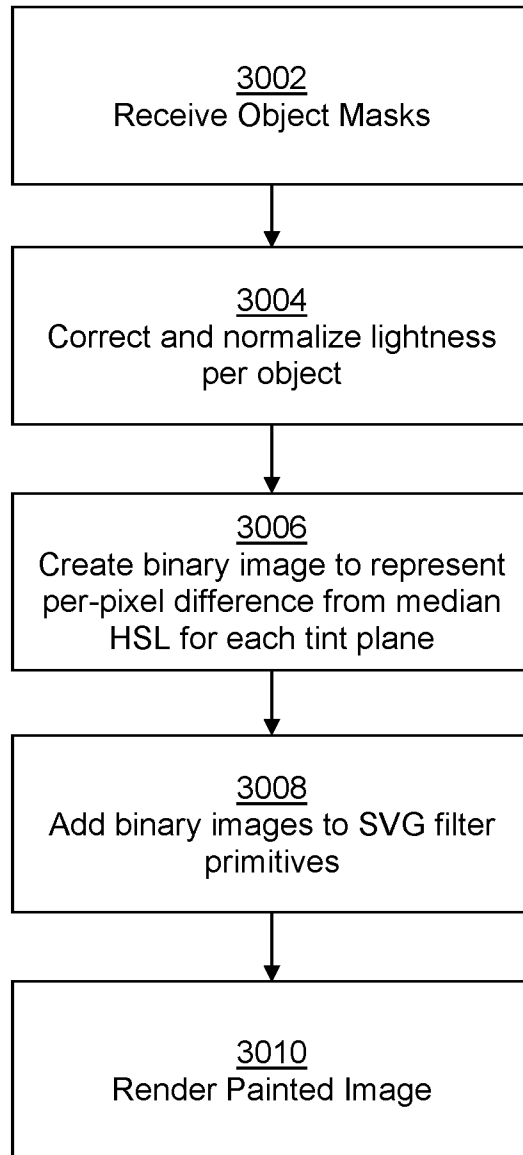
... cont'd from FIG. 12A

```
14:   <feBlend mode="multiply" in="selectedCoating" in2="roomImage"
      result="roomImage"></feBlend>
15: </filter>
16: <mask id="mask_0" x="0" y="0" width="100%" height="100%"
      maskUnits="objectBoundingBox">
17:   <image x="0" y="0" width="100%" height="100%"
        xlink:href="http://localhost/static/files/objectmasks.png"> </image>
18: </mask>
19: </defs>
20: </svg>
21: <svg class="image-tinted" version="1.1" xmlns="http://www.w3.org/2000/svg"
      viewBox="0 0 2400 1600" preserveAspectRatio="none">
22:   <rect fill="rgba(0,0,0,0)" x="0" y="0" width="100%" height="100%"
        mask="url(#mask_0)" filter="url(#filter_0)"></rect>
22: </svg>
```

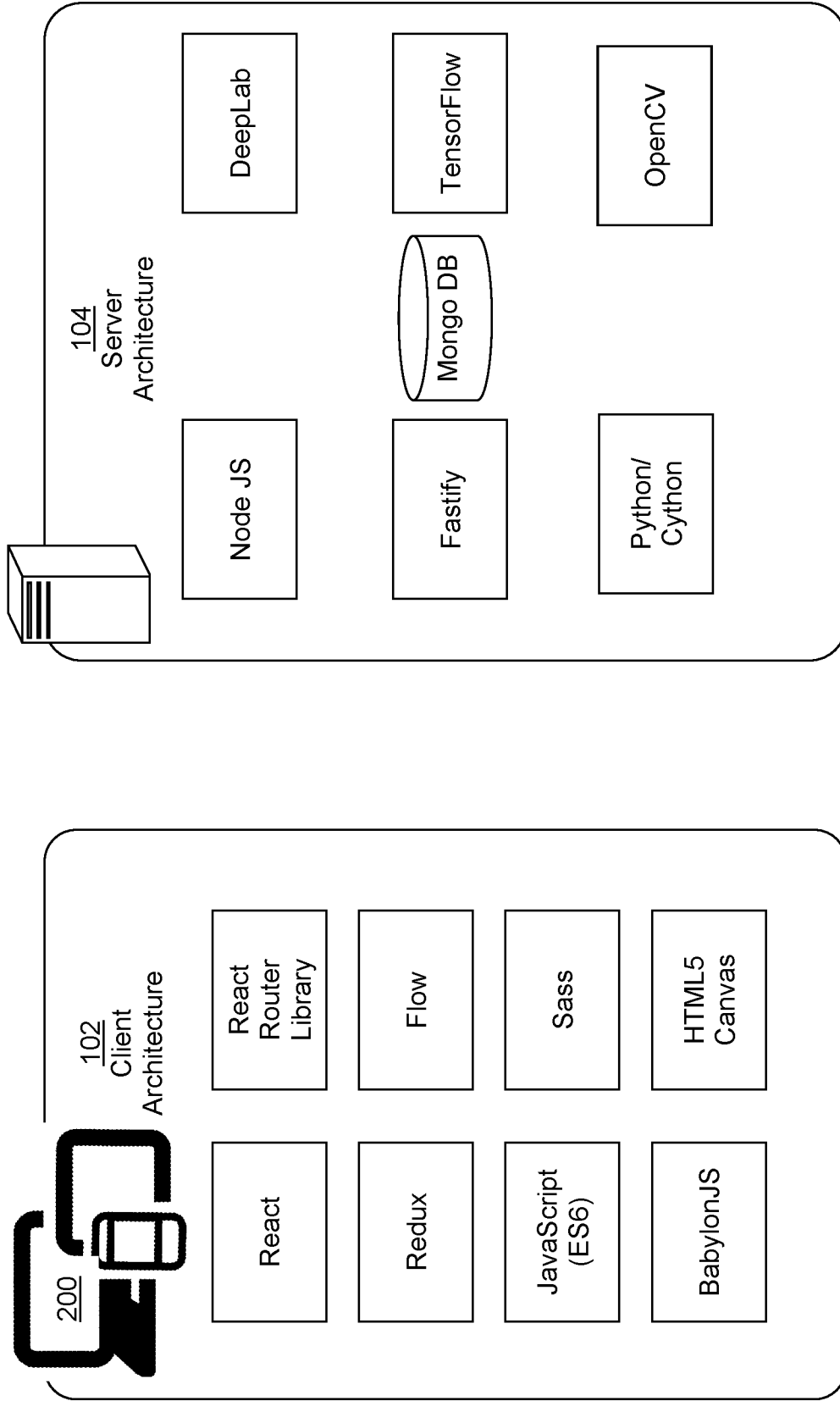
**FIG. 12B**

18/19

3000 ↙



**FIG. 13**



**FIG. 14**

**INTERNATIONAL SEARCH REPORT**

International application No.

PCT/US2020/054939

<p><b>A. CLASSIFICATION OF SUBJECT MATTER</b>                  IPC (20210101) G06T 19/00, G06K 9/00, G06K 9/46                  CPC (20161101) G06T 19/006, G06T 2207/20221, G06K 9/00228, G06K 9/4652, G06K 9/00624                  According to International Patent Classification (IPC) or to both national classification and IPC</p>																	
<p><b>B. FIELDS SEARCHED</b></p> <p>Minimum documentation searched (classification system followed by classification symbols)                  IPC (20210101) G06T 19/00, G06K 9/00, G06K 9/46                  CPC (20161101) G06T 19/006, G06T 2207/20221, G06K 9/00228, G06K 9/4652, G06K 9/00624</p> <p>Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched</p> <p>Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)                  Databases consulted: Esp@cenet, Google Patents, Orbit, SIMILARI                  Search terms used: pixel, mask, recogni+, detect+, object,overlay, mask</p>																	
<p><b>C. DOCUMENTS CONSIDERED TO BE RELEVANT</b></p> <table border="1" style="width:100%; border-collapse: collapse;"> <thead> <tr> <th style="width:10%;">Category*</th> <th style="width:70%;">Citation of document, with indication, where appropriate, of the relevant passages</th> <th style="width:20%;">Relevant to claim No.</th> </tr> </thead> <tbody> <tr> <td>Y</td> <td>EP 3423990 A1 HOLITION LTD 09 Jan 2019 (2019/01/09) ¶¶6, 20, 76, 77, 86, 92, 205, 206, 209, 210, processor 1004, main memory 1008 and secondary memory 1010</td> <td>1-14</td> </tr> <tr> <td>Y</td> <td>US 2019035125 A1 ROCKWELL COLLINS INC 31 Jan 2019 (2019/01/31) ¶57</td> <td>1-14</td> </tr> <tr> <td>A</td> <td>EP 3479344 A1 FITTINGBOX 08 May 2019 (2019/05/08) Entire document</td> <td>1-14</td> </tr> <tr> <td>A</td> <td>EP 2512141 A1 SONY COMP ENTERTAINMENT EUROPE 17 Oct 2012 (2012/10/17) Entire document</td> <td>1-14</td> </tr> </tbody> </table>			Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.	Y	EP 3423990 A1 HOLITION LTD 09 Jan 2019 (2019/01/09) ¶¶6, 20, 76, 77, 86, 92, 205, 206, 209, 210, processor 1004, main memory 1008 and secondary memory 1010	1-14	Y	US 2019035125 A1 ROCKWELL COLLINS INC 31 Jan 2019 (2019/01/31) ¶57	1-14	A	EP 3479344 A1 FITTINGBOX 08 May 2019 (2019/05/08) Entire document	1-14	A	EP 2512141 A1 SONY COMP ENTERTAINMENT EUROPE 17 Oct 2012 (2012/10/17) Entire document	1-14
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.															
Y	EP 3423990 A1 HOLITION LTD 09 Jan 2019 (2019/01/09) ¶¶6, 20, 76, 77, 86, 92, 205, 206, 209, 210, processor 1004, main memory 1008 and secondary memory 1010	1-14															
Y	US 2019035125 A1 ROCKWELL COLLINS INC 31 Jan 2019 (2019/01/31) ¶57	1-14															
A	EP 3479344 A1 FITTINGBOX 08 May 2019 (2019/05/08) Entire document	1-14															
A	EP 2512141 A1 SONY COMP ENTERTAINMENT EUROPE 17 Oct 2012 (2012/10/17) Entire document	1-14															
<p><input type="checkbox"/> Further documents are listed in the continuation of Box C.      <input checked="" type="checkbox"/> See patent family annex.</p>																	
<p>* Special categories of cited documents:</p> <p>“A” document defining the general state of the art which is not considered to be of particular relevance      “T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</p> <p>“D” document cited by the applicant in the international application      “X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone</p> <p>“E” earlier application or patent but published on or after the international filing date      “Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art</p> <p>“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</p> <p>“O” document referring to an oral disclosure, use, exhibition or other means      “&amp;” document member of the same patent family</p> <p>“P” document published prior to the international filing date but later than the priority date claimed</p>																	
<p>Date of the actual completion of the international search 02 Feb 2021</p>		<p>Date of mailing of the international search report 02 Feb 2021</p>															
<p>Name and mailing address of the ISA: Israel Patent Office Technology Park, Bldg.5, Malcha, Jerusalem, 9695101, Israel Email address: pctoffice@justice.gov.il</p>		<p>Authorized officer BARACH Chev  Telephone No. 972-73-3927232</p>															

**INTERNATIONAL SEARCH REPORT**  
Information on patent family members

International application No.  
PCT/US2020/054939

Patent document cited search report			Publication date	Patent family member(s)			Publication Date
EP	3423990	A1	09 Jan 2019	EP	3423990	A1	09 Jan 2019
-----							
				GB	201603662	D0	13 Apr 2016
				GB	2548087	A	13 Sep 2017
				GB	201603665	D0	13 Apr 2016
				GB	2548088	A	13 Sep 2017
				GB	201608424	D0	29 Jun 2016
				GB	2550344	A	22 Nov 2017
				GB	2550344	B	03 Jun 2020
				US	2019122404	A1	25 Apr 2019
				WO	2017149315	A1	08 Sep 2017
US	2019035125	A1	31 Jan 2019	US	2019035125	A1	31 Jan 2019
-----							
				US	10269159	B2	23 Apr 2019
				CN	109345556	A	15 Feb 2019
				EP	3435284	A1	30 Jan 2019
EP	3479344	A1	08 May 2019	EP	3479344	A1	08 May 2019
-----							
				EP	3479344	B1	09 Dec 2020
				CN	109983501	A	05 Jul 2019
				FR	3053509	A1	05 Jan 2018
				FR	3053509	B1	16 Aug 2019
				JP	2019527410	A	26 Sep 2019
				KR	20190021390	A	05 Mar 2019
				US	2018005448	A1	04 Jan 2018
				US	9892561	B2	13 Feb 2018
				WO	2018002533	A1	04 Jan 2018
EP	2512141	A1	17 Oct 2012	EP	2512141	A1	17 Oct 2012
-----							
				EP	2512141	B1	17 Jul 2019
				US	2012262486	A1	18 Oct 2012

**INTERNATIONAL SEARCH REPORT**  
Information on patent family members

International application No.  
PCT/US2020/054939

Patent document cited search report	Publication date	Patent family member(s)	Publication Date
		US 8941687 B2	27 Jan 2015