



US009270899B1

(12) **United States Patent**
Ivanchenko

(10) **Patent No.:** **US 9,270,899 B1**
(45) **Date of Patent:** **Feb. 23, 2016**

(54) **SEGMENTATION APPROACHES FOR OBJECT RECOGNITION**

(75) Inventor: **Volodymyr V. Ivanchenko**, Mountain View, CA (US)

(73) Assignee: **Amazon Technologies, Inc.**, Reno, NV (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 372 days.

(21) Appl. No.: **13/534,509**

(22) Filed: **Jun. 27, 2012**

(51) **Int. Cl.**
H04N 5/228 (2006.01)
H04N 5/262 (2006.01)

(52) **U.S. Cl.**
CPC *H04N 5/262* (2013.01)

(58) **Field of Classification Search**
CPC H04N 5/262
USPC 348/208, 14, 169
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

2007/0024710	A1*	2/2007	Nakamura	348/143
2009/0284613	A1*	11/2009	Kim	348/222.1
2010/0290710	A1*	11/2010	Gagvani et al.	382/224

* cited by examiner

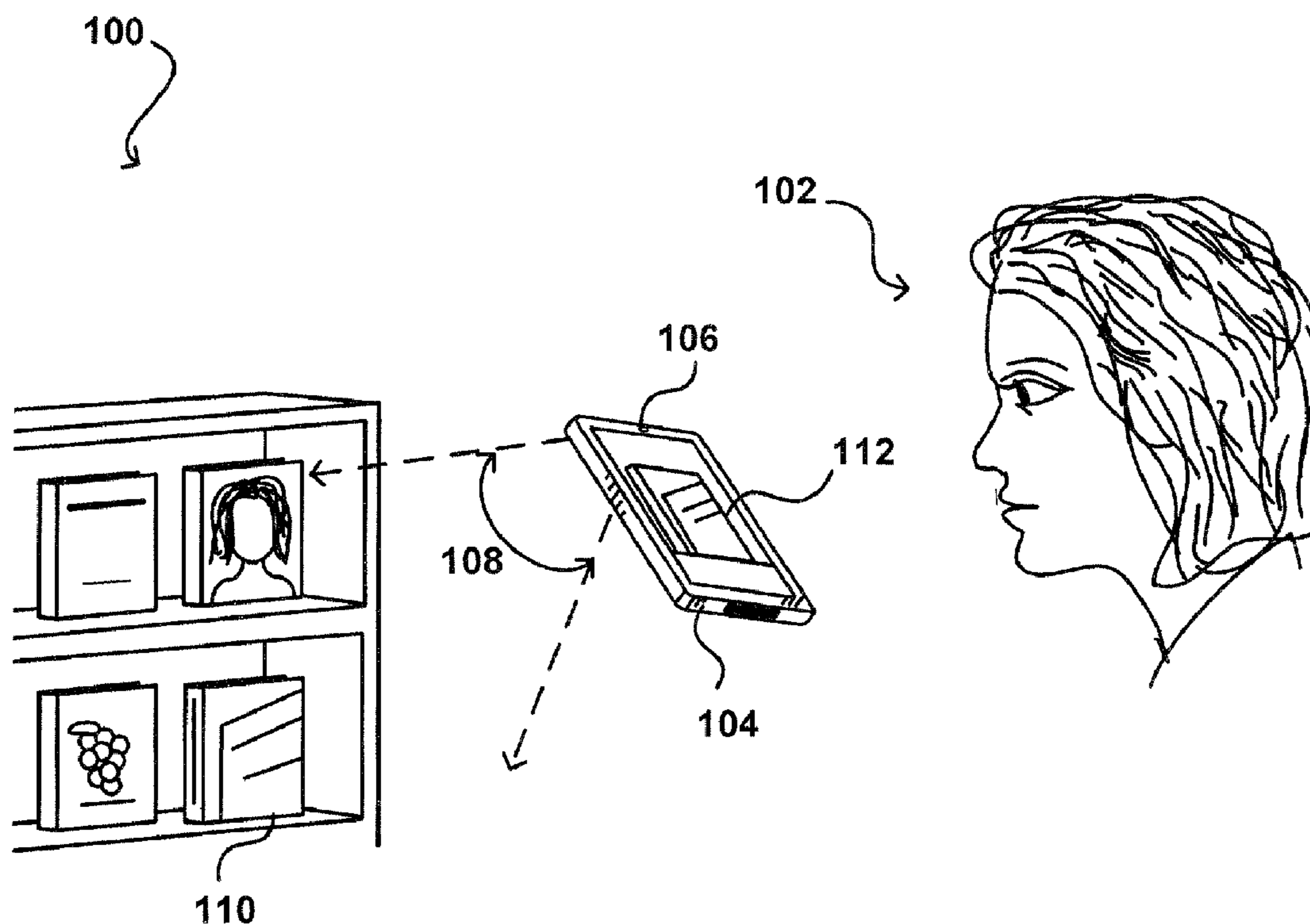
Primary Examiner — Usman Khan

(74) *Attorney, Agent, or Firm* — Novak Druce Connolly Bove + Quigg LLP

(57) **ABSTRACT**

An object represented in an image can be segmented from the image background by capturing a pair of images, one with flash and one without, and generating a differential image. This differential image can be analyzed using an algorithm, such as a connected components or computer vision algorithm, to determine one or more portions of the image that correspond to an object. An appropriate one of these objects can be selected as corresponding to the object of interest, and an outline of the selected object can be used to determine a portion of one of the original images that corresponds to the object. This portion then can be provided to an object recognition or other such process for analysis, which can increase the efficiency and accuracy of the analysis.

22 Claims, 9 Drawing Sheets



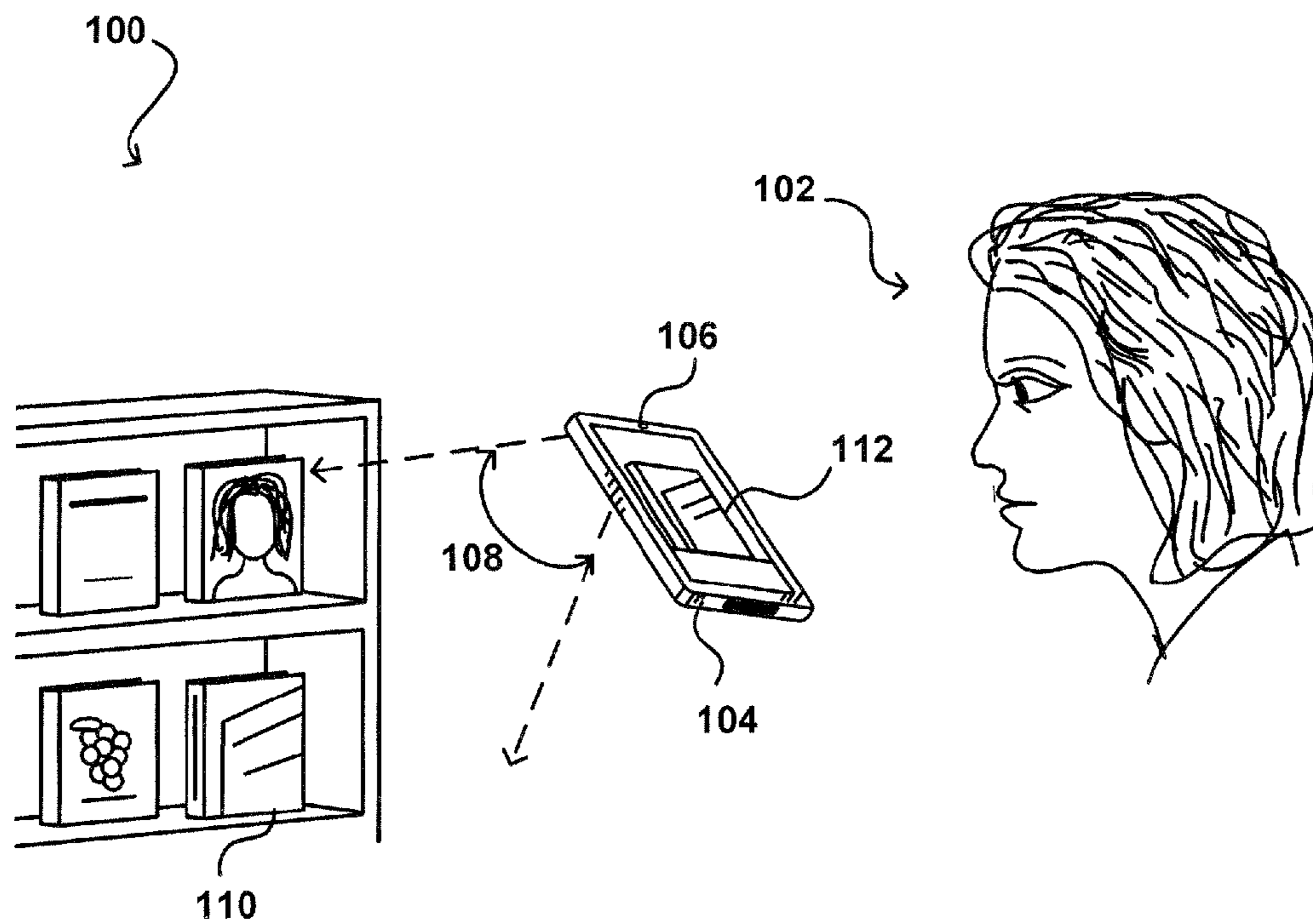


FIG. 1

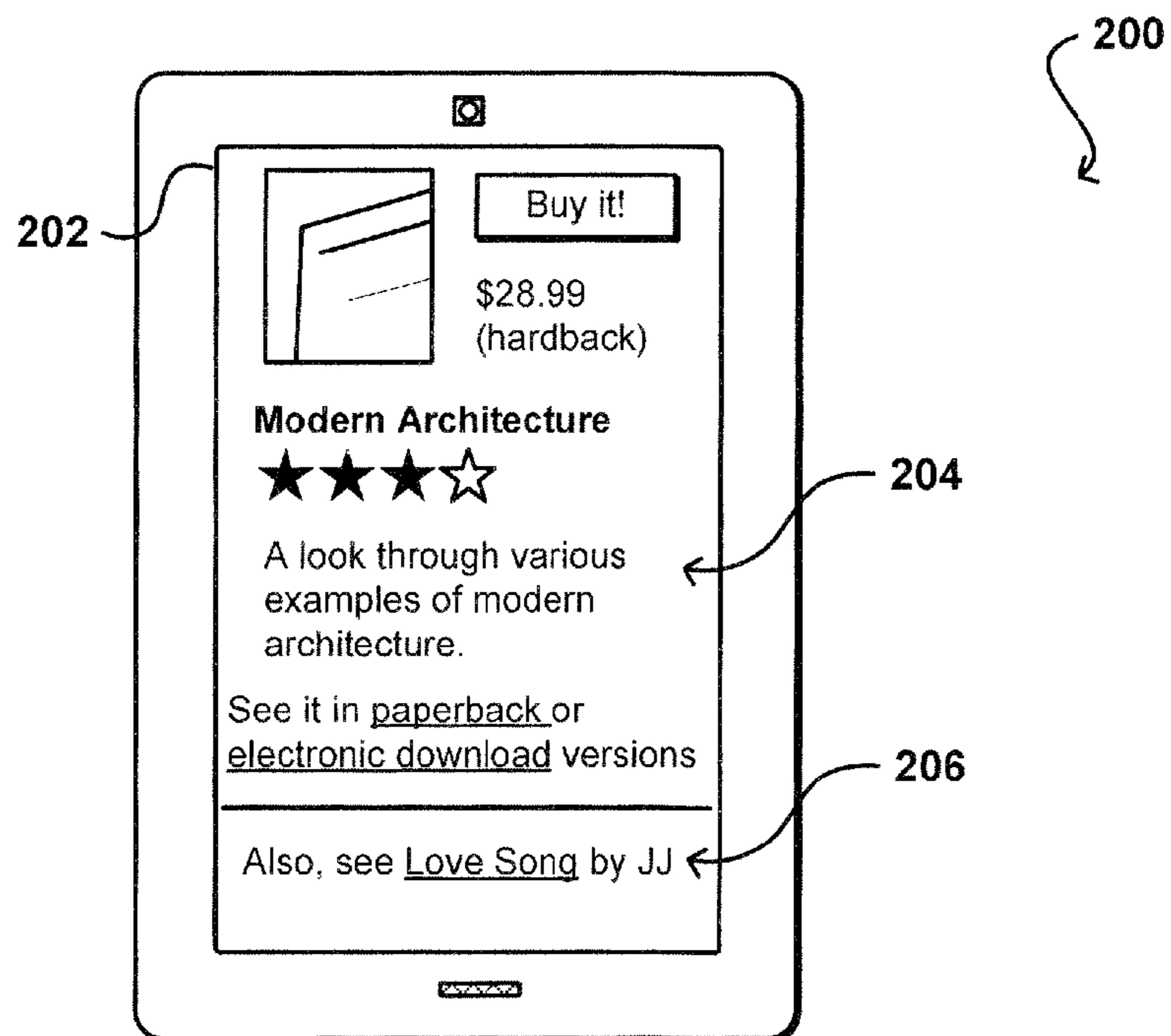


FIG. 2

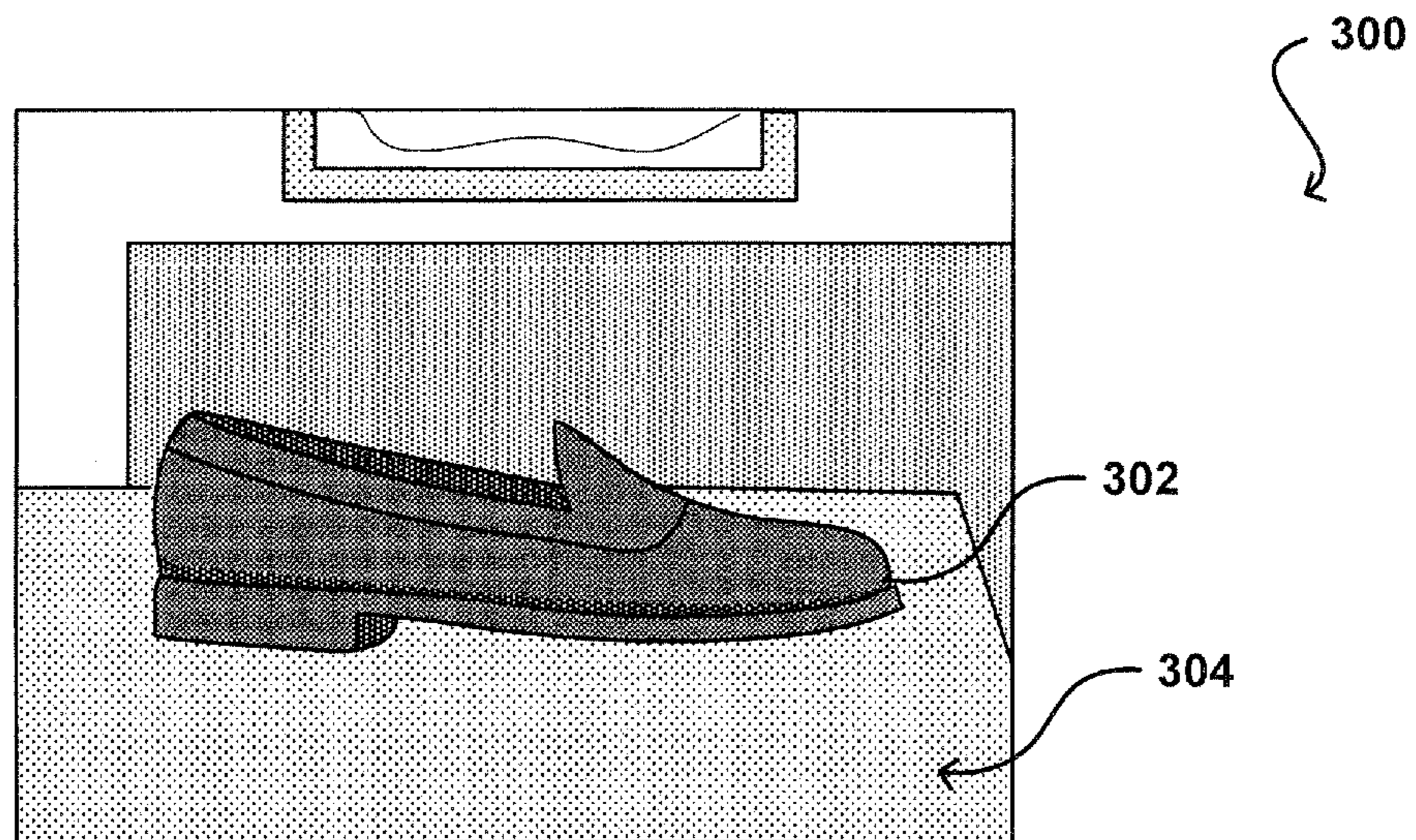


FIG. 3(a)

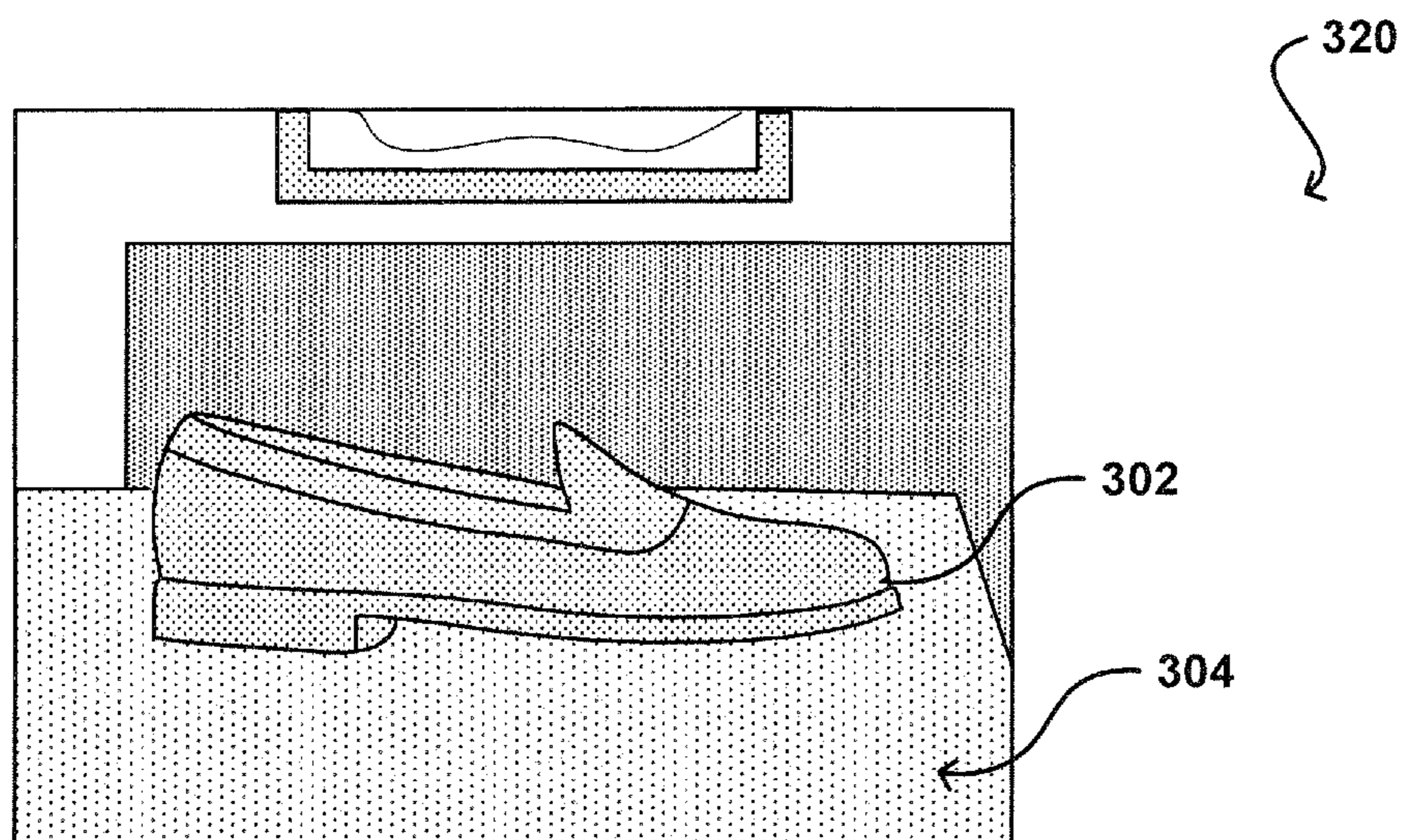


FIG. 3(b)

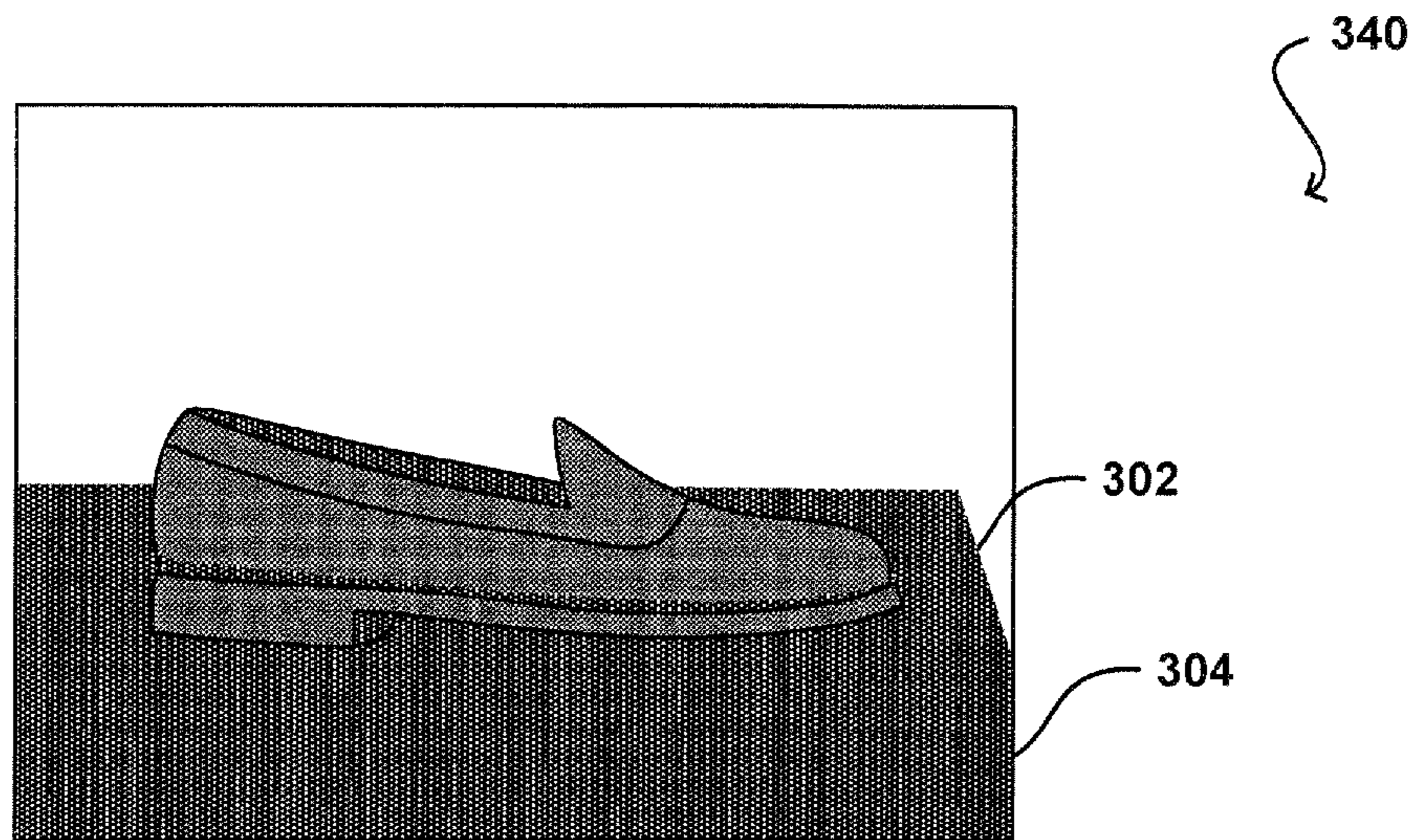


FIG. 3(c)

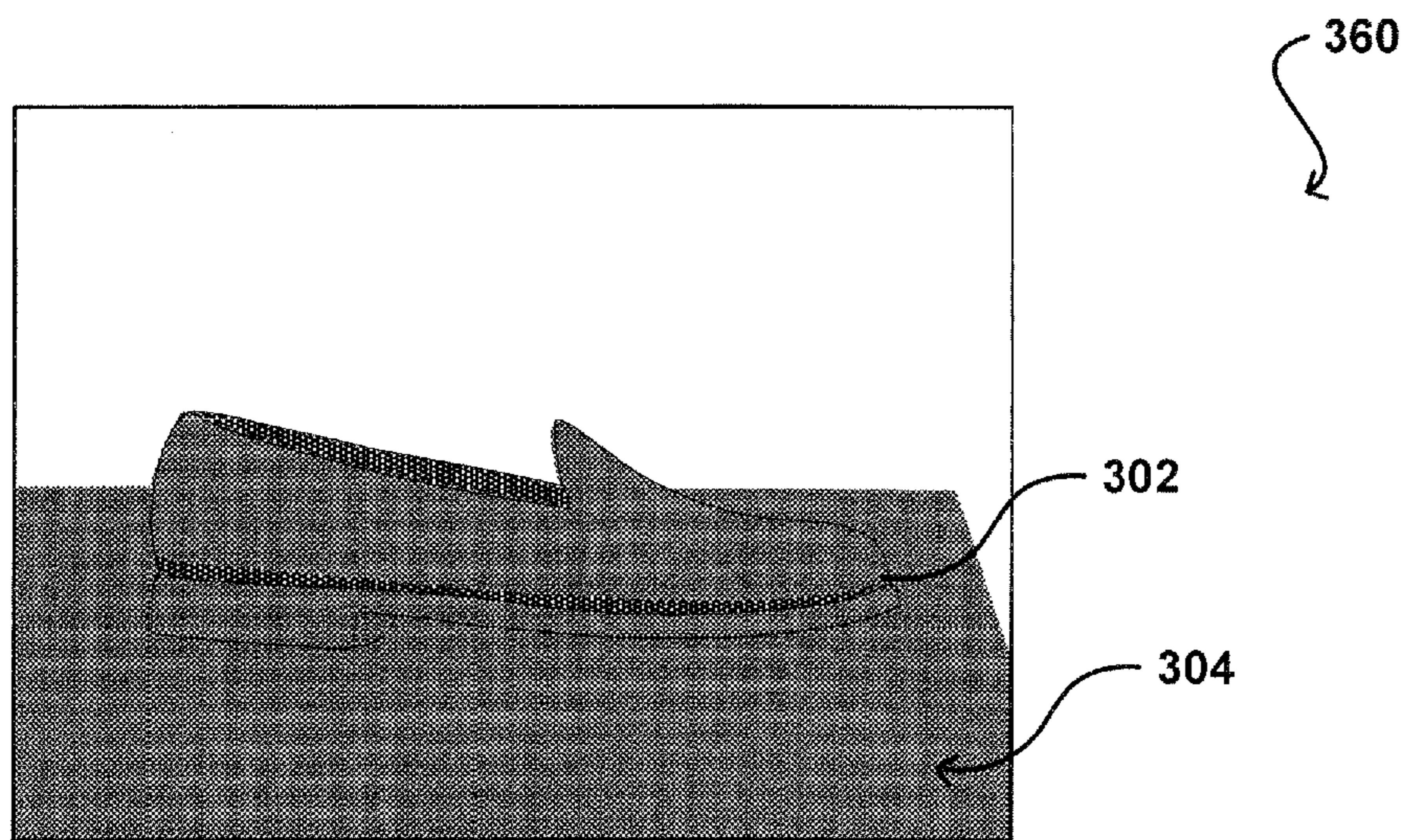


FIG. 3(d)

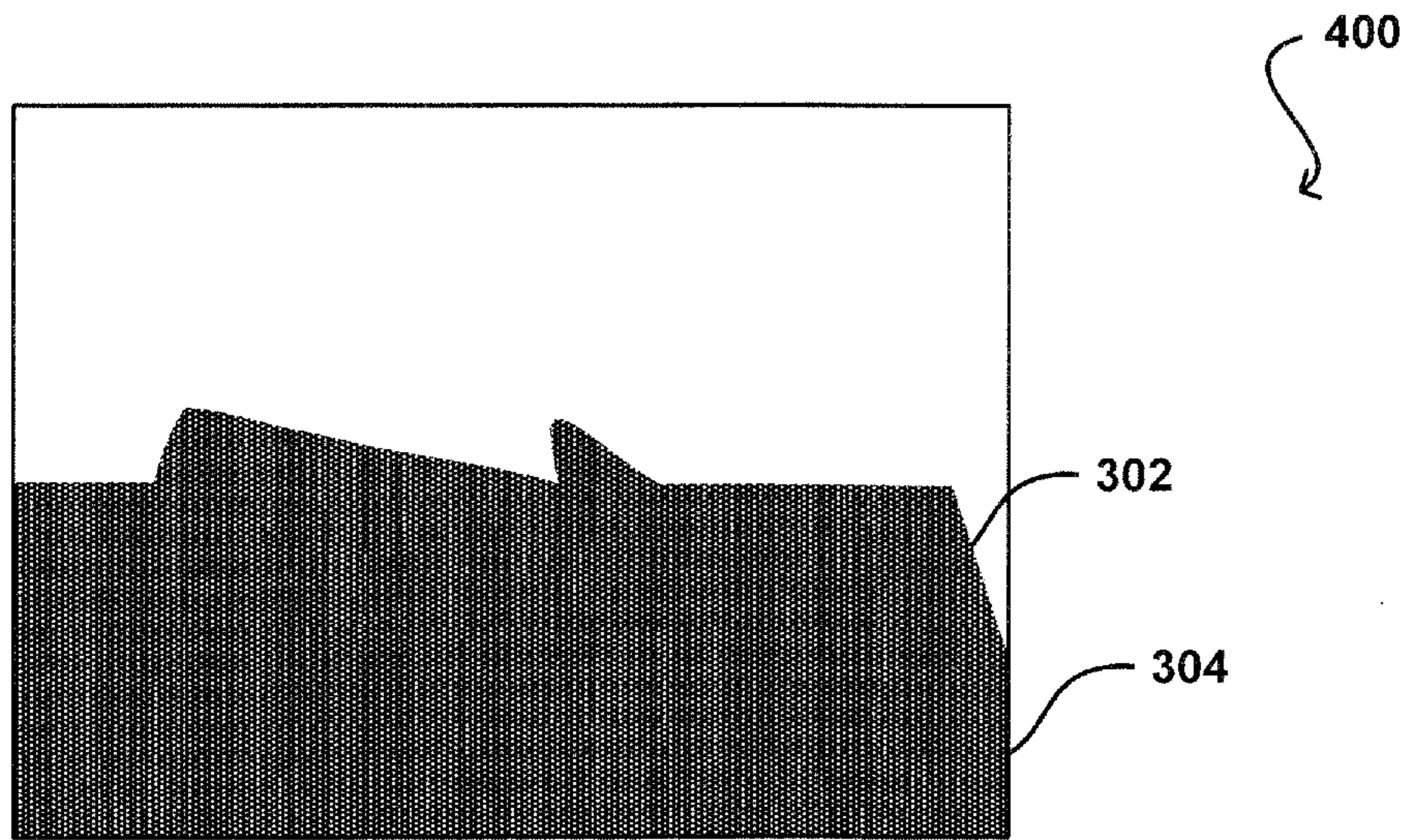


FIG. 4(a)

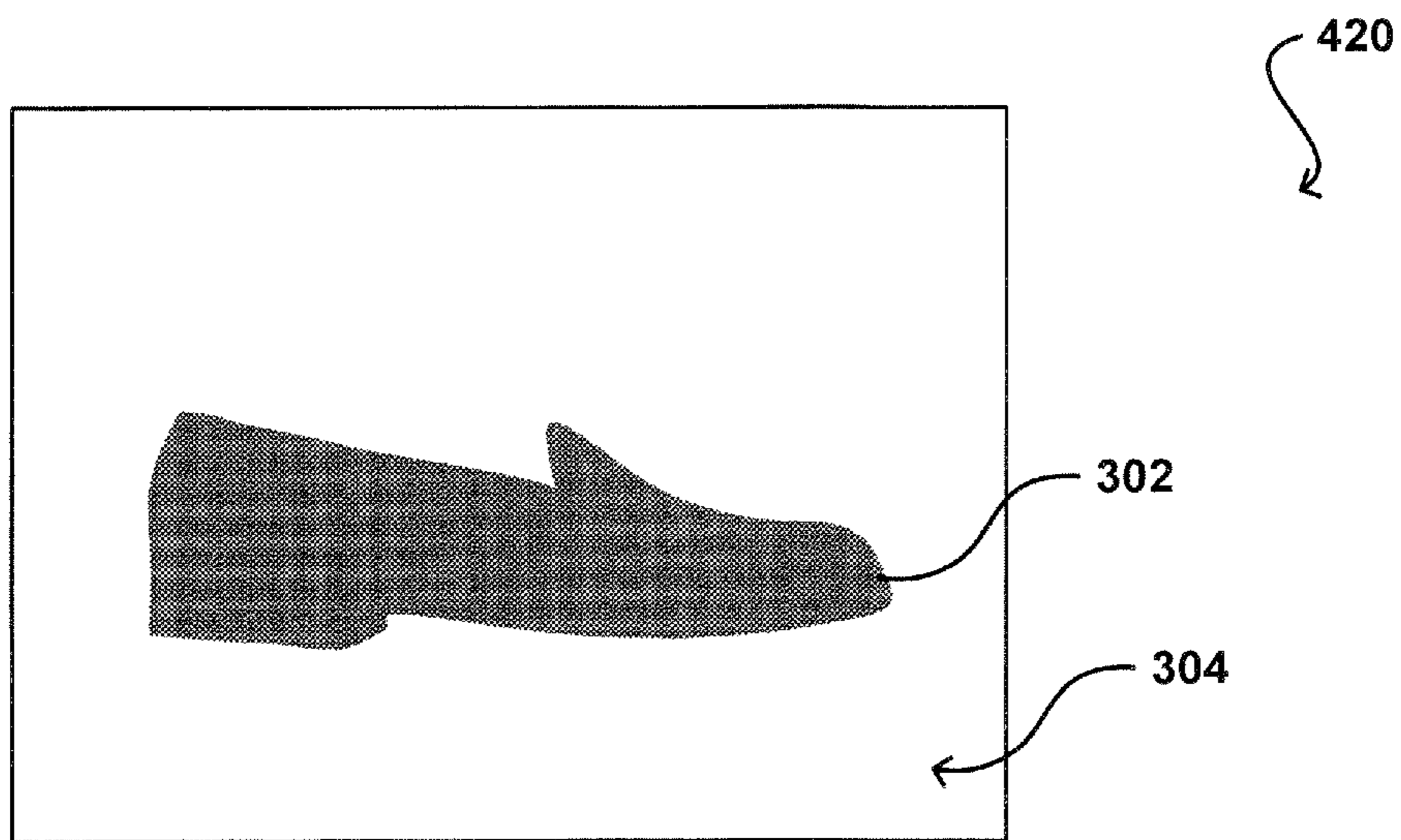


FIG. 4(b)

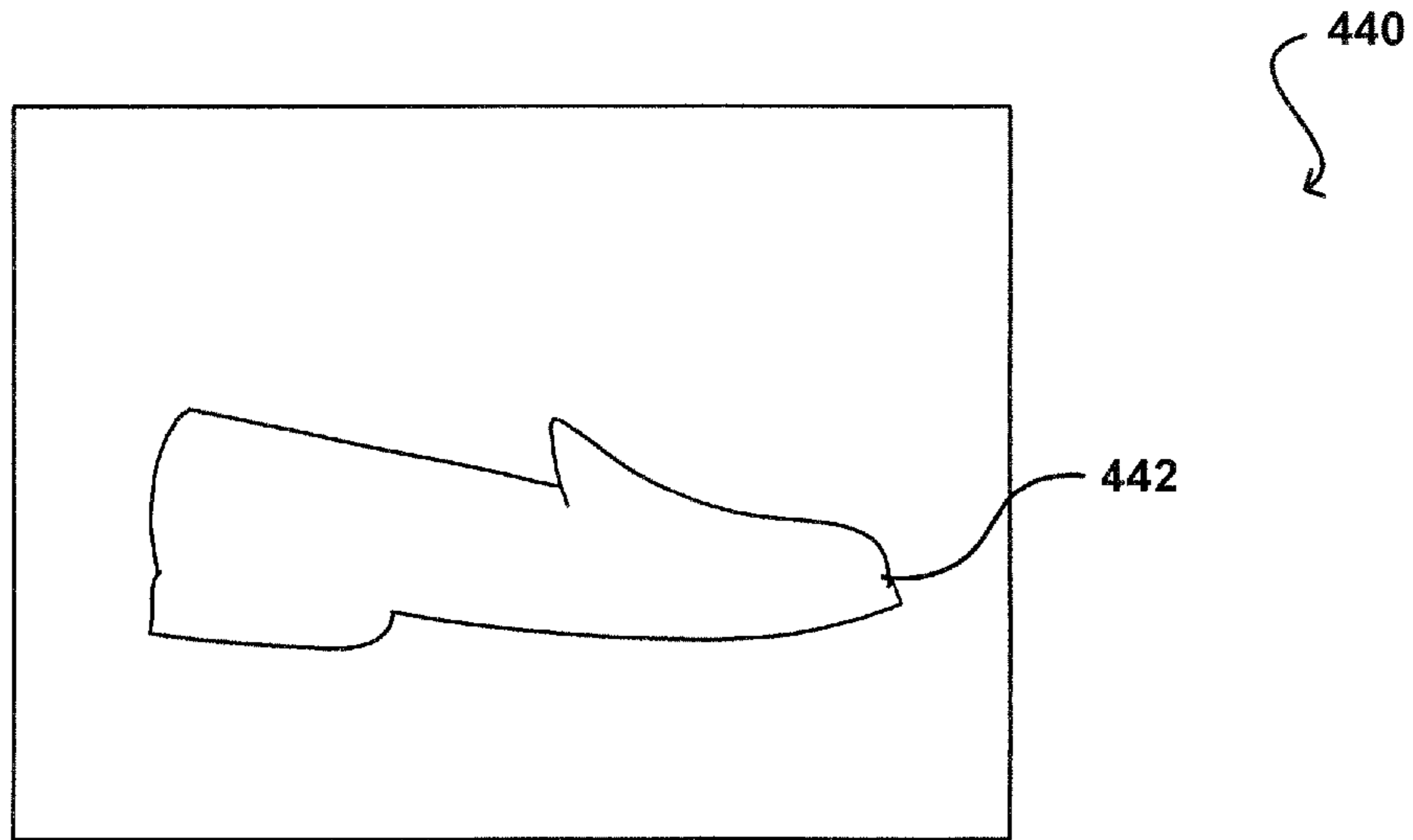


FIG. 4(c)

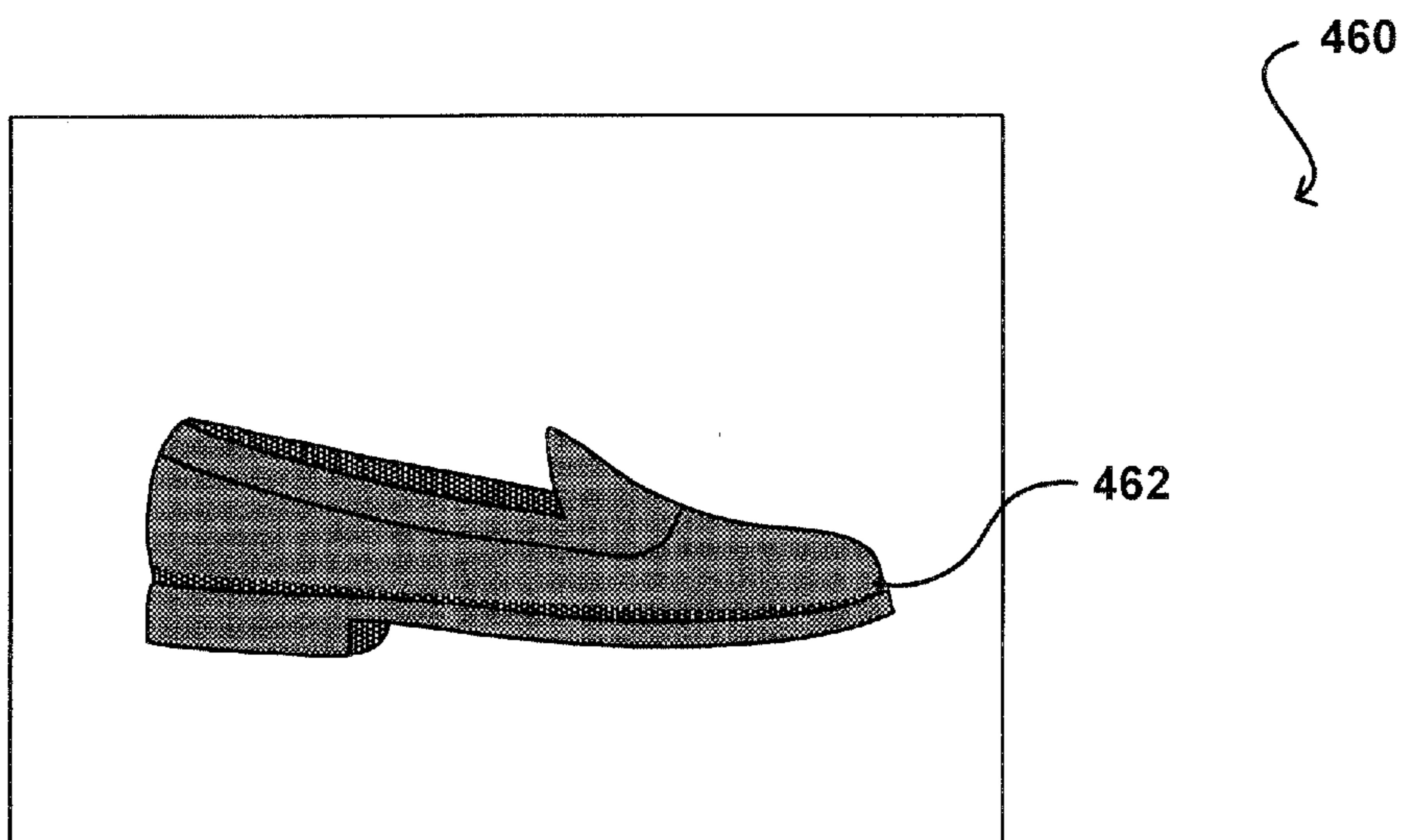


FIG. 4(d)

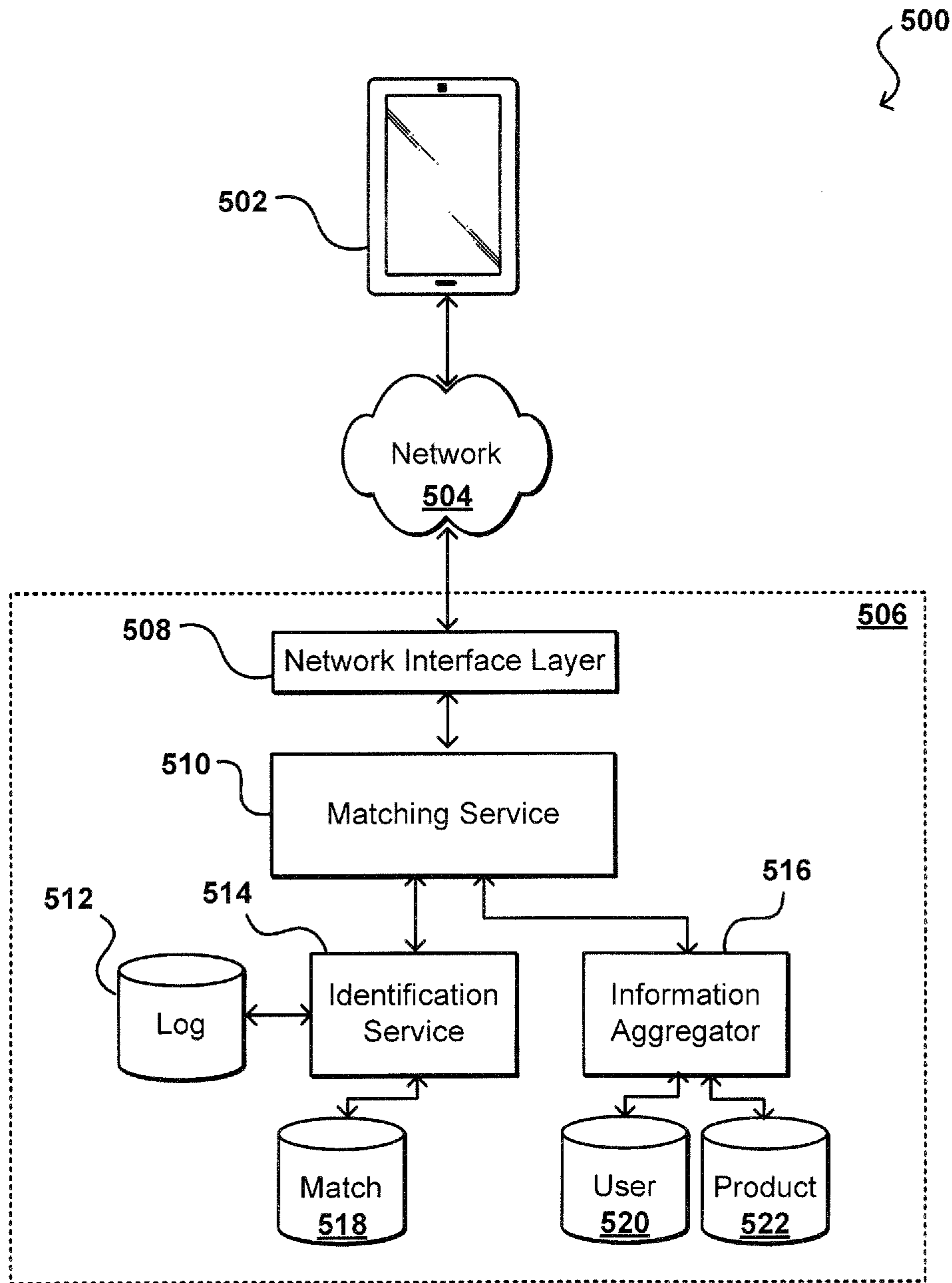


FIG. 5

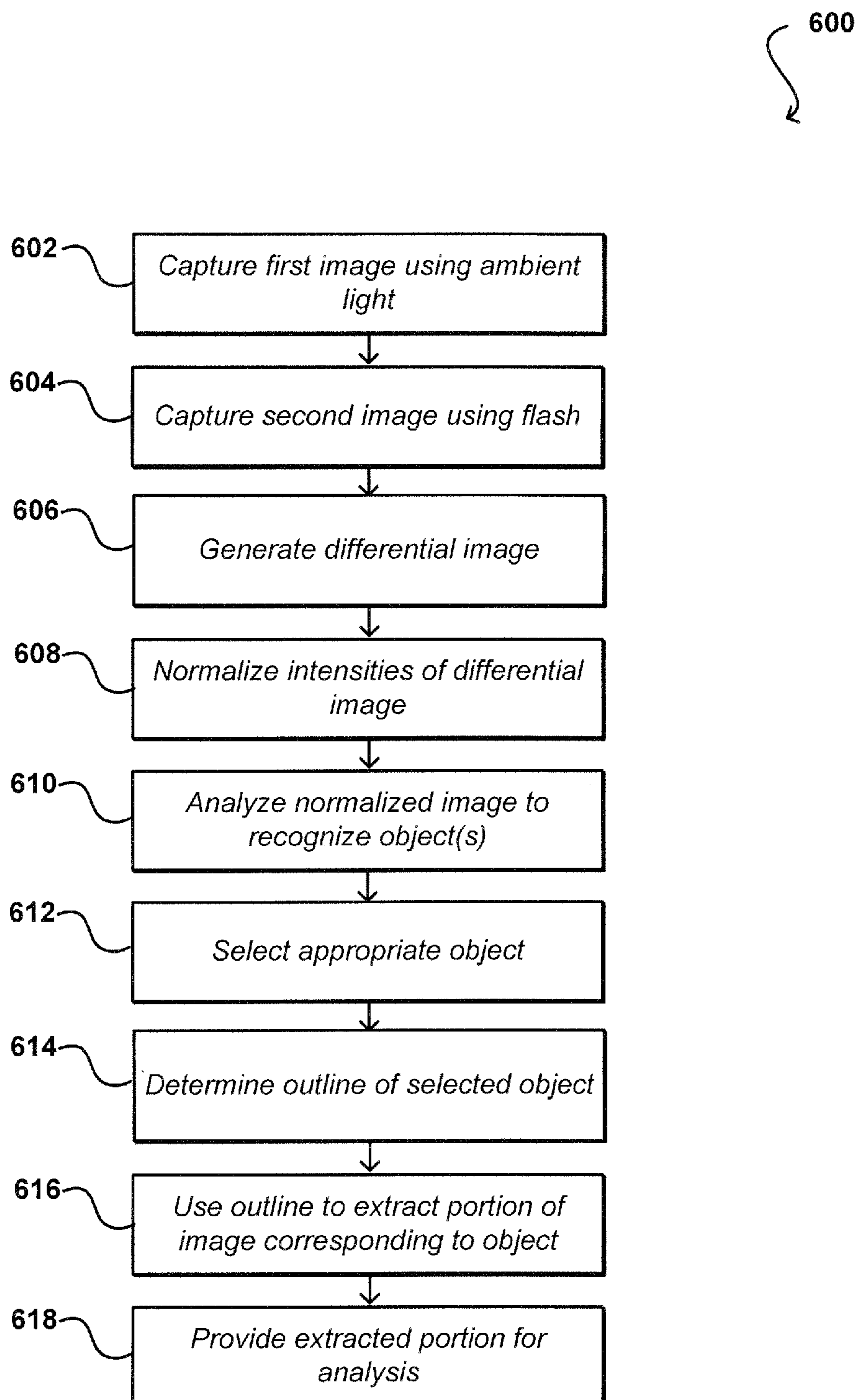


FIG. 6

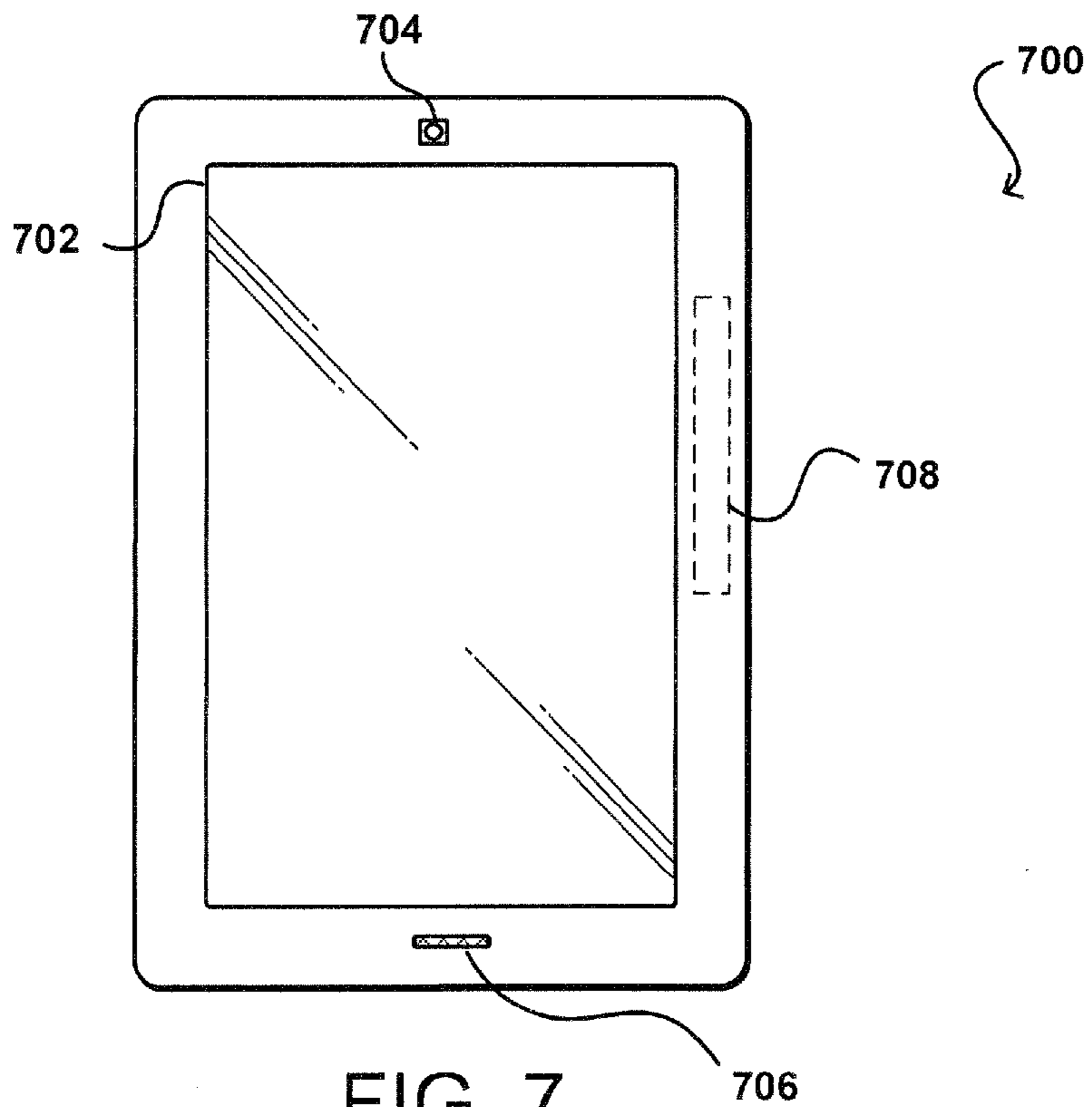


FIG. 7

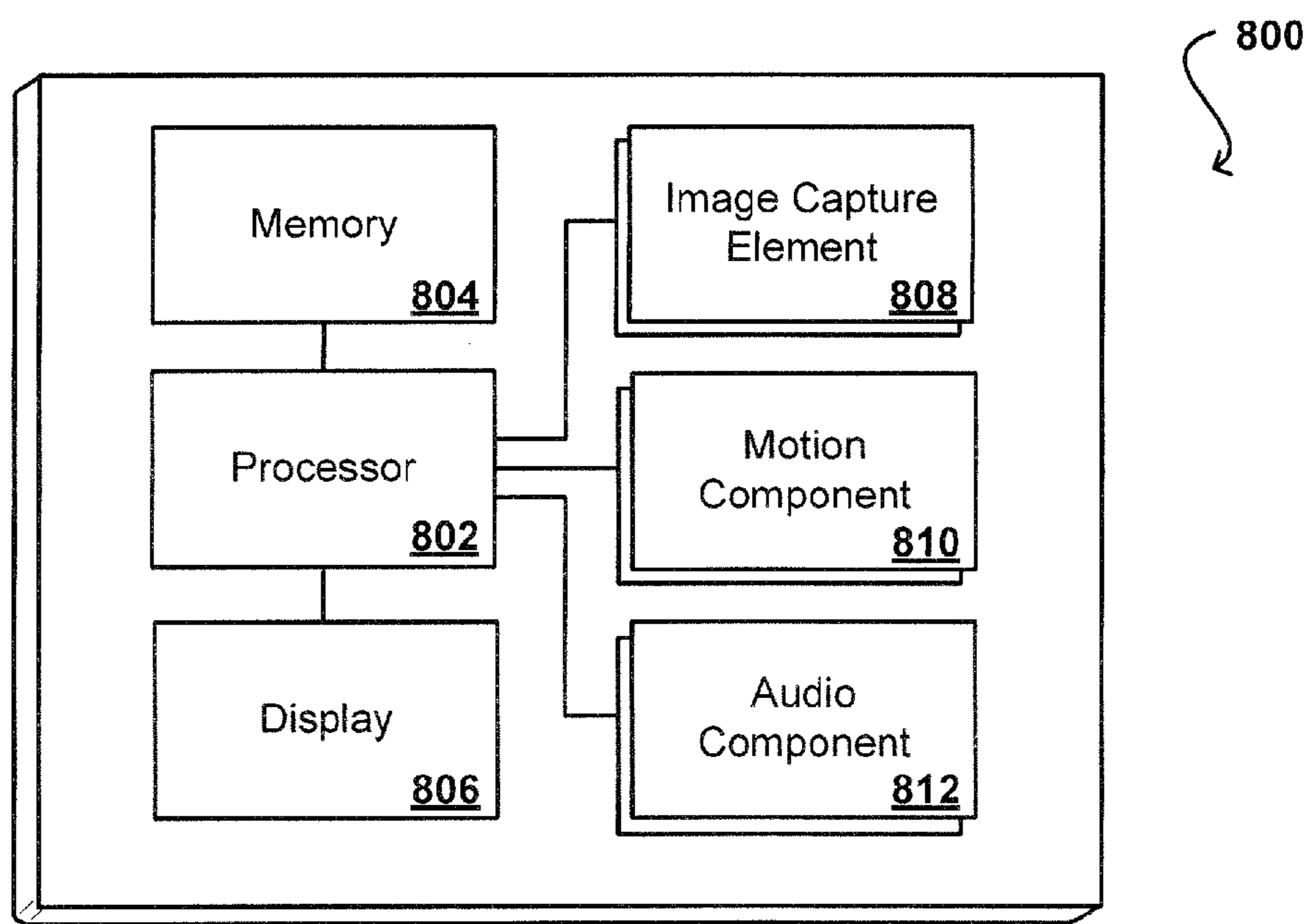


FIG. 8

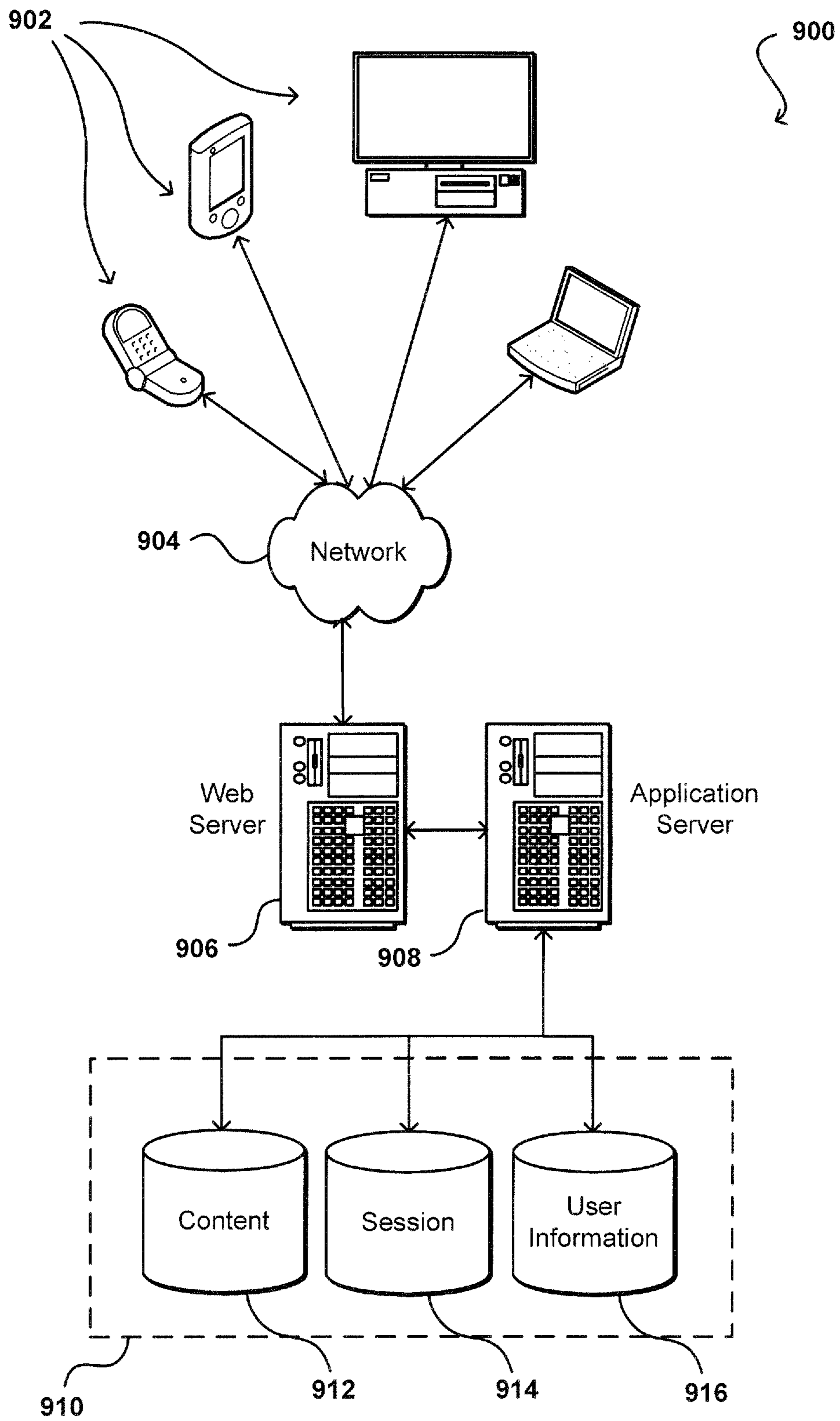


FIG. 9

SEGMENTATION APPROACHES FOR OBJECT RECOGNITION

BACKGROUND

Users are increasingly utilizing electronic devices to obtain various types of information. For example, a user wanting to obtain information about a book can capture an image of the cover of the book and upload that image to a book identification service for analysis. In many cases, the cover image will be matched against a set of two-dimensional images including views of objects from a particular orientation. While books are typically relatively easy to match, as a book cover generally includes several features that enable the cover to be matched against a set of cover images, other objects are not as straightforward to match. For example, an object such as a men's dress shoe that is captured from the side might not have many distinctive features, and may appear primarily as a shaped black object in the image. In order to efficiently perform image matching for such an object, the object of interest is often first separated from the background portion of the image. Unfortunately, it can be difficult to separate an object that does not have many unique features that help to distinguish the object from the background. Accordingly, objects such as shoes can take longer to recognize, and the results can be less accurate on average than for objects such as books or media packaging.

BRIEF DESCRIPTION OF THE DRAWINGS

Various embodiments in accordance with the present disclosure will be described with reference to the drawings, in which:

FIG. 1 illustrates an example environment in which aspects of the various embodiments can be that can be utilized;

FIG. 2 illustrates example display that can be presented in accordance with various embodiments;

FIGS. 3(a), 3(b), 3(c), and 3(d) illustrate example images of an object that can be captured and generated during processing in accordance with various embodiments;

FIGS. 4(a), 4(b), 4(c), and 4(d) illustrate example images of an object that can be generated during processing in accordance with various embodiments;

FIG. 5 illustrates example system for identifying objects and providing information about those objects that can be utilized in accordance with various embodiments;

FIG. 6 illustrates an example process for determining information about an object imaged by a user that can be utilized in accordance with various embodiments;

FIG. 7 illustrates an example device that can be used to implement aspects of the various embodiments;

FIG. 8 illustrates example components of a client device such as that illustrated in FIG. 7; and

FIG. 9 illustrates an environment in which various embodiments can be implemented.

DETAILED DESCRIPTION

Systems and methods in accordance with various embodiments of the present disclosure overcome one or more of the above-referenced and other deficiencies in conventional approaches to identifying objects using an electronic device. In particular, various embodiments enable a user to capture images including a view of an object of interest and receive information about one or more objects that are determined to match based at least in part on the captured images. A pair of images is captured in at least some embodiments, with a first

image being captured without use of a flash and a second image being captured with a flash (or other source of illumination). The images can be compared in order to attempt to suppress a significant portion of the background in the image, as the flash will generally affect objects in the foreground much more than most objects in the background. In some embodiments, the resulting image can be a grayscale image that can have the intensities normalized to assist with processing. An algorithm such as a connected components algorithm can be applied to the normalized image to attempt to locate portions corresponding to one or more objects in the image. Based upon information such as the shape of these located objects, whether edges of the objects appear in the image, and other such information, a selection process can determine the portion that likely corresponds to the object of interest. The shape or outline of this portion or region then can be used with one of the original captured images to extract the portion of the image that corresponds to the object of interest. This portion then can be provided to an object recognition, image matching, or other such process.

Various other functions and advantages are described and suggested below as may be provided in accordance with the various embodiments.

FIG. 1 illustrates an example environment 100 in which aspects of the various embodiments can be implemented. In this example, a user 102 is in a store that sells books, and is interested in obtaining information about a book 110 of interest. Using an appropriate application executing on a computing device 104, the user is able to obtain an image of the book 110 by positioning the computing device such that the book is within a field of view 108 of at least one camera 106 of the computing device. Although a portable computing device (e.g., an electronic book reader, smart phone, or tablet computer) is shown, it should be understood that any electronic device capable of receiving, determining, and/or processing input can be used in accordance with various embodiments discussed herein, where the devices can include, for example, desktop computers, notebook computers, personal data assistants, video gaming consoles, television set top boxes, and portable media players, among others.

In this example, a camera 106 on the device 104 can capture image information including the book 110 of interest, and at least a portion of the image can be displayed on a display screen 112 of the computing device. At least a portion of the image information can be analyzed and, upon a match being located, identifying information can be displayed back to the user via the display screen 112 of the computing device 104. The portion of the image to be analyzed can be indicated manually, such as by a user pointing to the book on the screen or drawing a bounding box around the book. In other embodiments, one or more image analysis algorithms can attempt to automatically locate one or more objects in an image. In some embodiments, a user can manually cause image information to be analyzed, while in other embodiments the image information can be analyzed automatically, either on the device or by transferring image data to a remote system or service as discussed later herein.

FIG. 2 illustrates an example of a type of information 204 that could be displayed to the user via a display screen 202 of a computing device 200 in accordance with various embodiments. In this example, the image captured by the user has been analyzed and related information 204 is displayed on the screen. The "related" information as discussed elsewhere herein can include any information related to an object, item, product, or other element that is matched (within at least a level of confidence) to the image data using one or more matching or identifying algorithms, or other such approaches.

These can include, for example, image recognition algorithms, object identification algorithms, facial recognition algorithms, or any other such approaches or techniques. The displayed information in this example includes the title of the located book, an image of the book (as captured by the user or otherwise obtained), pricing and description information, and review information. Also as shown are options to purchase the book, as well as options for various other versions or forms of that content, such as a paperback book or digital download. The type of information displayed (or otherwise conveyed) can depend at least in part upon the type of content located or matched. For example, a located book might include author and title information, as well as formats in which the book is available. For facial recognition, the information might include name, title, and contact information. Various other types of information can be displayed as well within the scope of the various embodiments.

As discussed, however, other types of objects can be more difficult to recognize based on a captured image. For example, FIG. 3(a) illustrates an image that has been captured that includes a representation of a men's dress shoe 302. One challenge when attempting to recognize such an object is that the object is a non-planar object, as opposed to a planar object such as a book cover as illustrated in FIG. 1. While users will typically capture an image of a book cover from a similar perspective, a non-planar object such as a shoe can be captured from various angles, where even a slight change in angle can significantly affect the overall shape of the object in the image. Another challenge arises from the fact that the dress shoe 302 does not have a significant number of unique features within the portion of the image corresponding to the shoe. While a book cover can include text, images, and other such features that may be unique to that book, the shoe might be a polished black shoe that can primarily be recognized using only the outline, silhouette, or shape of the shoe in the current view. Conventional algorithms extract features from both foreground and background objects or portions, without attempting to determine object boundaries or edges. These conventional approaches use a sheer force or SIFT-like method to eliminate any outliers, or features that likely correspond to the background based on factors such as location, density, etc. Such approaches, however, are not particularly effective for objects without distinctive features other than the outline or shape of the object.

Approaches in accordance with various embodiments can improve the accuracy and efficiency of object recognition for objects including those without several distinctive features, particularly those objects that belong to a class represented primarily by the shapes of the objects. Various approaches can utilize a pair of images of an object to assist in segmenting an object of interest from background or other objects in the images, enabling an object boundary to be determined. Such an approach effectively eliminates most of the outlier features, which can improve precision over conventional approaches for other types of objects as well.

An approach in accordance various embodiments uses a two-stage process for segmenting an object from other portions of an image, where those stages include a pre-processing stage and a processing stage. It should be understood that these stages are used for purposes of explanation only, and that approaches discussed herein can be performed as part of a single process or multiple processes.

In an example pre-processing stage, two captured images can be obtained that include similar views of an object of interest. The images can be digital still images captured at different times (within a determined allowable amount of time) or frames of video corresponding to different points in

time, among other such options. For one of the images, a flash or other source of illumination can be activated such that at least some objects in the images will reflect, or at least show the effects of, the flash. The flash element can be any appropriate source of illumination, such as a digital flash, a flash gun, a flashtube, a microflash, an LED, a ring lite, and the like. An advantage to using flash-type illumination for one of the images is that objects in the foreground will generally reflect more light, and thus appear brighter, than background objects that are further away. Such an approach enables foreground objects to be identified with respect to at least some background objects or areas, although objects such as mirrors might reflect very well even when in the background of an image.

As an example, FIG. 3(a) illustrates an image 300 of a shoe 302 captured at a first time without an active flash, and FIG. 3(b) illustrates a second image 320 of the same shoe, from approximately the same angle and point of view, where the second image 320 is captured while a flash or other source of illumination is active. Auto-exposure can be performed on the image captured with flash in order to optimally utilize the camera's dynamic range for a relatively bright foreground portion. As can be seen, objects in the foreground such as the shoe 302 and table 304 appear significantly brighter in the image 320 captured with flash than the image 300 captured without flash. On the other hand, objects in the background do not substantially change in appearance as a result of the flash between images. Since the objects in the foreground are primarily the objects that change in appearance between the two images, performing an image subtraction process or calculating a differential image or frame (i.e., pixel-by-pixel comparisons for at least a portion of the image, frame_flash-frame_noflash) can cause a portion of the background to be effectively removed and/or suppressed, as the pixel, intensity, and/or color values will be substantially similar for those background portions between images since flash modulation is weak for a large part of the distant background. These background regions can appear substantially white or substantially black in the images, depending upon the approach, and among other such options. As an example, FIG. 3(c) illustrates an example of a differential image 340 that could be generated using the two images, where intensity values for corresponding objects can be subtracted in order to obtain an image that reflects the differences in intensities for each location in the view shared by the two images. A differential image (I_{diff}) can be calculated in at least some embodiments according to the following:

$$I_{diff} = I_2 - I_1,$$

where

$$I_1 = E_1 * R$$

and

$$I_2 = E_2 * R * \cos(\alpha) / Z^4,$$

with I_1 being the image without flash and only ambient light E_1 and matte reflectance map R , and I_2 being the image with flash, which depends upon the angle α between the light source direction and the surface normal. Also, the normalized image can be determined from:

$$I_{norm} = I_{diff} / I_1 - \cos(\alpha) / Z^4,$$

where the normalized image does not depend on object color variation, but retains the dependence on distance and surface curvature. While in this example the differential image includes substantially only the portions corresponding to the

5

shoe **302** and the table **304**, it should be understood that due to noise, auto-exposure adjustments, and other such reasons that there can be at least some other features represented in a differential image as well in at least some examples. Further, curves, angles, and other aspects of the object can cause at least certain portions of the object to reflect differently, which can also impact the accuracy of the comparison process. As can be seen, however, the portions of the original image that are primarily represented in the differential image correspond to the foreground objects. Unfortunately, factors such as object color variation can still complicate segmentation of this differential image.

In order to minimize this problem, a normalization process can be applied whereby the colors (or intensities for a grayscale image) of the differential image can be normalized using the colors (or intensities) from original non flash image, or the average of flash and no-flash images in some embodiments. Such an approach can help to reduce the effective differences due to color. FIG. **3(d)** illustrates an example output image **360**, herein referred to as a normalized image, which shows the foreground objects with reduced color and/or intensity variations, as may be due to the flash or other such factors. As discussed, in some embodiments the background can be dark or black, but in order to comply with figure requirements the background is shown as being substantially white for these examples. An image such as the normalized image **360** of FIG. **3(d)** can be the end result of the pre-processing stage, which then can be provided to the processing or segmentation stage in at least some embodiments. Eliminating color variations inside the object region can significantly facilitate object segmentation, as features in the resulting image depend primarily upon surface curvature and distance from the camera.

During a processing stage, one or more segmentation techniques can be applied to the normalized image. Any of a number of conventional and/or modified segmentation techniques can be used for such purposes. For example, a computer vision algorithm or other segmentation algorithm (e.g., GrabCut, WaterShed, or QuadTree) can be used if a sufficient initialization process is used that determines the approximate region of the object within an allowable amount of variation. In the present example, however, a connected component algorithm can be used with a Canny edge filter (which locates edges based on changes in color or intensity) to select raw foreground information and use this information to generate a raw outline of one or more objects in the normalized image, as objects near the foreground might provide similar intensity or color values, and thus each be picked up by a connected component algorithm. Such an algorithm can look at the intensity value of a point and compare that value to the intensity value of nearby points to attempt to determine points that likely correspond to a common object, based on factors such as the amount of variation in intensity over a given distance. The process can continue expanding out from the point until reaching the "edge" of a region where the points no longer appear to belong to the same object. The shape of this region should roughly approximate the shape of an item in the image. As an example, a first result of such an algorithm can be an object **402** that essentially corresponds to the table, as illustrated in the image **400** of FIG. **4(a)**. Such an approach can analyze portions of the table in the image and attempt to connect those according to criteria of the algorithm, which can attempt to determine continuous surfaces, etc. Another result can provide data for an object **422** that essentially corresponds to the shoe portion **422** of the image. In some embodiments where a user manually selects the appropriate image, each object can be represented in a different color for

6

ease of distinction. In embodiments where an automated process analyzes the objects for selection, a single color, value, or setting can be used, among other such options. In some embodiments, the selection can be based upon an arrangement of the objects, such as where one object is determined to be on, or in front of, the other object. In some embodiments, the selection can be based at least in part upon a recognized shape of one of the objects, or whether the edge or outline of the object can be determined from the image. Other such criteria or factors can be used as well.

Once a portion is selected that corresponds to a foreground object and likely corresponds to the object of interest, an outline, edge, or shape of that object can be determined. For example, if the shoe portion **422** of FIG. **4(b)** is selected as the appropriate object, an outline **442** encompassing the outer edge of that portion can be determined, as illustrated in the image **440** of FIG. **4(c)**. In at least some embodiments, one or more auto-detect thresholds can be applied and the calculations repeated in order to iteratively refine the outline until at least one completion threshold or criterion is reached. This final outline then can be used to select a portion of the first image, for example, that corresponds to the object of interest. This portion **462**, as illustrated in the image **460** of FIG. **4(d)**, corresponds substantially to the shoe in the first image, and can be provided to an appropriate process for identification, matching, recognition, etc. The resulting image can be pulled from the original color image, while at least a portion of the processing and/or pre-processing can be performed on grayscale or similar versions of the original images. The efficiency and/or accuracy of the subsequent process then can be improved since the image substantially contains information only for the object of interest.

Certain approaches use infrared (IR) illumination instead of a flash, since IR is faster and the processing can be done in near real time. The use of flash, however, is more powerful than IR, even though flash may require an offline process. An offline process can be acceptable for processes such as object recognition, however, where the user might be willing to wait up to a couple of seconds to receive the results.

A potential downside to using flash, however, is that there will necessarily be some delay between capturing an image of an object with flash and another image of the object without flash. Such a delay can allow for movement of the camera, which can impact the image subtraction or differential process as the portions to be compared will no longer align. In some embodiments a sensor such as an electronic gyroscope or accelerometer can be used to detect motion, which then can be used to attempt to align the object in the images. Various other approaches exist for aligning images as well. In some embodiments, the sensor data can detect when more than an allowable amount of movement has occurred between image captures, and might simply indicate to the user that too much movement occurred and the user should attempt to capture the images again. Such an approach also has the benefit that it can help to minimize blur in the images, which can also improve segmentation, matching, and other such processes.

As discussed, information such as that illustrated in FIG. **2** can be located by providing the image data before and/or after processing to a system or service operable to find one or more potential matches for that data and provide related information for those potential matches. FIG. **5** illustrates an example environment **500** in which such information can be located and transferred in accordance with various embodiments. In this example, a user is able to capture one or more types of information using at least one computing device **502**. For example, a user can cause a device to capture audio and/or video information around the device, and can send at least a

portion of that audio and/or video information across at least one appropriate network **504** to attempt to obtain information for one or more objects, persons, or occurrences within a field of view of the device. The network **504** can be any appropriate network, such as may include the Internet, a local area network (LAN), a cellular network, and the like. The request can be sent to an appropriate content provider **506**, as may provide one or more services, systems, or applications for processing such requests. The information can be sent by streaming or otherwise transmitting data as soon as it is obtained and/or ready for transmission, or can be sent in batches or through periodic communications. In some embodiments, the computing device can invoke a service when a sufficient amount of image data is obtained in order to obtain a set of results. In other embodiments, image data can be streamed or otherwise transmitted as quickly as possible in order to provide near real-time results to a user of the computing device.

In this example, the request is received to a network interface layer **508** of the content provider **506**. The network interface layer can include any appropriate components known or used to receive requests from across a network, such as may include one or more application programming interfaces (APIs) or other such interfaces for receiving such requests. The network interface layer **508** might be owned and operated by the provider, or leveraged by the provider as part of a shared resource or “cloud” offering. The network interface layer can receive and analyze the request, and cause at least a portion of the information in the request to be directed to an appropriate system or service, such as a matching service **510** as illustrated in FIG. 5. A matching service in this example includes components operable to receive image data about an object, analyze the image data, and return information relating to people, products, places, or things that are determined to match objects in that image data.

The matching service **510** in this example can cause information to be sent to at least one identification service **514**, device, system, or module that is operable to analyze the image data and attempt to locate one or more matches for objects reflected in the image data. In at least some embodiments, an identification service **514** will process the received data, such as to extract points of interest or unique features in a captured image, for example, then compare the processed data against data stored in a matching data store **520** or other such location. In other embodiments, the unique feature points, image histograms, or other such information about an image can be generated on the device and uploaded to the matching service, such that the identification service can use the processed image information to perform the match without a separate image analysis and feature extraction process. Certain embodiments can support both options, among others. The data in an image matching data store **520** might be indexed and/or processed to facilitate with matching, as is known for such purposes. For example, the data store might include a set of histograms or feature vectors instead of a copy of the images to be used for matching, which can increase the speed and lower the processing requirements of the matching. Approaches for generating image information to use for image matching are well known in the art and as such will not be discussed herein in detail.

The matching service **510** can receive information from each contacted identification service **514** as to whether one or more matches could be found with at least a threshold level of confidence, for example, and can receive any appropriate information for a located potential match. The information from each identification service can be analyzed and/or processed by one or more applications of the matching service, such as to determine data useful in obtaining information for

each of the potential matches to provide to the user. For example, a matching service might receive bar codes, product identifiers, or any other types of data from the identification service(s), and might process that data to be provided to a service such as an information aggregator service **516** that is capable of locating descriptions or other content related to the located potential matches.

In at least some embodiments, an information aggregator might be associated with an entity that provides an electronic marketplace, or otherwise provides items or content for consumption (e.g., purchase, rent, lease, or download) by various customers. Although products and electronic commerce are presented in this and other examples presented, it should be understood that these are merely examples and that approaches presented in the present disclosure can relate to any appropriate types of objects or information as discussed and suggested elsewhere herein. In such an instance, the information aggregator service **516** can utilize the aggregated data from the matching service **510** to attempt to locate products, in a product data store **524** or other such location, which are offered through the marketplace and that match, or are otherwise related to, the potential match information. For example, if the identification service identifies a book in the captured image or video data, the information aggregator can attempt to determine whether there are any versions of that book (physical or electronic) offered through the marketplace, or at least for which information is available through the marketplace. In at least some embodiments, the information aggregator can utilize one or more suggestion algorithms or other such approaches to attempt to determine related elements that might be of interest based on the determined matches, such as a movie or audio tape version of a book. In some embodiments, the information aggregator can return various types of data (or metadata) to the environmental information service, as may include title information, availability, reviews, and the like. For facial recognition applications, a data aggregator might instead be used that provides data from one or more social networking sites, professional data services, or other such entities. In other embodiments, the information aggregator might instead return information such as a product identifier, uniform resource locator (URL), or other such digital entity enabling a browser or other interface on the client device **502** to obtain information for one or more products, etc. The information aggregator can also utilize the aggregated data to obtain various other types of data as well. Information for located matches also can be stored in a user data store **522** of other such location, which can be used to assist in determining future potential matches or suggestions that might be of interest to the user. Various other types of information can be returned as well within the scope of the various embodiments.

The matching service **510** can bundle at least a portion of the information for the potential matches to send to the client as part of one or more messages or responses to the original request. In some embodiments, the information from the identification services might arrive at different times, as different types of information might take longer to analyze, etc. In these cases, the matching service might send multiple messages to the client device as the information becomes available. The potential matches located by the various identification services can be written to a log data store **512** or other such location in order to assist with future matches or suggestions, as well as to help rate a performance of a given identification service. As should be understood, each service can include one or more computing components, such as at least one server, as well as other components known for

providing services, as may include one or more APIs, data storage, and other appropriate hardware and software components.

It should be understood that, although the identification services are shown to be part of the provider environment **506** in FIG. **5**, that one or more of these identification services might be operated by third parties that offer these services to the provider. For example, an electronic retailer might offer an application that can be installed on a computing device for identifying music or movies for purchase. When a user transfers a video clip, for example, the provider could forward this information to a third party who has software that specializes in identifying objects from video clips. The provider could then match the results from the third party with items from the retailer's electronic catalog in order to return the intended results to the user as one or more digital entities, or references to something that exists in the digital world. In some embodiments, the third party identification service can be configured to return a digital entity for each match, which might be the same or a digital different digital entity than will be provided by the matching service to the client device **502**.

FIG. **6** illustrates an example process **600** for segmenting an image, to locate a portion corresponding to an object of interest, that can be utilized in accordance with various embodiments. It should be understood that there can be additional, fewer, or alternative steps performed in similar or alternative orders, or in parallel, within the scope of the various embodiments unless otherwise stated. In this example, a first image is captured **602** using ambient light and a second image is captured **604** while using a flash or other source of illumination. It should be understood that the flash image could be captured first in other embodiments. The images could be captured by a camera associated with a computing device, or the computing device can obtain the images once captured by another device in other embodiments. A differential image is generated **606** using the first and second images, in order to suppress a significant amount of background data in many images. The intensity and/or color values of the differential image can be normalized **608**, such as by using the color or intensity data from the first, non-flash image. The normalized image then can be analyzed **610** using an appropriate algorithm, such as a connected components or computer vision algorithm, to determine the presence and/or approximate location of one or more objects in the normalized image. From the one or more objects, an object can be selected **612** that corresponds to the object of interest. As discussed, the selection can be a manual or automated process, or combination thereof. An outline (or edge, etc.) of the selected object can be determined **614**, and that outline can be used to extract **616** or determine a corresponding portion of the first or second image, or combination thereof. This portion then can be provided **618** to a process, system, or service for subsequent processing, such as to identify or recognize the object. Various other approaches can be utilized as well within the scope of the various embodiments.

FIG. **7** illustrates an example electronic user device **700** that can be used in accordance with various embodiments. Although a portable computing device (e.g., an electronic book reader or tablet computer) is shown, it should be understood that any electronic device capable of receiving, determining, and/or processing input can be used in accordance with various embodiments discussed herein, where the devices can include, for example, desktop computers, notebook computers, personal data assistants, smart phones, video gaming consoles, television set top boxes, and portable media players. In this example, the computing device **700** has a display screen **702** on the front side, which under normal

operation will display information to a user facing the display screen (e.g., on the same side of the computing device as the display screen). The computing device in this example includes at least one camera **704** or other imaging element for capturing still or video image information over at least a field of view of the at least one camera. In some embodiments, the computing device might only contain one imaging element, and in other embodiments the computing device might contain several imaging elements. Each image capture element may be, for example, a camera, a charge-coupled device (CCD), a motion detection sensor, or an infrared sensor, among many other possibilities. If there are multiple image capture elements on the computing device, the image capture elements may be of different types. In some embodiments, at least one imaging element can include at least one wide-angle optical element, such as a fish eye lens, that enables the camera to capture images over a wide range of angles, such as 180 degrees or more. Further, each image capture element can comprise a digital still camera, configured to capture subsequent frames in rapid succession, or a video camera able to capture streaming video.

The example computing device **700** also includes at least one microphone **706** or other audio capture device capable of capturing audio data, such as words or commands spoken by a user of the device. In this example, a microphone **706** is placed on the same side of the device as the display screen **702**, such that the microphone will typically be better able to capture words spoken by a user of the device. In at least some embodiments, a microphone can be a directional microphone that captures sound information from substantially directly in front of the microphone, and picks up only a limited amount of sound from other directions. It should be understood that a microphone might be located on any appropriate surface of any region, face, or edge of the device in different embodiments, and that multiple microphones can be used for audio recording and filtering purposes, etc.

The example computing device **700** also includes at least one orientation sensor **708**, such as a position and/or movement-determining element. Such a sensor can include, for example, an accelerometer or gyroscope operable to detect an orientation and/or change in orientation of the computing device, as well as small movements of the device. An orientation sensor also can include an electronic or digital compass, which can indicate a direction (e.g., north or south) in which the device is determined to be pointing (e.g., with respect to a primary axis or other such aspect). An orientation sensor also can include or comprise a global positioning system (GPS) or similar positioning element operable to determine relative coordinates for a position of the computing device, as well as information about relatively large movements of the device. Various embodiments can include one or more such elements in any appropriate combination. As should be understood, the algorithms or mechanisms used for determining relative position, orientation, and/or movement can depend at least in part upon the selection of elements available to the device.

FIG. **8** illustrates a logical arrangement of a set of general components of an example computing device **800** such as the device **700** described with respect to FIG. **7**. In this example, the device includes a processor **802** for executing instructions that can be stored in a memory device or element **804**. As would be apparent to one of ordinary skill in the art, the device can include many types of memory, data storage, or non-transitory computer-readable storage media, such as a first data storage for program instructions for execution by the processor **802**, a separate storage for images or data, a removable memory for sharing information with other devices, etc.

11

The device typically will include some type of display element **806**, such as a touch screen or liquid crystal display (LCD), although devices such as portable media players might convey information via other means, such as through audio speakers. As discussed, the device in many embodiments will include at least one image capture element **808** such as a camera or infrared sensor that is able to image projected images or other objects in the vicinity of the device. Methods for capturing images or video using a camera element with a computing device are well known in the art and will not be discussed herein in detail. It should be understood that image capture can be performed using a single image, multiple images, periodic imaging, continuous image capturing, image streaming, etc. Further, a device can include the ability to start and/or stop image capture, such as when receiving a command from a user, application, or other device. The example device similarly includes at least one audio capture component **812**, such as a mono or stereo microphone or microphone array, operable to capture audio information from at least one primary direction. A microphone can be a uni- or omni-directional microphone as known for such devices.

In some embodiments, the computing device **800** of FIG. **8** can include one or more communication elements (not shown), such as a Wi-Fi, Bluetooth, RF, wired, or wireless communication system. The device in many embodiments can communicate with a network, such as the Internet, and may be able to communicate with other such devices. In some embodiments the device can include at least one additional input device able to receive conventional input from a user. This conventional input can include, for example, a push button, touch pad, touch screen, wheel, joystick, keyboard, mouse, keypad, or any other such device or element whereby a user can input a command to the device. In some embodiments, however, such a device might not include any buttons at all, and might be controlled only through a combination of visual and audio commands, such that a user can control the device without having to be in contact with the device.

The device **800** also can include at least one orientation or motion sensor **810**. As discussed, such a sensor can include an accelerometer or gyroscope operable to detect an orientation and/or change in orientation, or an electronic or digital compass, which can indicate a direction in which the device is determined to be facing. The mechanism(s) also (or alternatively) can include or comprise a global positioning system (GPS) or similar positioning element operable to determine relative coordinates for a position of the computing device, as well as information about relatively large movements of the device. The device can include other elements as well, such as may enable location determinations through triangulation or another such approach. These mechanisms can communicate with the processor **802**, whereby the device can perform any of a number of actions described or suggested herein.

As an example, a computing device such as that described with respect to FIG. **7** can capture and/or track various information for a user over time. This information can include any appropriate information, such as location, actions (e.g., sending a message or creating a document), user behavior (e.g., how often a user performs a task, the amount of time a user spends on a task, the ways in which a user navigates through an interface, etc.), user preferences (e.g., how a user likes to receive information), open applications, submitted requests, received calls, and the like. As discussed above, the information can be stored in such a way that the information is linked or otherwise associated whereby a user can access the information using any appropriate dimension or group of dimensions.

12

As discussed, different approaches can be implemented in various environments in accordance with the described embodiments. For example, FIG. **9** illustrates an example of an environment **900** for implementing aspects in accordance with various embodiments. As will be appreciated, although a Web-based environment is used for purposes of explanation, different environments may be used, as appropriate, to implement various embodiments. The system includes an electronic client device **902**, which can include any appropriate device operable to send and receive requests, messages or information over an appropriate network **904** and convey information back to a user of the device. Examples of such client devices include personal computers, cell phones, handheld messaging devices, laptop computers, set-top boxes, personal data assistants, electronic book readers and the like. The network can include any appropriate network, including an intranet, the Internet, a cellular network, a local area network or any other such network or combination thereof. Components used for such a system can depend at least in part upon the type of network and/or environment selected. Protocols and components for communicating via such a network are well known and will not be discussed herein in detail. Communication over the network can be enabled via wired or wireless connections and combinations thereof. In this example, the network includes the Internet, as the environment includes a Web server **906** for receiving requests and serving content in response thereto, although for other networks an alternative device serving a similar purpose could be used, as would be apparent to one of ordinary skill in the art.

The illustrative environment includes at least one application server **908** and a data store **910**. It should be understood that there can be several application servers, layers or other elements, processes or components, which may be chained or otherwise configured, which can interact to perform tasks such as obtaining data from an appropriate data store. As used herein the term "data store" refers to any device or combination of devices capable of storing, accessing and retrieving data, which may include any combination and number of data servers, databases, data storage devices and data storage media, in any standard, distributed or clustered environment. The application server can include any appropriate hardware and software for integrating with the data store as needed to execute aspects of one or more applications for the client device and handling a majority of the data access and business logic for an application. The application server provides access control services in cooperation with the data store and is able to generate content such as text, graphics, audio and/or video to be transferred to the user, which may be served to the user by the Web server in the form of HTML, XML or another appropriate structured language in this example. The handling of all requests and responses, as well as the delivery of content between the client device **902** and the application server **908**, can be handled by the Web server **906**. It should be understood that the Web and application servers are not required and are merely example components, as structured code discussed herein can be executed on any appropriate device or host machine as discussed elsewhere herein.

The data store **910** can include several separate data tables, databases or other data storage mechanisms and media for storing data relating to a particular aspect. For example, the data store illustrated includes mechanisms for storing production data **912** and user information **916**, which can be used to serve content for the production side. The data store also is shown to include a mechanism for storing log or session data **914**. It should be understood that there can be many other aspects that may need to be stored in the data store, such as page image information and access rights information, which

can be stored in any of the above listed mechanisms as appropriate or in additional mechanisms in the data store 910. The data store 910 is operable, through logic associated therewith, to receive instructions from the application server 908 and obtain, update or otherwise process data in response thereto. In one example, a user might submit a search request for a certain type of element. In this case, the data store might access the user information to verify the identity of the user and can access the catalog detail information to obtain information about elements of that type. The information can then be returned to the user, such as in a results listing on a Web page that the user is able to view via a browser on the user device 902. Information for a particular element of interest can be viewed in a dedicated page or window of the browser.

Each server typically will include an operating system that provides executable program instructions for the general administration and operation of that server and typically will include computer-readable medium storing instructions that, when executed by a processor of the server, allow the server to perform its intended functions. Suitable implementations for the operating system and general functionality of the servers are known or commercially available and are readily implemented by persons having ordinary skill in the art, particularly in light of the disclosure herein.

The environment in one embodiment is a distributed computing environment utilizing several computer systems and components that are interconnected via communication links, using one or more computer networks or direct connections. However, it will be appreciated by those of ordinary skill in the art that such a system could operate equally well in a system having fewer or a greater number of components than are illustrated in FIG. 9. Thus, the depiction of the system 900 in FIG. 9 should be taken as being illustrative in nature and not limiting to the scope of the disclosure.

As discussed above, the various embodiments can be implemented in a wide variety of operating environments, which in some cases can include one or more user computers, computing devices, or processing devices which can be used to operate any of a number of applications. User or client devices can include any of a number of general purpose personal computers, such as desktop or laptop computers running a standard operating system, as well as cellular, wireless, and handheld devices running mobile software and capable of supporting a number of networking and messaging protocols. Such a system also can include a number of workstations running any of a variety of commercially-available operating systems and other known applications for purposes such as development and database management. These devices also can include other electronic devices, such as dummy terminals, thin-clients, gaming systems, and other devices capable of communicating via a network.

Various aspects also can be implemented as part of at least one service or Web service, such as may be part of a service-oriented architecture. Services such as Web services can communicate using any appropriate type of messaging, such as by using messages in extensible markup language (XML) format and exchanged using an appropriate protocol such as SOAP (derived from the "Simple Object Access Protocol"). Processes provided or executed by such services can be written in any appropriate language, such as the Web Services Description Language (WSDL). Using a language such as WSDL allows for functionality such as the automated generation of client-side code in various SOAP frameworks.

Most embodiments utilize at least one network that would be familiar to those skilled in the art for supporting communications using any of a variety of commercially-available protocols, such as TCP/IP, OSI, FTP, UPnP, NFS, CIFS, and

AppleTalk. The network can be, for example, a local area network, a wide-area network, a virtual private network, the Internet, an intranet, an extranet, a public switched telephone network, an infrared network, a wireless network, and any combination thereof.

In embodiments utilizing a Web server, the Web server can run any of a variety of server or mid-tier applications, including HTTP servers, FTP servers, CGI servers, data servers, Java servers, and business application servers. The server(s) also may be capable of executing programs or scripts in response requests from user devices, such as by executing one or more Web applications that may be implemented as one or more scripts or programs written in any programming language, such as Java®, C, C# or C++, or any scripting language, such as Perl, Python, or TCL, as well as combinations thereof. The server(s) may also include database servers, including without limitation those commercially available from Oracle®, Microsoft®, Sybase®, and IBM®.

The environment can include a variety of data stores and other memory and storage media as discussed above. These can reside in a variety of locations, such as on a storage medium local to (and/or resident in) one or more of the computers or remote from any or all of the computers across the network. In a particular set of embodiments, the information may reside in a storage-area network ("SAN") familiar to those skilled in the art. Similarly, any necessary files for performing the functions attributed to the computers, servers, or other network devices may be stored locally and/or remotely, as appropriate. Where a system includes computerized devices, each such device can include hardware elements that may be electrically coupled via a bus, the elements including, for example, at least one central processing unit (CPU), at least one input device (e.g., a mouse, keyboard, controller, touch screen, or keypad), and at least one output device (e.g., a display device, printer, or speaker). Such a system may also include one or more storage devices, such as disk drives, optical storage devices, and solid-state storage devices such as random access memory ("RAM") or read-only memory ("ROM"), as well as removable media devices, memory cards, flash cards, etc.

Such devices also can include a computer-readable storage media reader, a communications device (e.g., a modem, a network card (wireless or wired), an infrared communication device, etc.), and working memory as described above. The computer-readable storage media reader can be connected with, or configured to receive, a computer-readable storage medium, representing remote, local, fixed, and/or removable storage devices as well as storage media for temporarily and/or more permanently containing, storing, transmitting, and retrieving computer-readable information. The system and various devices also typically will include a number of software applications, modules, services, or other elements located within at least one working memory device, including an operating system and application programs, such as a client application or Web browser. It should be appreciated that alternate embodiments may have numerous variations from that described above. For example, customized hardware might also be used and/or particular elements might be implemented in hardware, software (including portable software, such as applets), or both. Further, connection to other computing devices such as network input/output devices may be employed.

Storage media and computer readable media for containing code, or portions of code, can include any appropriate media known or used in the art, including storage media and communication media, such as but not limited to volatile and non-volatile, removable and non-removable media imple-

15

mented in any method or technology for storage and/or transmission of information such as computer readable instructions, data structures, program modules, or other data, including RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disk (DVD) 5 or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by the a system device. Based on the disclosure and teachings provided 10 herein, a person of ordinary skill in the art will appreciate other ways and/or methods to implement the various embodiments.

The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense. It 15 will, however, be evident that various modifications and changes may be made thereunto without departing from the broader spirit and scope of the invention as set forth in the claims.

What is claimed is:

1. A computer-implemented method of identifying an object, comprising:

capturing a first image of an object of interest using a camera of a computing device, the first image captured 25 without illumination by an illumination source of the computing device;

capturing a second image of the object of interest using the camera, the second image being captured with the object 30 of interest being at least partially illuminated by the illumination source;

generating differential image data by determining a difference between intensity values of pixels for a first location of the second image, and intensity values of pixels 35 of a corresponding second location of the first image, wherein the intensity values of the pixels of the first image comprise a product of intensity of ambient light and a matte reflectance map;

determining a first pixel and a second pixel included in the differential image data;

determining that the first pixel and the second pixel are located in a region based at least in part on a first intensity value of the first pixel and a second intensity value of 40 the second pixel being similar;

determining a portion of the region by iteratively selecting 45 pixels in the differential image data until determining a respective pixel having a respective intensity value different than the first intensity value of the first pixel;

determining a corresponding portion of the first image based on the portion of the region; and

providing image data for the corresponding portion of the first image to an object identification process.

2. The computer-implemented method of claim 1, further comprising:

converting the first image to a first grayscale image before 55 the generating; and

converting the second image to a second grayscale image before the generating.

3. The computer-implemented method of claim 1, further comprising analyzing the differential image data using at 60 least one of a connected components algorithm or a computer vision algorithm.

4. A computer-implemented method, comprising:

obtaining a first image of an object of interest and a second 65 image of the object of interest as captured by a camera, the first image captured without illumination by an illumination source associated with the camera and the sec-

16

ond image captured with the object illuminated at least partially by the illumination source;

comparing intensity values for corresponding locations in the first image and the second image to generate differential image data, by determining a difference between intensity values of pixels of the second image, and intensity values of pixels of the first image, wherein the intensity values of the pixels of the first image comprise at least a product between an intensity of ambient light and a matte reflectance map;

determining that a first pixel in the differential image data is located in a region indicative of a potential object;

determining an edge of a portion of the region by iteratively selecting pixels in the differential image data until determining a respective pixel having a respective intensity value different than an intensity value of the first pixel; determining a shape of the potential object using at least the edge of the portion of the region; and

20 generating a result image including a portion of at least one of the first image or the second image, the portion corresponding to the shape of the potential object and a location of the potential object.

5. The computer-implemented method of claim 4, further comprising:

providing the result image as input to an object recognition process.

6. The computer-implemented method of claim 4, further comprising using at least one of a connected components algorithm or a computer vision algorithm to locate the potential object.

7. The computer-implemented method of claim 6, wherein the computer vision algorithm is one of a GrabCut, Watershed, or QuadTree algorithm.

8. The computer-implemented method of claim 4, wherein the first image is a first frame of video data and the second image is a second frame of the video data.

9. The computer-implemented method of claim 4, further comprising:

prompting a user of the camera to capture at least one of a new first image or a new second image when more than a threshold amount of movement of the camera occurred between a first capture time of the first image and a second capture time of the second image.

10. The computer-implemented method of claim 9, wherein the amount of movement is determined using at least one of an electronic gyroscope, an inertial sensor, an accelerometer, or an electronic compass associated with the camera.

11. The computer-implemented method of claim 4, wherein the illumination source is a camera flash element.

12. The computer-implemented method of claim 4, further comprising locating the potential object in the differential image data by identifying one or more object regions in the differential image data and selecting one of the one or more object regions.

13. The computer-implemented method of claim 12, further comprising:

selecting the potential object from one or more identified object regions.

14. The computer-implemented method of claim 13, wherein one of the one or more identified object regions is selected automatically based at least in part upon at least one of a location of one each of the object regions, a recognized shape of at least one of the object regions, a visible portion of each of the object regions in the image, and a viewable edge of each of the object regions in the image.

17

15. The computer-implemented method of claim 13, wherein comparing intensity values for corresponding locations in the first image and the second image to generate differential image data includes analyzing the first image and the second image to determine the corresponding locations for objects represented in the first and second images, and subtracting intensity values for the corresponding points in the second image from the corresponding points in the first image.

16. A computing device, comprising:

a processor;

a camera;

a camera flash element; and

a memory device including instructions that, when executed by the processor, cause the computing device to:

capture a first image of an object of interest, using the camera, without activating the camera flash element;

capture a second image of the object of interest, using the camera, with the camera flash element activated to at least partially illuminate the object of interest;

generate differential image data by, at least in part, determining a difference between intensity values of pixels of the second image, and intensity values of pixels of the first image, wherein the intensity values of the pixels of the first image comprise at least a product between an intensity of ambient light and a matte reflectance map;

determine that a first pixel is located in a region of the differential image data corresponding to the object of interest;

determine an outline of a portion of the region of the differential image data by iteratively selecting pixels in the differential image data until determining a respective pixel having a respective intensity value different than an intensity value of the first pixel and

generate a result image including a portion of at least one of the first image or the second image, the portion of the region of the differential image data corresponding to the outline and a location of the portion of the region of the differential image data.

17. The computing device of claim 16, wherein the instructions when executed further cause the computing device to:

provide the result image as input to an object recognition process.

18. The computing device of claim 16, further comprising using at least one of a connected components algorithm or a computer vision algorithm to determine the outline.

18

19. The computing device of claim 16, further comprising locating the portion of the region of the differential image data by identifying one or more object regions in the differential image data and selecting one of the one or more object regions.

20. A non-transitory computer-readable storage medium including instructions that, when executed by at least one processor of a computing device, cause the computing device to:

capture a first image of an object of interest using a camera of the computing device, the first image being captured without illumination by an illumination source of the computing device;

capture a second image of the object of interest using the camera of the computing device, the second image being captured with the object of interest at least partially illuminated by the illumination source;

generate differential image data by determining a difference between intensity values of pixels of the second image, and intensity values of pixels of the first image, wherein the intensity values of the pixels of the first image comprise at least a product between an intensity of ambient light and a matte reflectance map;

determine that a first pixel is located in a region indicative of the object in the differential image data;

determine an outline of a region by iteratively selecting pixels in the differential image data until determining a respective pixel having a respective intensity value different than a first intensity value of the first pixel;

use the outline to select a corresponding portion of the first image; and

provide image data for the corresponding portion to an object identification process.

21. The non-transitory computer-readable storage medium of claim 20, wherein the instructions when executed further cause the computing device to:

convert the first image to a first grayscale image before the generate differential image data; and

convert the second image to a second grayscale image before the generate differential image data.

22. The non-transitory computer-readable storage medium of claim 20, wherein the instructions when executed further cause the computing device to compare intensity values for each corresponding location in at least a portion of the first image and at least a portion of the second image to generate the differential image data.

* * * * *