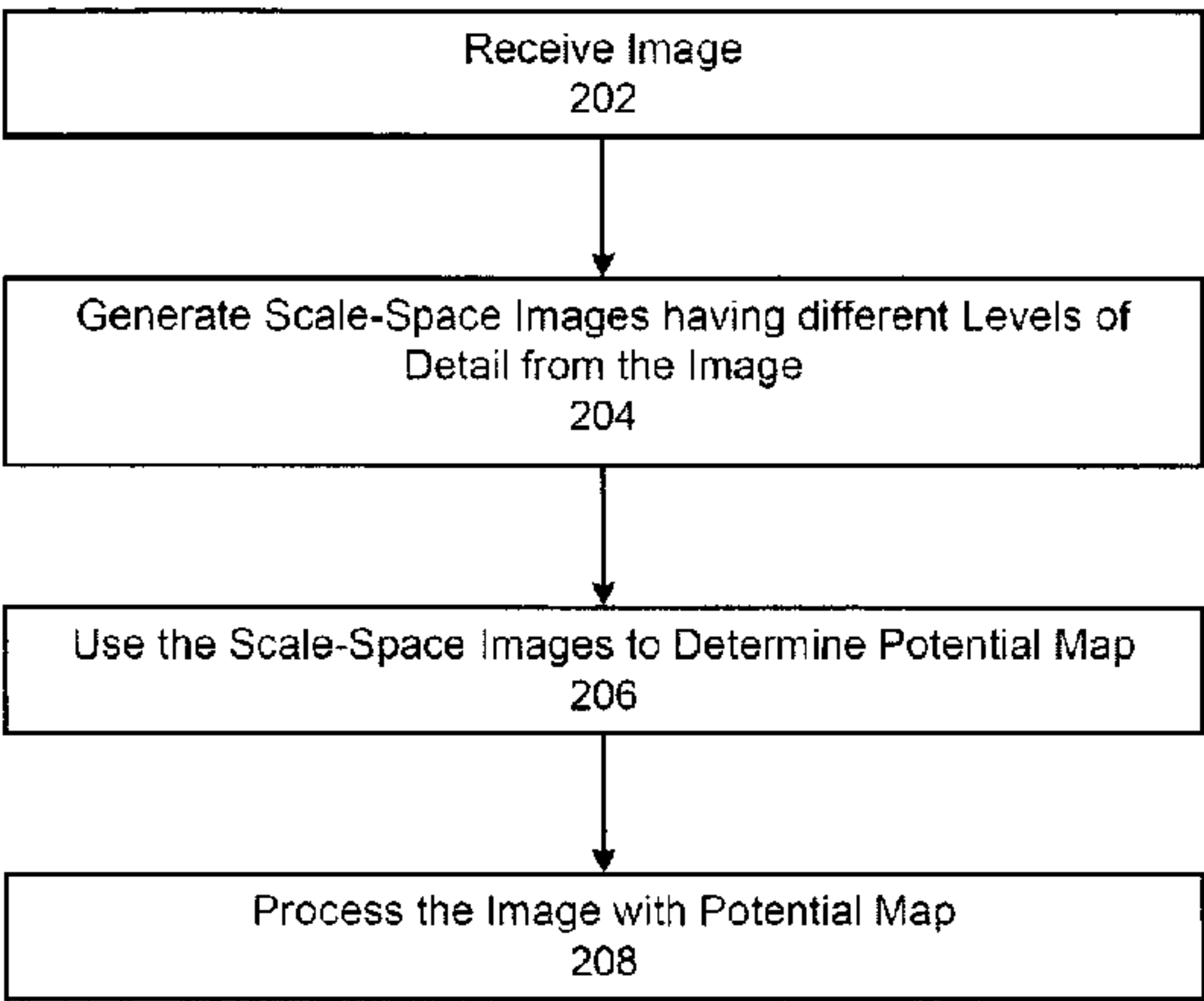




(86) Date de dépôt PCT/PCT Filing Date: 2009/12/11	(51) Cl.Int./Int.Cl. <i>G06T 5/50</i> (2006.01), <i>H04N 5/262</i> (2006.01)
(87) Date publication PCT/PCT Publication Date: 2010/06/17	
(45) Date de délivrance/Issue Date: 2018/07/17	(72) Inventeurs/Inventors:
(85) Entrée phase nationale/National Entry: 2011/06/01	EL-MARAGHI, THOMAS F., CA;
(86) N° demande PCT/PCT Application No.: IB 2009/007723	RZESZUTEK, RICHARD J., CA;
(87) N° publication PCT/PCT Publication No.: 2010/067191	ANDROUTSOS, DIMITRIOS, CA;
(30) Priorité/Priority: 2008/12/11 (US61/201,531)	PHAN, RAYMOND, CA;
	ZHOU, SAMUEL, CA
	(73) Propriétaire/Owner:
	IMAX CORPORATION, CA
	(74) Agent: BERESKIN & PARR LLP/S.E.N.C.R.L.,S.R.L.

(54) Titre : DISPOSITIFS ET PROCEDES DE TRAITEMENT D'IMAGES A L'AIDE DE REPRESENTATIONS ESPACE-ECHELLE
(54) Title: DEVICES AND METHODS FOR PROCESSING IMAGES USING SCALE SPACE



(57) **Abrégé/Abstract:**
Certain embodiments relate to processing images by creating scale space images from an image and using them to identify boundaries of objects in the image. The scale space images can have varying levels of detail. They are used to determine a potential map, which represents a likelihood for pixels to be within or outside a boundary of an object. A label estimating an object boundary can be generated and used to identify pixels that potentially may be within the boundary. An image with object boundaries identified can be further processed before exhibition. For example, the images may be two-dimensional images of a motion picture. Object boundaries can be identified and the two- dimensional (2D) images can be processed using the identified object boundaries and converted to three-dimensional (3D) images for exhibition.

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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
17 June 2010 (17.06.2010)



(10) International Publication Number
WO 2010/067191 A1

(51) International Patent Classification:

G06T 5/50 (2006.01) *H04N 5/262* (2006.01)
G06T 7/00 (2006.01)

(21) International Application Number:

PCT/IB2009/007723

(22) International Filing Date:

11 December 2009 (11.12.2009)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

61/201,531 11 December 2008 (11.12.2008) US

(71) Applicant (for all designated States except US): **IMAX CORPORATION** [CA/CA]; 2525 Speakman Drive, Mississauga, Ontario L5K 1B1 (CA).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **EL-MARAGHI, Thomas, F.** [CA/CA]; 148 Mill Street RR-3, Hawkeston, Ontario L0L 1T0 (CA). **RZESZUTEK, Richard, J.** [CA/CA]; 3 Warren Crescent, Toronto, Ontario M6S 4S1 (CA). **ANDROUTSOS, Dimitrios** [CA/CA]; 68 Nealon Ave., Toronto, Ontario M4K 1Z2 (CA). **PHAN, Ray-**

mond [CA/CA]; 2520-230 Queens Quay West, Toronto, Ontario (CA). **ZHOU, Samuel** [CA/CA]; 1510 Flamborough Circle, Mississauga, Ontario (CA).

(81) Designated States (unless otherwise indicated, for every

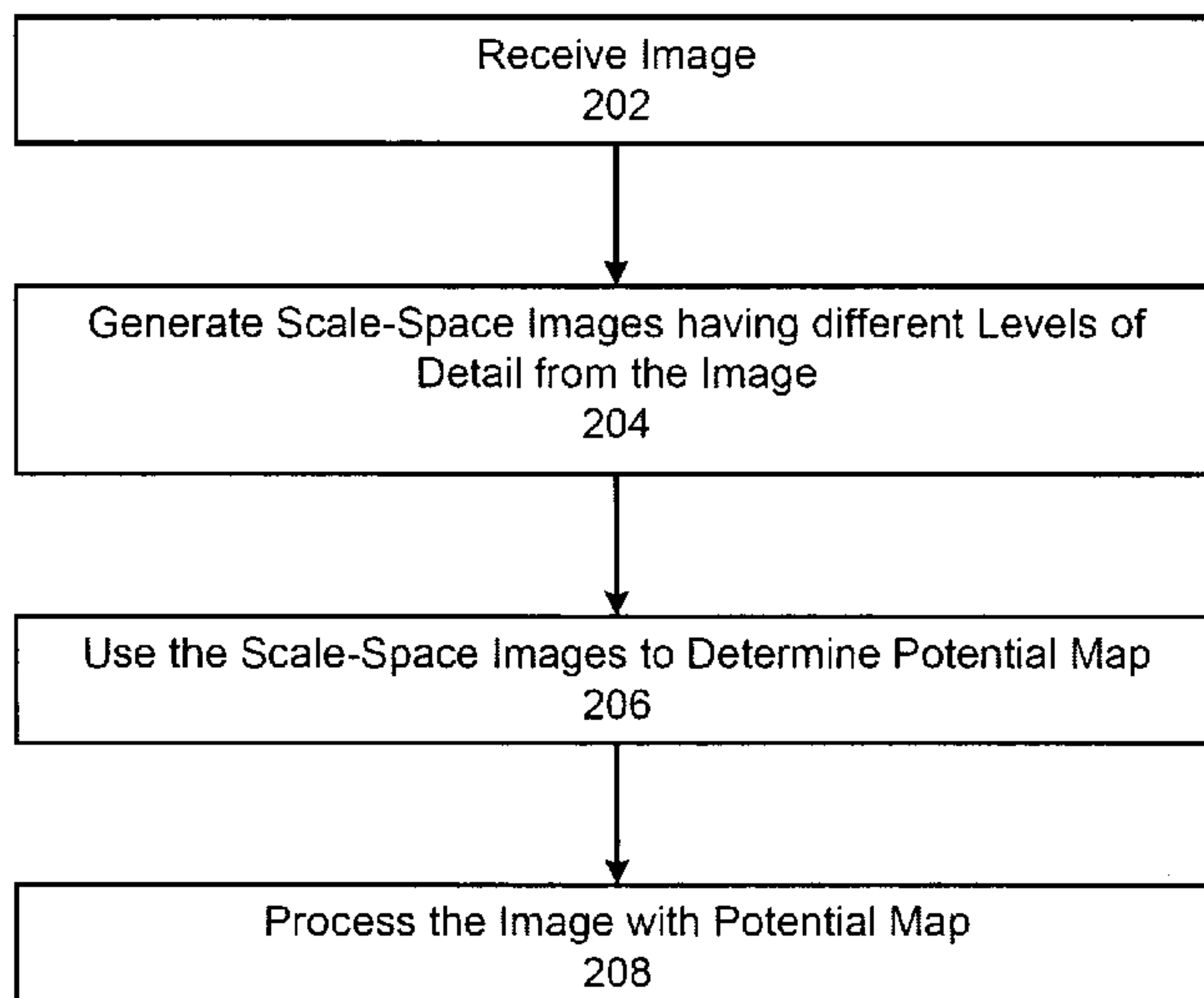
kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME,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

(84) Designated States (unless otherwise indicated, for every

kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (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, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: DEVICES AND METHODS FOR PROCESSING IMAGES USING SCALE SPACE



(57) Abstract: Certain embodiments relate to processing images by creating scale space images from an image and using them to identify boundaries of objects in the image. The scale space images can have varying levels of detail. They are used to determine a potential map, which represents a likelihood for pixels to be within or outside a boundary of an object. A label estimating an object boundary can be generated and used to identify pixels that potentially may be within the boundary. An image with object boundaries identified can be further processed before exhibition. For example, the images may be two-dimensional images of a motion picture. Object boundaries can be identified and the two-dimensional (2D) images can be processed using the identified object boundaries and converted to three-dimensional (3D) images for exhibition.

Figure 2

WO 2010/067191 A1



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))*

DEVICES AND METHODS FOR PROCESSING IMAGES USING SCALE SPACE

[0001]

Field of the Disclosure

[0002] This disclosure relates generally to image processing and, more particularly, to processing images using scale space representations of the image.

Background

[0003] Processing images for motion pictures or otherwise can include identifying objects in one or more frames. The objects can be identified by determining object boundaries. An object boundary can be identified using rotoscoping. Rotoscoping includes tracing boundaries of objects in a motion picture frame-by-frame via digital means. Rotoscoping can extract digital mattes to allow special effects and other image processing to be applied to the image. Examples of special effects include replacing objects in a scene with other objects generated via a computer, compositing an actor from one scene to another, and changing a two-dimensional (2D) motion picture to a three-dimensional (3D) motion picture. Examples of other image processes include tracking an object in an image sequence and tracking an object in a left and right eye image sequence.

[0004] Rotoscoping can be used when other techniques, such as a blue screen method, fail to produce a matte within an acceptable accuracy tolerance. For example, images may include a complex background, such as images using archival footage, for which a blue screen method may fail to produce acceptable results. Rotoscoping can be labor-intensive. Semiautomatic rotoscoping techniques can be used for efficiency. Such techniques include contour-based methods and alpha-channel algorithms.

[0005] Contour-based methods can involve a user that specifies a contour in one or more frames and as accurately as possible. The contour is a rough estimate of an object boundary. An energy function is evaluated and an active contour is associated with the boundary based on the rough estimate contour. The energy function is minimized iteratively, producing an optimal contour around the object. In some contour-based methods, stroke matching is performed that includes analyzing cost functions to determine which strokes match certain contours of objects between key frames. The algorithms can output relatively smooth contours and establish correspondence between contours in neighboring frames, but often require a skilled user to delineate an object of interest.

[0006] Alpha-channel algorithms can extract soft boundaries of objects by analyzing three regions of color: foreground with respect to the object, background with respect to the object, and blended foreground and background in an intermediate region along a boundary of the object. Alpha-channel algorithms often are applied to individual frames, although application to sequences is also possible.

[0007] Alpha-channel algorithms can extract soft boundaries, as well as the alpha value, or transparency, of the value. In some alpha-channel algorithms, a Bayesian approach is applied that models both the foreground and background color distributions with spatially varying sets of Gaussians and that assumes a fractional blending of the foreground and background colors to produce a final output. Other alpha-channel algorithms assume that a clean foreground color is a linear combination of a set of clusters and calculate the color and alpha values by examining pairs of clusters in the foreground and the background.

[0008] Still other alpha-channel algorithms perform one or more of: (i) estimate the alpha matte in high resolution images and image sequences by assuming that the clusters are prolate, or cigar-shaped, in the red, green, blue (RGB) color space; (ii) derive a cost function from local smoothness assumptions on foreground and background colors and obtaining a quadratic cost function in terms of alpha by analytically eliminating foreground and background colors; (iii) derive a partial differential equation that relates the

gradient of an image to the alpha values and describe an efficient algorithm, providing the alpha values as the solution of the equation; (iv) formulate the problem of natural image matting as one of solving Poisson equations with the matte gradient field and extract mattes using a pair of flash/no-flash images, referred to as “flash matting”; (v) allow construction of environment mattes “on-the-fly,” without a need for specialized calibration; (vi) perform environment matting by capturing a description of how the object refracts and reflects light, in addition to capturing foreground objects and the alpha matte, and by placing the foreground object in a new environment using environment compositing. Such alpha-channel algorithms, however, seek to produce soft segmentations without producing hard segmentations, which can result in segmentation that is more accurate.

[0009] Another rotoscoping technique is random walks. Random walks is a graphical image segmentation algorithm that attempts to identify a probability that a random walker, starting at some “seed” pixel and traveling to some “sink” pixel, would cross a particular pixel. Edges between pixels can be weighted such that pixels considered similar by some criteria have low edge weights, making it more likely for the walker to cross that edge. Probabilities can be determined as a solution to a combinatorial Dirichlet problem. Random walks can also use Locality Preserving Projections to transform a colorspace so that similar colors, such as those in slow-varying gradients, can be brought together and dissimilar colors can be moved apart. Random walks, however, may be unable to segment images cleanly in the presence of noise, resulting in inaccurate object boundary identifications.

[0010] Accordingly, methods, devices, and systems are desired that can identify boundaries of objects efficiently and accurately. Methods, devices and systems are also desirable that can produce hard segmentations and identify object boundaries accurately in the presence of noise.

Summary

[0011] Certain aspects and embodiments relate to processing images by creating scale space images from an image and using the scale space images to identify boundaries of objects in the image. Scale space images are a multi-resolution signal representation of an image. The scale space

images represent varying levels of detail of the image. An image having object boundaries identified can be further processed before exhibition. For example, the images may be two-dimensional images of a motion picture. Object boundaries can be identified and the two-dimensional (2D) images can be processed using the identified object boundaries to convert the 2D images to three-dimensional (3D) images.

[0012] In an embodiment, an image having at least one object is received. A computing device can generate two or more scale space images from the image. The scale space images can be used to determine a potential map. The potential map can represent a likelihood of whether a pixel is within a boundary of the object or outside the boundary of the object. The potential map can be used to identify the boundary of the object.

[0013] In some embodiments, the image is converted to a color model. The color model can be a CIE $L^*a^*b^*$ color space.

[0014] In some embodiments, the scale space images are generated from the image by converting the image to a scale space using two or more low-pass filters. The low-pass filters can include Gaussian kernels. The different levels of detail can include different degrees of blur.

[0015] In some embodiments, the scale space images are generated from the image by converting the image to scale space using one of (i) two or more wavelet filters, or (ii) an edge persevering decomposition process.

[0016] In some embodiments, the scale space images can be used to determine, for each pixel of the image, weights. Each weight can be associated with a link. The weights can be determined by determining the links associated with the pixel. A weight for each link associated with the pixel can be determined. The weight for each link can be collected to form the weights for the pixel.

[0017] In some embodiments, a label for the image is received. Potential values can be determined from the weights and using the label. Each potential value can represent a likelihood of an associated pixel being within a boundary of the object or being outside the boundary of the object. The potential map can be determined from the potential values by determining

a geometric mean the potential values. The potential map can include the geometric mean for the potential values.

[0018] In some embodiments, a label can be generated and used to determine the potential map. An object mask for the image is received. An inverted object mask is computed from the object mask for the image. A first distance transform is determined from the inverted object mask. A second distance transform is determined from the object mask. Foreground pixels in the image are identified using the first distance transform. Background pixels in the image are identified using the second distance transform. The label is generated based on the identified foreground pixels and the identified background pixels.

[0019] In some embodiments, a label can be generated from an initial potential map. An object mask for the image is received. An inverted object mask from the object mask for the image is computed. The inverted object mask is shrunk using a morphological thinning process. The object mask for the image is shrunk using the morphological thinning process. An initial label is generated based on the shrunk inverted object mask and the shrunk object mask for the image. An initial potential map for the image is determined using the initial label. The label is generated using the initial potential map, the shrunk inverted object mask, and the shrunk object mask.

[0020] In some embodiments, the potential map is used to generate an image mask. Two or more key points identifying an estimated boundary of the object are received. A label is computed based on the key points. An image segment is cropped based on the label. A potential map is determined from the image segment. Boundary points are created from the potential map. If a command is received that identifies the boundary points as being unacceptable, a second potential map is computed using new key points. If a command is received that identifies the boundary points as being acceptable, the boundary points are outputted. The image mask can be generated using the boundary points.

[0021] In some embodiments, the new key points are a greater number than the key points. Furthermore, the key points identify the estimated boundary of the object in two or more image frames in some embodiments.

The boundary points identify the portion of the estimated boundary of the object in one or more image frames located between the two or more image frames.

[0022] In some embodiments, the key points identify the estimated boundary of the object in two or more image frames. The first set of new points identify the portion of the estimated boundary of the object in at least one image frame located between the two or more image frames.

[0023] In some embodiments, the potential map can be determined by program code stored on a computer-readable medium.

[0024] In some embodiments, the potential map can be determined by a scale space engine stored on a computer-readable medium and executed by a processor of a computing device.

[0025] These illustrative embodiments are mentioned not to limit or define the disclosure, but to provide examples to aid understanding thereof. Additional embodiments are discussed in the Detailed Description, and further description is provided there. Advantages offered by one or more of the various embodiments may be further understood by examining this specification or by practicing one or more embodiments presented.

Brief Description of the Drawings

[0026] Figure 1 is a block diagram of a system for generating a potential map that can be used to process an image according to one embodiment of the present invention.

[0027] Figure 2 is a flow diagram of a method for generating a potential map with which to process an image according to one embodiment of the present invention.

[0028] Figure 3 is a flow diagram of method for generating a potential map with which to process an image according to a second embodiment of the present invention.

[0029] Figure 4 is an illustration of a graph for a pixel of an image based on scale space images according to one embodiment of the present invention.

[0030] Figure 5 is a flow diagram of a method for determining a label from an image based on an object mask according to one embodiment of the present invention.

[0031] Figure 6 is a flow diagram of a method for determining a label from an image based on an object mask according to a second embodiment of the present invention.

[0032] Figures 7A-7D are illustrations of determining a boundary of an object in an image according to one embodiment of the present invention.

[0033] Figure 8 is a flow diagram of a method for determining object boundary points based on key points using a potential map according to one embodiment of the present invention.

Detailed Description

[0034] Certain aspects and embodiments relate to processing images by creating scale space images from an image and using the scale space images to identify boundaries of objects in the image. The scale space images may have varying levels of detail. An image having object boundaries identified can be further processed before exhibition. For example, the images may be two-dimensional images of a motion picture. Object boundaries can be identified and the two-dimensional (2D) images can be processed using the identified object boundaries to convert the 2D images to three-dimensional (3D) images.

[0035] Scale space images are a multi-resolution signal representation of an image. Scale space images can be formed by filtering the image using filters of varying characteristics, such as different filter kernel sizes. Scale space images formed using filters of varying characteristics can have varying levels of detail. In some embodiments, scale space images are formed by convolving an image with Gaussian kernels having different sizes. Scale space images can correspond with levels. For example, a scale space image can correspond to a level that represents a level of detail in the image.

[0036] In some embodiments, scale space images are formed by filtering the image multiple times using different sized filtering components to remove information related to "fine" details in the image. The filtering components may be low-pass filter kernels that have progressively larger

sizes. A scale space image generated using a large-sized kernel filter can be a higher level scale space image that includes a lower level of detail. A scale space image generated using a smaller size kernel filter is a lower level scale space image that includes a higher level of detail. In one embodiment, the filters kernels are isometric Gaussian low-pass filter kernels and the resulting scale space images have varying characteristics that include blur.

[0037] In some embodiments, the scale space images can be used to compute a three-dimensional graph for each pixel. The pixel can be linked to adjacent pixels of a number of scale space images. A weight can be associated with a link between the pixel and an adjacent pixel. A value of the weight can be determined by the similarity between the pixels.

[0038] The scale space images can be used to determine a potential map for the image. For example, the three-dimensional graph can be used to compute a potential map. A potential map may include, for each pixel in the image or a portion of the image, a potential value representing a likelihood of the pixel being within a boundary of an object in the image or being outside the boundary of the object. The potential map can be used to process the image. For example, the potential map can be used to identify a boundary for an image object to allow the object to be modified for 3D exhibition, among other purposes.

[0039] In some embodiments, a label for the image can be received to facilitate computing the potential map. A label can identify image pixels that can be candidates for boundary pixels of an object. A label can be associated with a boundary tolerance to ensure true object boundary pixels are contained within the label. A label can be computed based on imprecise boundary pixel information. In some embodiments, a label is received from a human operator using interactive means. An example of a label is a trimap that is drawn around, and that includes, the boundary of an object. A trimap divides image pixels into three groups: pixels that belong to the object (foreground), pixels that are outside the object (background) and pixels between the foreground and the background that may be object boundary pixels, but undetermined. A potential map can be computed using the label.

[0040] In some embodiments, the potential map can be computed by applying a random walks algorithm to scale space images. The resulting potential map is used to identify an object boundary. The potential map can be used with the random walks algorithm to improve object boundary identification in the presence of noise.

[0041] These illustrative examples are given to introduce the reader to the general subject matter discussed here and are not intended to limit the scope of the disclosed concepts. The following sections describe various additional embodiments and examples with reference to the drawings in which like numerals indicate like elements.

Illustrative System Implementation

[0042] Figure 1 depicts a system that is capable of determining a potential map using scale space images generated from the image and of allowing the potential map to be used to process the image according to certain embodiments. Other embodiments may be utilized. The system includes a computing device 102 having a processor 104 that can execute code stored on a computer-readable medium, such as a memory 106, to cause the computing device 102 to determine the potential map using scale space images generated from the image. The computing device 102 may be any device that can process data and execute code that is a set of instructions to perform actions. Examples of the computing device 102 include a desktop personal computer, a laptop personal computer, a server device, a handheld computing device, and a mobile device.

[0043] Examples of the processor 104 include a microprocessor, an application-specific integrated circuit (ASIC), a state machine, or other suitable processor. The processor 104 may include one processor or any number of processors. In some embodiments, the processor 104 includes a Graphics Processing Unit (GPU) associated with a high-end graphics card with 1 GB or less of VRAM. In other embodiments, the processor 104 is a multi-core processors that include two, four, or more processing units. The multi-core processors may include single-instruction, multiple-data (SIMD) compatibilities, such as Streaming SIMD Extensions (SSE) and 3DNow!.

Linear algebra packages, such as LAPACK, can be used to use capabilities of SIMD extensions and processors advantageously.

[0044] The processor 104 can access code stored in the memory 106 via a bus 108. The memory 106 may be any tangible computer-readable medium capable of storing code. The memory 106 can include electronic, magnetic, or optical devices, capable of providing processor 104 with executable code. Examples of the memory 106 include random access memory (RAM), read-only memory (ROM), a floppy disk, compact disc, digital video device, magnetic disk, an ASIC, a configured processor, or other storage device capable of tangibly embodying code. The bus 108 may be any device capable of transferring data between components of the computing device 102. The bus 108 can include one device or multiple devices.

[0045] The computing device 102 can share data with additional components through an input/output (I/O) interface 110. The I/O interface 110 can include a USB port, an Ethernet port, a serial bus interface, a parallel bus interface, a wireless connection interface, or any suitable interface capable of allowing data transfers between the computing device and another component. The additional components can include a user interface (UI) device 112, a display 114, and a network 116. The UI device 112 can include a keyboard, a mouse device, a touch screen interface, or other device capable of receiving commands from a user and providing the commands to the computing device 102. The display 114 can include a liquid crystal display (LCD), a plasma screen, cathode ray tube (CRT), or any device capable of displaying images generated by the computing device 102. The network 116 can include the internet, an intranet, wide area network (WAN), local area network (LAN), virtual private network (VPN), or any suitable communications network that allows computing device 102 to communicate with other components. In other embodiments, the computing device 102 is an offline device capable of performing various methods according to various embodiments of the present invention in an offline manner.

[0046] Instructions can be stored in the memory 106 as executable code. The instructions can include processor-specific instructions generated by a compiler and/or an interpreter from code written in any suitable computer-

programming language, such as C, C++, C#, Visual Basic, Java, Python, Perl, JavaScript, and ActionScript. The instructions can include a image processing application 118 that, when executed by the processor 104, can cause the computing device 102 to determine a potential map using scale space images generated from the image and to use the potential map to process the image. The image processing application 118 includes a scale space engine 120 that, when executed with the image processing application 118 by the processor 104 can cause the computing device 102 to generate scale space images from the image to determine the potential map from the scale space images, and to use the potential map to process the image.

[0047] This exemplary system configuration is provided merely to illustrate a potential configuration that can be used to implement certain embodiments. Other configurations may of course be utilized.

Exemplary Methods of Generating a Potential Map

[0048] Potential maps used to process images can be generated using a variety of methods according to various embodiments of the present invention. Figure 2 illustrates one embodiment of method for generating a potential map from scale space images. The method illustrated in Figure 2 is described with reference to the system configuration of Figure 1. However, other system implementations are possible.

[0049] In block 202, the image processing application 118 receives an image. The image may be, for example, an image of a motion picture that is a sequence of images. In some embodiments, the image includes one or more objects. Objects may be tangible items that are visually represented in the image. The image may be an RGB (red, green, blue,) image or in a different color space, such as YUV, XYZ, or CIE L*a*b* color space.

[0050] In block 204, the scale space engine 120 generates scale space images having different levels of detail from the image. Each scale space image can have a different level of detail as compared to the other scale space images generated. The scale space engine 120 can generate scale space images using a variety of methods. One method includes using filters with different kernel sizes to filter the image. Each filter can produce a scale space image that has a certain level of detail that is different than other scale

space images produced by different sized filters. The filters may be implemented as software, for example as part of the scale space engine 120. In other embodiments, the filters are implemented in hardware that the scale space engine 120 accesses or controls.

[0051] In some embodiments, the scale space images may be generated after the scale space engine 120 converts the image to scale space. An image can be converted to scale space using, for example, wavelet filters or an edge persevering decomposition process.

[0052] In block 206, the scale space engine 120 uses the scale space images to determine a potential map for the image. A potential map can include potential values associated with the pixels or part of the pixels of the image. For example, a potential value can be associated with a pixel of the image. The potential value can represent a likelihood of the associated pixel being within a boundary of an object in the image or being outside the boundary of the object. In some embodiments, a potential value is determined for a pixel based on weights that are associated with links between the pixel and neighboring pixels. The links and associated weights can be identified and can be determined using the scale space images.

[0053] In block 208, the image processing application 118 processes the image using the potential map. For example, the image processing application can generate an image mask for the image using the potential map. The image mask can be used to identify objects in the image. In some embodiments, the object location in the image can be modified after using the potential map to identify the object boundary.

[0054] Figure 3 depicts another embodiment of a method for generating potential maps for use in processing images. The method of Figure 3 is described with reference to the system of Figure 1 and graphical illustration of Figure 4. However, other implementations are possible.

[0055] In block 302, the image processing application 118 receives an image that includes at least one object. The image processing application 118 can receive the image as in block 202 of Figure 2.

[0056] In block 304, the scale space engine 120 converts the image to a color model. Converting to a color model may allow the scale space engine

120 to discriminate between colors of the image more easily. In some embodiments, the image is an RGB image that is converted to a color model that is the image in a CIE L*a*b* color space.

[0057] In block 306, the scale space engine 120 generates at least two scale space images from the color model. The scale space images can have different levels of detail. Each scale space image has a level of detail that is different from the other scale space images. Scale space images can be performed using a variety of methods. In one embodiment, Gaussian low-pass filter (LPF) kernels of progressively larger sizes are used to filter the image to generate scale space images. Using Gaussian kernels may help prevent undesirable structures from being introduced into the image, particularly for filters of relatively large size. The scale space images generated by the Gaussian kernels may have different levels of detail that include different amounts of image blur.

[0058] Scale space, ζ , with N-levels can be represented as:

$$\zeta = \{I[x, y] * f(x, y | \sigma_n) : \sigma_n = 0, 1, 2, 4, \dots, 2^{N-1}\}$$

where,

$I[x, y]$ is an initial image; and

$$f(x, y | \sigma_n) = \frac{1}{\sqrt{2\pi\sigma_n^2}} \exp\left\{-\frac{(x^2 + y^2)}{2\sigma_n^2}\right\}, \text{ which is a Gaussian}$$

kernel of size n.

[0059] From the N scale space images in the scale space, ζ , the scale space engine 120 can generate a three-dimensional graph for each pixel of an image or a portion of the image. The graph can illustrate the relationship between pixels of different levels of the scale space images and can illustrate the relationship between a pixel of a certain level and a neighboring pixel.

[0060] Figure 4 illustrates a graph 402 generated from scale space, ζ , according to one embodiment of the present invention. The graph includes different layers 404a-c. Although three layers 404a-c are shown, any number of layers can be implemented. Each layer of layers 404a-c may correspond to a scale space image generated from the image. Each node in a layer can correspond to a pixel in a corresponding scale space image. For example,

layer 404a can correspond to a first scale space image, layer 404b can correspond to a second scale space image, and layer 404c can correspond to a third scale space image.

[0061] Furthermore, the layers 404a-c can be arranged from a greater level of detail to a least level of detail. For example, layer 404a can correspond to the first scale space image that has the greatest level of detail, such as a fine level of detail, among the first scale space image, a second scale space image, and a third scale space image. Layer 404b can correspond to the second scale space image that has a medium level of detail among the first scale space image, the second scale space image, and the third scale space image. Layer 404c can correspond to the third scale space image that has the lowest level of detail, which can be referred to as the course level, among the first scale space image, the second scale space image, and the third scale space image.

[0062] The graph 402 in Figure 4 is a six-connected structure that includes pixels 406. Other types of connected structures, such as eight-connected structures, or otherwise, can be implemented. Each pixel of pixels 406 corresponds to a scale space image and can be connected to six corresponding pixels, such as four neighboring pixels in the same layer and corresponding pixels from a higher layer and a lower layer. Each pair of connected pixels can have a link between the connected pixels. Link 410, for example, is located between pixel 408 on layer 404a and pixel 406 on layer 404b. This arrangement can allow each layer to influence other layers such that detail can be retained at higher layers and effects from noise can be controlled at lower layers.

[0063] In block 308, the scale space engine 120 uses the scale space images to determine, for each pixel of the image, links and weight for each link. For example, each link in Figure 4 represents the connection between a pixel and a neighboring pixel, or a corresponding pixel in another layer of a potential map, and the links may be associated with weights. A weight for a link can be determined using the following relationship:

$$G_{i,j} = \frac{2}{1 + \exp(\beta \|\bar{c}_i - \bar{c}_j\|)}$$

where,

$G_{i,j}$ is a weight for a link between pixel i and j ;

$\vec{c}_i$ is a color vector representing pixel i ;

$\vec{c}_j$ is a color vector representing pixel j , which is adjacent to pixel i ; and

β is a free parameter that, in some embodiments, is set to a value of ninety.

[0064] In block 310, the scale space engine 120 receives a label for pixels. A label may represent an estimation of a boundary of an object for an image. In some embodiments, a label is received through a user interface from an individual that is inputting commands via an input device to identify an estimation of a boundary for an object. In other embodiments, a label is received by the scale space engine 120 when the scale space engine 120 generates the label as discussed below, for example, with reference to Figures 5 and 6. The label may be used by the scale space engine 120 to identify pixels that potentially may be a boundary of an object. For example, the undetermined pixels may be identified by the label and then further processed to determine further the pixels that are associated with the boundary of the object.

[0065] In block 312, the scale space engine 120 determines potential values from the weights and by using the label. The potential values can be associated with a pixel and can represent a likelihood of the pixel being inside or outside of a boundary of the object. In some embodiments, the label is used to identify the pixels for which a potential value is to be determined. The potential value for a pixel may be determined using the following relationship:

$$L\vec{v} = \vec{b}$$

where,

$\vec{v}$ is a vector potential value to be determined;

$\vec{b}$ is a boundary vector that defines the boundary conditions of the system; and

L is a Laplacian matrix in which each element is determined by the weights in a graph.

[0066] The Laplacian matrix, L , can be determined using weights, $G_{i,j}$, of a graph, such as the graph of Figure 4, using the following relationship:

$$L_{i,j} = \begin{cases} \sum_{k=1}^n G_{i,k}, & i = j, \\ -G_{i,j}, & i \neq j, \text{ where node } i \text{ is adjacent to node } j, \\ 0, & \text{elsewhere.} \end{cases}$$

The weights, $G_{i,j}$, can be determined as described with reference to Figure 4.

[0067] The Laplacian matrix, L , can be decomposed to the following form:

$$L = \begin{bmatrix} L'' & 0 & 0 \\ 0 & L^b & 0 \\ 0 & 0 & I \end{bmatrix}$$

where,

L'' is a sub-matrix of a Laplacian matrix that includes rows and columns associated with undetermined pixels;

L^b is a sub-matrix that includes the boundary pixels, but not the undetermined pixels;

I is the identity matrix representing pixels that are assigned as a source or sink pixels that may not affect a solution; and

0 is a "zero matrix," a matrix that includes zeros only.

[0068] Potential values can therefore be determined using the following relationship:

$$\vec{v} = (L'')^{-1} \vec{b}$$

The potential values can form a potential map. The potential values for a pixel (x, y) can be denoted by $P(x, y)$.

[0069] The result of solving for the potential values based on an N level scale space images can be a new scale space, Π , represented by the following:

$$\Pi = (P(x, y | n) : n = 0, 1, 2, 3, \dots, (N-1))$$

where,

n is an index for a particular level in an N level scale space; and

$P(x, y | n)$ is a potential map for the n th level.

[0070] In block 314, the scale space engine 120 determines a potential map from the potential values. The potential values can be represented by the N level scale space, Π .

[0071] In some embodiments, a final potential map, $P(x, y)$, is determined by performing a geometric mean on each level of the scale space, as represented by the following relationship:

$$P(x, y) = \left(\prod_{i=0}^{N-1} P(x, y | i) \right)^{1/N}$$

[0072] In some embodiments, an arithmetic mean may be used instead of a geometric mean. However, the geometric mean can be more effective than an arithmetic mean

result in the result being incorrect as compared to an actual tendency. In some embodiments, the result details at finer scale space images are retained, and areas that are fuzzy in the course scale space images are removed.

[0073] Furthermore, a geometric mean can result in a potential map that removes dependency between different levels established through layer-to-layer linkages in scale space.

[0074] In some embodiments, the potential map is locally smooth. For example, regions that are the same or similar have an overall gradient that does not change abruptly. Strong edges include abrupt gradient changes that can assist in identifying such strong edges. Furthermore, the potential map can eliminate small variations (i.e. noise) in the image at higher levels in the scale space.

[0075] In block 316, the image processing application 118 uses the potential map to generate an image mask for the image. An image mask may be a representation of an image that can be used to identify image object boundaries. The potential map can be used, for example, to identify boundary pixels of an object. An image mask can be generated using the identified object boundary pixels. In some embodiments, the scale space engine 120 may use a threshold to determine a digital value for each pixel. For example, if the potential value, $P(x, y)$, is greater than or equal to 0.5, a digital "one" can be assigned to the pixel. If the potential value, $P(x, y)$, is less than 0.5, a

digital “zero” can be assigned to the pixel. The threshold of 0.5 represents an equal likelihood that a pixel is a foreground pixel or is a background pixel. Any pixel with a probability that is greater than 0.5 can therefore be considered a foreground pixel, represented by a digital “one.” Any pixel with a probability that is less than 0.5 can therefore be considered a background pixel, represented by a digital “zero.”

Exemplary Methods of Generating a Label

[0076] Labels according to some embodiments of the present invention can be generated from the image. For example, the scale space engine 120 can receive an object mask for the image and use the object mask to determine a label. An object mask can be a rough estimate of an object by estimating pixels that are associated with the boundary of an object, which can include imprecise pixel designations as boundary pixels. An object mask can be received by the scale space engine 120. Figures 5 and 6 illustrate embodiments of methods for receiving a label in block 310 by generating the label. The embodiments for generating a label are described with reference to the system of Figure 1. Other system implementations, however, are possible.

[0077] Furthermore, the embodiment depicted in Figure 5 is described with reference to the illustrations in Figures 7A-7D.

[0078] In block 502, the scale space engine 120 receives an object mask for an image. In some embodiments, the object mask can identify a portion of the image that is larger than the object of interest. Furthermore, an object mask can delineate more than one object, which may require the object of interest to be separated before further processing. Figure 7A depicts an example of an object mask for object 702. The object 702 shown is a pear in an image that has foreground pixels represented using white color and background pixels represented using black color.

[0079] In block 504, the scale space engine 120 inverts the object mask to generate an inverted object mask. In an inverted object mask, background pixels become foreground pixels and vice versa. The object mask can be inverted by changing values of high-value pixels to low and vice versa. For example, an inverted object mask of the object mask in Figure 7A may include

pixels that make up the object being designated by the color black and all other pixels being designated by the color white.

[0080] In block 506, the scale space engine 120 determines a distance transform for the inverted object mask. A distance transform may be a representation that indicates, for each background pixel, the distance to the nearest boundary pixel. Figure 7B depict an example of a distance transform for the inverted object mask of the object 702. In Figure 7B, the pixels that represent the object 702 are background pixels, and the background pixels that are closest to a boundary pixel are darker than those background pixels that are further away.

[0081] In block 508, the scale space engine 120 determines a distance transform for the object mask. A distance transform for the object mask may appear as the opposite as the image in Figure 7B. The pixels that represent the object 702 become foreground pixels. The pixels in the background that are closest to a boundary pixel may be darker than those pixels that are further away.

[0082] In block 510, the scale space engine 120 identifies foreground pixels in the image using the distance transform for the inverted object mask. In some embodiments, the value of the distance transform for a pixel of the inverted object mask is compared to the first value that is a boundary tolerance. If the value of the distance transform for the pixel is greater than the first value that is a boundary tolerance, the pixel can be identified as a foreground pixel. The process can be repeated for each pixel to identify the foreground pixels.

[0083] In block 512, the scale space engine 120 identifies background pixels in the image using the distance transform for the object mask. The value of the distance transform for a pixel of the object mask can be compared to the second value that is a boundary tolerance. If the value of the distance transform for the pixel is greater than the second value that is a boundary tolerance, the pixel can be identified as a background pixel. The process can be repeated for each pixel to identify the background pixels.

[0084] In block 514, the scale space engine 120 generates a label that is based on the identification of the foreground pixels and the background pixels.

The foreground pixels identified in block 510 can form a subset of the foreground pixels shown in Figure 7A, for example. Similarly, background pixels identified in block 512 can form a subset of the background pixels shown in Figure 7A, for example. Undetermined pixels – pixels that are not identified as foreground pixels or background pixels – form an unknown region. The scale space engine 120 can identify and store the unknown region as a label for the object 702. Figure 7C depicts an example of a label (depicted in black color) for object 702.

[0085] The first value and the second value, which are boundary tolerances, can be selected such that the label is sufficiently large to include the actual boundary of the object 702. In some embodiments, the first value and the second value are uniform across the boundary of the object. In other embodiments, one or both of the first value and the second value are not uniform across the boundary of the object. For example, varying one or both of the first value and the second value can result in a label with a varying width.

[0086] In block 516, the scale space engine 120 outputs the label. A label can define an unknown region around the boundary of the object that is of interest. A label can be used to determine a potential map for underdetermined pixels as described, for example, above with reference to Figure 3. In some embodiments, an object mask can be produced based on the potential map and the object mask can be more accurate than an original mask. In some embodiments, the label is provided with the image as band covering the boundary of the object of interest. Figure 7D depicts the object 702 with a boundary 704 of a new object mask that is outputted as a comparison with the boundary 706 of the original mask mask. The boundary 704 of the new object mask more closely identifies an actual boundary of the object 702 as compared to the boundary 706 of the original mask.

[0087] In some embodiments, a more precise label can be generated from an initial label that is computed from an object mask. Figure 6 depicts one embodiment of generating a label.

[0088] In block 602, the scale space engine 120 receives an object mask for an image. The object mask may be a representation of an image that

represents pixels of an object (foreground pixels) with a first color and pixels outside the object (background pixels) with a second color.

[0089] In block 604, the scale space engine 120 inverts the object mask. For example, the foreground pixels can be inverted to background pixels and designated with a second color. The background pixels can be inverted to foreground pixels with a first color.

[0090] In block 606, the scale space engine 120 shrinks the inverted object mask. Shrinking the inverted object mask can include morphologically thinning the inverted object mask to determine hard background constraints, which may include a partial skeletonization of the background. The hard background constraints may prevent fine details in the mask from disappearing during further processing. In some embodiments, the hard background constraints are used as background pixels.

[0091] In block 608, the scale space engine 120 shrinks the object mask. In some embodiments, the object mask is shrunk by morphological thinning the object mask to determine hard foreground constraints, which may include a partial skeletonization of the object mask. The hard foreground constraints may prevent fine details in the mask from disappearing during further processing. The hard foreground constraints can be used as foreground pixels.

[0092] In some embodiments, the object mask is padded by twice the tolerance before thinning, thinned by twice the tolerance, and then unpadded by twice the tolerance to avoid edge effects and to support constraints being computed for the unknown region. Constraints within the tolerance of the object mask boundary may be kept.

[0093] In block 610, the scale space engine 120 generates an initial label based on the shrunk inverted mask, and the shrunk mask. The shrunk inverted mask may be represented by hard background constraints. The shrunk mask may be represented by hard foreground constraints. The initial label may represent an estimation of the unknown region of the image from the outline of the object mask and the hard constraints. In some embodiments, certain foreground pixels are determined from a combination of the hard constraints obtained from morphological thinning and a portion of the

boundary of the mask. This portion may be located at least $1/8$ of the tolerance away from the foreground constraints obtained through the morphological thinning. The background pixels may be identified as those located greater than the tolerance away from the boundary of the mask, in addition to the hard background constraints determined through morphological thinning.

[0094] In block 612, the scale space engine 120 determines an initial potential map using the initial label. For example, the initial potential map may be determined in the same or similar manner as described above with reference to blocks 312 and 314 of Figure 3 by using the initial label.

[0095] In block 614, the scale space engine 120 generates a final label using the initial potential map, the shrunk inverted mask, and the shrunk mask. The shrunk inverted mask and the shrunk mask may be represented by the hard foreground constraints and the hard background constraints.

[0096] In block 616, the scale space engine 120 outputs the final label. The final label may be outputted overlaying the image or otherwise. In some embodiments, scale space engine 120 receives the final label by outputting it and can use the final label for further processing as discussed with reference to Figure 3.

Exemplary Methods of Processing an Image using a Potential Map

[0097] Potential maps according to various embodiments of the present invention can be used to improve processing of the images to produce desired quality and processing efficiency. In some embodiments, potential maps can be used to process images in methods that require user interfaces for a skilled user. For example, a method may be an interactive method that uses potential maps with inputs received from a user to process images.

[0098] Figure 8 depicts one embodiment a method for processing images using a potential map and user inputs. The method of Figure 8 is described with reference to the system depicted in Figure 1. Other implementations, however, are possible.

[0099] In block 802, the scale space engine 120 receives an image that has at least one object. The object may be a representation of an actual object. The image can include background objects that are different from the

object. For example, the object may be an “object-of-interest” and the background objects may be considered the same as other background pixels not associated with an object.

[00100] In block 804, the scale space engine 120 generates scale space images from the image. Scale space images can be generated, for example, using the methods described with reference to Figure 2 or Figure 3.

[00101] In block 806, the scale space engine 120 receives key points. Key points may be points on a boundary of an object that are received from a user through the UI device 112. For example, the image with the label overlayed on it can be displayed to the user on the display 114. A user can use a mouse or other device to identify two or more key points located on a boundary of the object. The key points can be spaced apart by a certain amount. The key points may be used to refine the label, for example, or otherwise. The scale space engine 120 may also received an indication from a user of the region associated with inside the boundary and the region associated with outside the boundary.

[00102] In block 808, the scale space engine 120 computes a label based on the key points. A label can be computed by estimating a line segment or a spline segment connecting the key points. In one embodiment, a line segment or a spline segment is estimated by interpolation between the key points. A boundary tolerance value can be assigned to the segment to produce a label that is extended along the segment and that has a width equivalent to the boundary tolerance value. The region covered by the label can be determined by the value for the boundary tolerance selected to ensure true boundary pixels are included in the resulting label. If the selected tolerance value exceeds the distance between the key points, the tolerance value can be reduced proportionally. The scale space engine 120 can also determine the side of the label that is associated with foreground pixels (inside the object) and the side of the label that is associated with the background pixels (outside the object).

[00103] In block 810, the scale space engine 120 crops an image segment from the image based on the label. For example, the scale space engine 120 can isolate the image segment from the image to analyze further.

[00104] In block 812, the scale space engine 120 determines a potential map. The potential map can be determined from the image segment. For example, the potential map can be determined using a method as described with reference to Figure 2 or Figure 3.

[00105] In block 814, the scale space engine 120 computes boundary points associated with the boundary of the object from the potential map. In some embodiments, the potential map is used to identify pixels between the key points that are likely to be the boundary of the object. The pixels may be identified using the key points and the potential map that identifies the likelihood of the pixels between the key points being associated with the boundary of an object.

[00106] The scale space engine 120 can receive an indication of whether the boundary points are acceptable in block 816. The boundary points may be acceptable if the boundary points appear to the user to be associated with the boundary of the object. The points may be unacceptable if one or more of the points do not appear to the user to be associated with the boundary of the object.

[00107] If the scale space engine 120 receives an indication that the points are not acceptable, the process returns to block 806, in which the scale space engine 120 receives additional key points. The additional key points may be points on a boundary of an object that are identified from a user through the UI device 112. For example, the image with the label overlaid on it can be displayed to the user on the display 114. A user can use a mouse or other UI device 112 to identify additional key points located on a boundary of the object. The additional key points can be received in block 806, and a more accurate label computed based on these additional key points in block 808. This process can be repeated until the boundary points are found accepted in block 816. If the scale space engine 120 receives an indication that the boundary points are acceptable, the scale space engine 120 outputs the boundary points in block 820 for further processing. In some embodiments, the boundary points can be outputted to the display 114 by overlaying the boundary points on the image.

[00108] Further processing can include various processes. For example, the boundary points can be used to generate an object mask for the image segment between the key points. A complete object mask can be generated by repeating the process as described in Figure 8 for all key point pairs.

[00109] Examples of other image processing methods that may use potential maps include (a) semiautomatic improvement of an existing object boundary and (b) improvement of boundaries in interpolated frames.

[00110] In a semiautomatic improvement method, a boundary for an object has been determined and is associated with control points. The control points can be treated as “key points” and the method described with reference to Figure 8 can be used to generate pixel-accurate boundaries between the control points. Crop boxes can be generated for each segment or spline of the boundary, and labeling of each segment or spline can be set to be associated with the control points. Each crop box can be processed independent of the others. In some embodiments, the scale space transform is applied to all of the crop boxes together. The process can result in a replacement object boundary. A user can choose between the original object boundary and the replacement object boundary.

[00111] In an improvement of boundaries in interpolated frames method, interpolated points can be moved. In the interpolated frame, unknown regions for each segment of an object boundary can be extended to overlap by a certain tolerance around an interpolated point. The intersection of the unknown regions can be considered the unknown region for applying the method of Figure 8, for example. The foreground and background pixels can be determined from the intersection of boundaries of segment boundaries. After applying the method of Figure 8, for example, the interpolated point can be moved to a point that is closest to an original position on the estimated boundary, which may be overridden or modified by a user as needed.

General

[00112] While the present subject matter has been described in detail with respect to specific embodiments thereof, it will be appreciated that those skilled in the art, upon attaining an understanding of the foregoing may readily produce alterations to, variations of, and equivalents to such embodiments.

Accordingly, it should be understood that the present disclosure has been presented for purposes of example rather than limitation, and does not preclude inclusion of such modifications, variations and/or additions to the present subject matter as would be readily apparent to one of ordinary skill in the art.

Claims

1. A method comprising:
 - receiving an image having at least one object;
 - generating, by a computing device, at least two scale space images from the image, the computing device comprising a processor configured to cause the computing device to create the at least two scale space images, the at least two scale space images having different levels of detail and being the same size;
 - using the at least two scale space images to determine, for each pixel of the image, a three dimensional graph with a plurality of links to adjacent pixels to that pixel;
 - determining a plurality of weights for the plurality of links of the three-dimensional graph;
 - determining potential values from the plurality of weights, each potential value representing a likelihood of an associated pixel being within a boundary of the object or being outside the boundary of the object;
 - determining a potential map from the potential values; and using the potential map to process the image.
2. The method of claim 1, further comprising converting the image to a color model, wherein the color model comprises the image in a CIE L*a*b* color space.
3. The method of claim 1, wherein generating the at least two scale space images from the image comprises converting the image to scale space using at least two low-pass filters.
4. The method of claim 3, wherein the at least two low-pass filters comprise Gaussian kernels, wherein the different levels of detail comprises different degrees of blur.
5. The method of claim 1, wherein generating the at least two scale space

images from the image comprises converting the image to scale space using at least two wavelet filters.

6. The method of claim 1, wherein generating the at least two scale space images from the image comprises converting the image to scale space using an edge persevering decomposition process.

7. The method of claim 1, wherein determining the plurality of weights for the plurality of links of the three-dimensional graph comprises:

- determining the plurality of links associated with a pixel;
- determining a weight for each link of the plurality of links associated with the pixel; and
- collecting the weight for each link of the plurality of links to form the plurality of weights.

8. The method of claim 1, wherein determining the potential map from the potential values comprises determining a geometric mean for the potential values, the potential map comprising the geometric mean for the potential values.

9. The method of claim 1, further comprising receiving a label for the at least one object, the label including boundary pixels of the at least one object, wherein determining potential values from the plurality of weights comprises determining the potential values using the label.

10. The method of claim 9, wherein receiving the label comprises generating the label comprising:

- receiving an object mask for the image;
- computing an inverted object mask from the object mask for the image;
- determining a first distance transform from the inverted object mask;
- determining a second distance transform from the object mask;
- identifying foreground pixels in the image using the first distance transform;
- identifying background pixels in the image using the second distance transform; and

generating the label based on the identified foreground pixels and the identified background pixels.

11. The method of claim 9, wherein receiving the label comprises generating the label comprising:

receiving an object mask for the image;

computing an inverted object mask from the object mask for the image;

shrinking the inverted object mask;

shrinking the object mask for the image;

generating an initial label based on the shrunk inverted object mask and based on the shrunk object mask for the image;

determining an initial potential map for the image using the initial label;

and

generating the label using the initial potential map, the shrunk inverted object mask, and the shrunk object mask.

12. The method of claim 11, wherein shrinking the inverted object mask comprises using a morphological thinning process on the inverted object mask,

wherein shrinking the object mask for the image comprises using the morphological thinning process on the object mask for the image.

13. The method of claim 1, wherein using the potential map to process the image comprises using the potential map to generate an image mask.

14. The method of claim 13, wherein using the potential map to generate the image mask comprises:

receiving at least two key points identifying an estimated boundary of the object;

computing a label based on the at least two key points; cropping an image segment based on the label; determining a potential map from the image segment; creating boundary points from the potential map;

responsive to receiving a command identifying the boundary points as being unacceptable, computing a second potential map using new key points;

responsive to receiving a command identifying the boundary points as being acceptable, outputting the boundary points; and
generating the image mask using the boundary points.

15. The method of claim 14, wherein the new key points comprises a greater number than the at least two key points.

16. The method of claim 14, wherein the at least two key points identify the estimated boundary of the object in at least two image frames,
wherein the boundary points identify the portion of the estimated boundary of the object in at least one image frame located between the at least two image frames.

17. A computing device comprising:

a processor; and

a computer-readable medium for storing a scale space engine, the scale space engine being executable by the processor to cause the computing device to:

receive an image having at least one object;

generate at least two scale space images from the image, the at least two scale space images having different levels of detail and being the same size;

use the at least two scale space images to determine, for each pixel of the image, a plurality of weights, each weight of the plurality of weights being associated with a link;

receive a label for the at least one object, the label including boundary pixels of the at least one object;

determine potential values from the plurality of weights and using the label, each potential value representing a likelihood of an associated pixel being within a boundary of the object or being outside the boundary of the object;

determine a potential map from the potential values; and

use the potential map to create an image mask for processing the image.

18. The computing device of claim 17, wherein the scale space engine is configured to cause the computing device to convert the image to a color model, wherein the color model comprises the image in a CIE L*a*b* color space.

19. The computing device of claim 17, wherein the scale space engine is configured to cause the computing device to generate the at least two scale space images from the image by converting the image to scale space using one of (i) at least two wavelet filters, (ii) an edge persevering decomposition process, or (iii) at least two low-pass filters.

20. The computing device of claim 17, wherein the scale space engine is configured to cause the computing device to determine the potential map from the potential values by determining a geometric mean for the potential values, the potential map comprising the geometric mean for the potential values.

21. The computing device of claim 17, wherein the scale space engine is configured to cause the computing device to receive the label for the image by generating the label, wherein the scale space engine is configured to cause the computing device to generate the label by:

- receiving an object mask for the image;
- computing an inverted object mask from the object mask for the image;
- determining a first distance transform from the inverted object mask;
- determining a second distance transform from the object mask;
- identifying foreground pixels in the image using the first distance transform;
- identifying background pixels in the image using the second distance transform; and
- generating the label based on the identified foreground pixels and the identified background pixels.

22. The computing device of claim 17, wherein the scale space engine is configured to cause the computing device to receive the label for the image by generating the label, wherein the scale space engine is configured to

- cause the computing device to generate the label by:
 - receiving an object mask for the image;
 - computing an inverted object mask from the object mask for the image;
 - shrinking the inverted object mask;
 - shrinking the object mask for the image;
 - generating an initial label based on the shrunk inverted object mask and based on the shrunk object mask for the image;
 - determining an initial potential map for the image using the initial label;
 - and
 - generating the label using the initial potential map, the shrunk inverted object mask, and the shrunk object mask.

23. A computer-readable medium having program code stored on the computer-readable medium, the program code comprising:

- code for receiving an image having at least one object;
- code for generating at least two scale space images from the image, the at least two scale space images having different levels of detail and being the same size;

- code for using the at least two scale space images to determine a potential map, the potential map representing a likelihood of whether a pixel is within a boundary of the object or outside the boundary of the object, wherein code for using the at least two scale space images to determine the potential map comprises:

- code for using the at least two scale space images to determine, for each pixel of the image, a plurality of weights, each weight of the plurality of weights being associated with a link, comprising:
 - code for determining a plurality of links associated with a pixel;
 - code for determining a weight for each link of the plurality of links associated with the pixel; and
 - code for collecting the weight for each link of the plurality of links to form the plurality of weights;
 - code for receiving a label for the image;
 - code for determining potential values from the plurality of weights

and using the label, each potential value representing a likelihood of an associated pixel being within the boundary of the object or being outside the boundary of the object; and

code for determining the potential map from the potential values, comprising code for determining a geometric mean for the potential values, the potential map comprising the geometric mean for the potential values; and

code for using the potential map to identify the boundary of the object.

24. The computer-readable medium of claim 23, further comprising code for converting the image to a color model, wherein the color model comprises the image in a CIE L*a*b* color space.

25. The computer-readable medium of claim 23, wherein code for generating the at least two scale space images from the image comprises code for converting the image to scale space using at least two low-pass filters comprising Gaussian kernels, wherein the different levels of detail comprise different degrees of blur.

26. The computer-readable medium of claim 23, wherein code for generating the at least two scale space images from the image comprises code converting the image to scale space using one of (i) at least two wavelet filters or (ii) an edge persevering decomposition process.

27. The computer-readable medium of claim 23, further comprising code for generating a label to be used to determine the potential map, wherein the code for generating the label comprises:

code for receiving an object mask for the image;

code for computing an inverted object mask from the object mask for the image;

code for determining a first distance transform from the inverted object mask;

code for determining a second distance transform from the object mask;

code for identifying foreground pixels in the image using the first

- distance transform;
 - code for identifying background pixels in the image using the second distance transform; and
 - code for generating the label based on the identified foreground pixels and the identified background pixels.

28. The computer-readable medium of claim 23, further comprising code for generating a label to be used to determine the potential map, wherein the code for generating the label comprises:

- code for receiving an object mask for the image;
- code for computing an inverted object mask from the object mask for the image;
- code for shrinking the inverted object mask using a morphological thinning process;
- code for shrinking the object mask for the image using the morphological thinning process;
- code for generating an initial label based on the shrunk inverted object mask and based on the shrunk object mask for the image;
- code for determining an initial potential map for the image using the initial label; and
- code for generating the label using the initial potential map, the shrunk inverted object mask, and the shrunk object mask.

29. The computer-readable medium of claim 23, wherein code for using the potential map to identify the boundary of the object comprises code for using the potential map to generate an image mask comprising:

- program code for receiving at least two key points identifying an estimated boundary of the object;
- program code for computing a label based on the at least two key points;
- program code for cropping an image segment based on the label;
- program code for determining a potential map from the image segment;
- program code for creating boundary points from the potential map;

program code for, responsive to receiving a command identifying the boundary points as being unacceptable, computing a second potential map using new key points, wherein the new key points comprises a greater number than the at least two key points;

program code for responsive to receiving a command identifying the boundary points as being acceptable, outputting the boundary points; and

program code for generating the image mask from the boundary points.

30. The computer-readable medium of claim 29, wherein the at least two key points identify the estimated boundary of the object in at least two image frames,

wherein the boundary points identify the portion of the estimated boundary of the object in at least one image frame located between the at least two image frames.

31. A method comprising:

receiving an image having at least one object;

generating, by a computing device, at least two scale space images from the image, the computing device comprising a processor configured to cause the computing device to create the at least two scale space images, the at least two scale space images having different levels of detail;

using the at least two scale space images and potential values to determine a potential map by determining a geometric mean for the potential values, the potential map comprising the geometric mean for the potential values and representing a likelihood of whether pixels of the image are within a boundary of the object or outside the boundary of the object, wherein using the at least two scale space images and the potential values to determine the potential map comprises:

using the at least two scale space images to determine, for each pixel of the image, a plurality of weights, each weight of the plurality of weights being associated with a link, comprising:

determining a plurality of links associated with a pixel;

determining a weight for each link of the plurality of links

associated with the pixel; and

collecting the weight for each link of the plurality of links to form the plurality of weights;

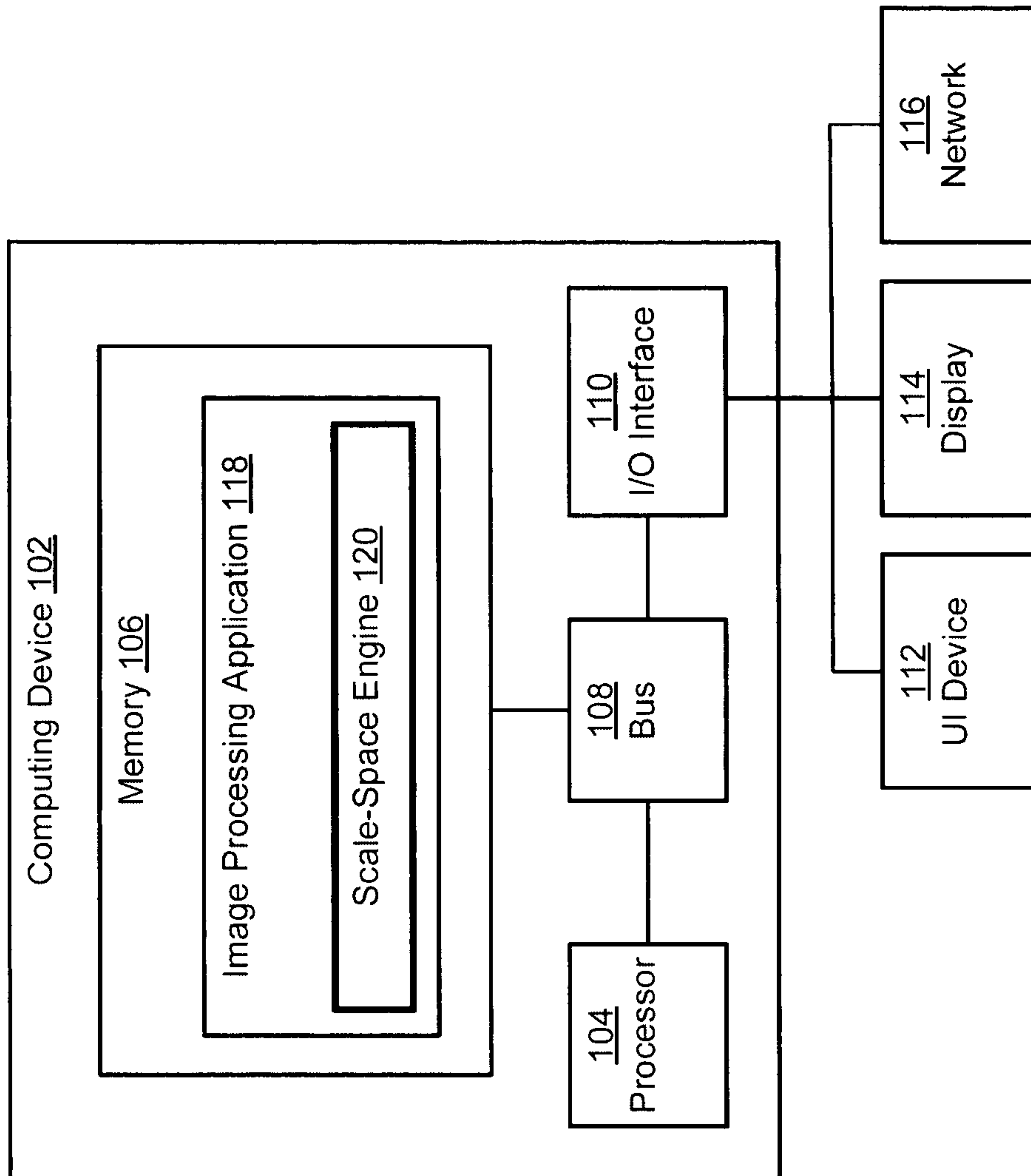
receiving a label for the image:

determining potential values from the plurality of weights and using the label, each potential value representing the likelihood of an associated pixel being within the boundary of the object or being outside the boundary of the object; and

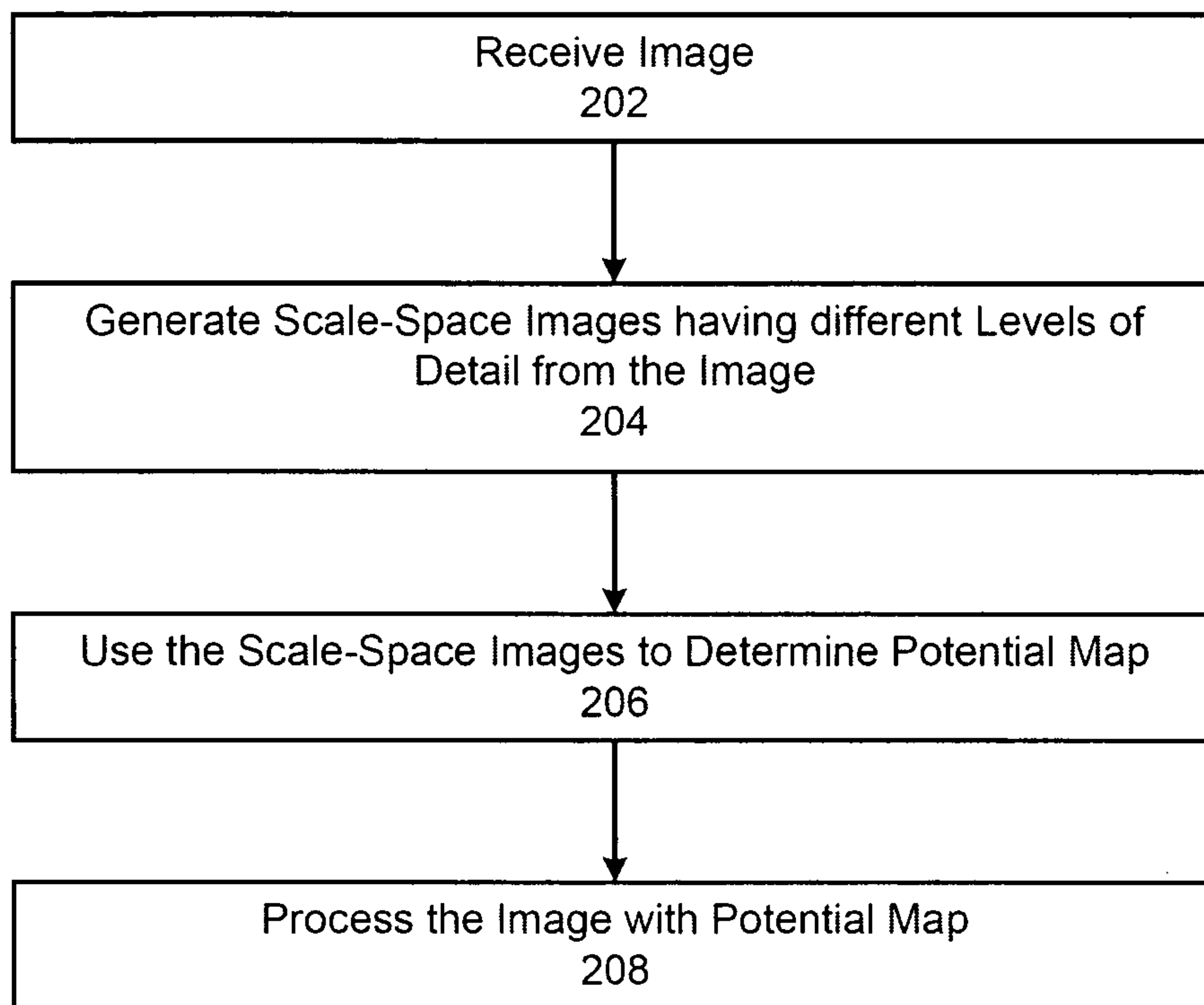
determining the potential map from the potential values by determining the geometric mean for the potential values; and

using the potential map to identify the boundary of the object.

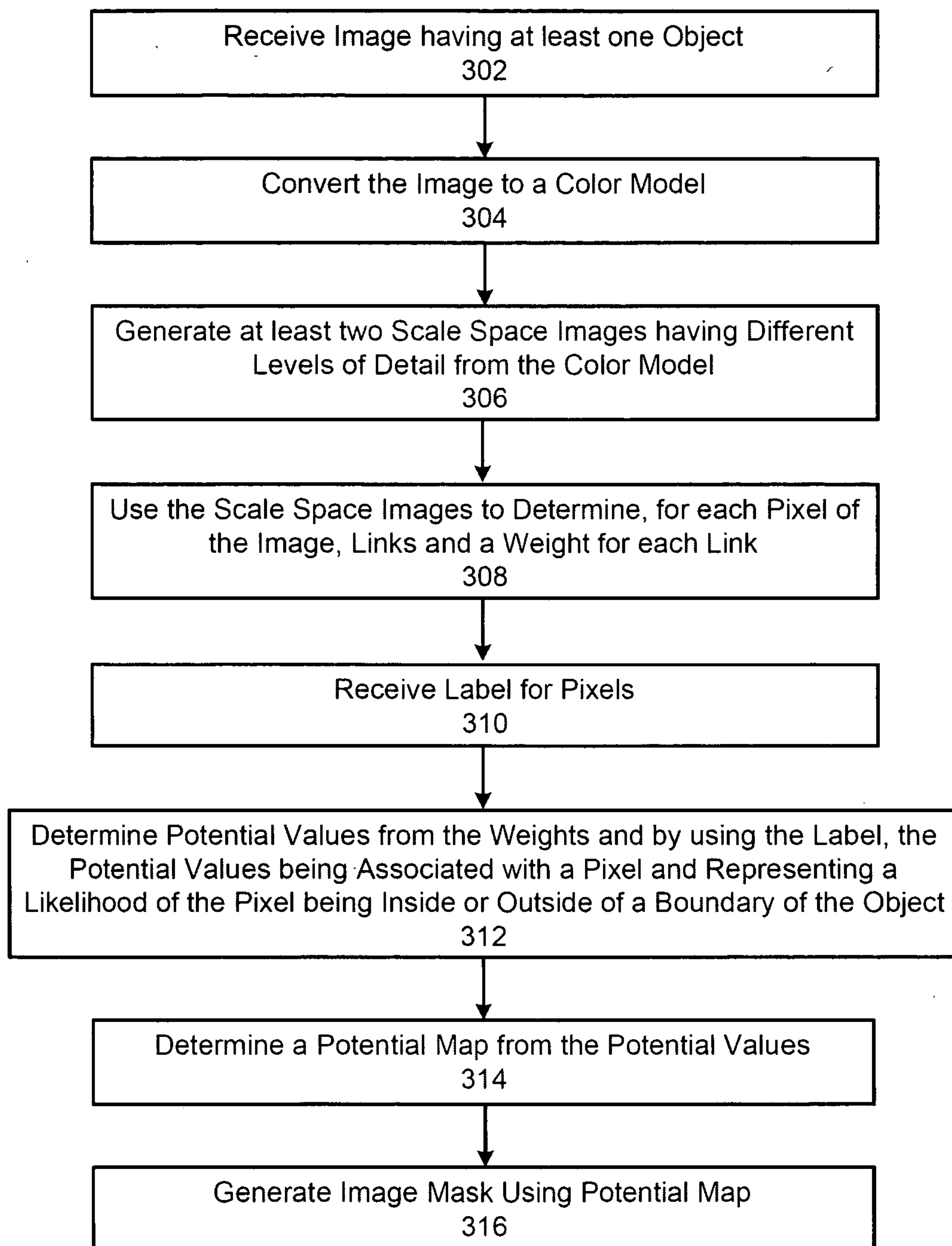
1/8

**Figure 1**

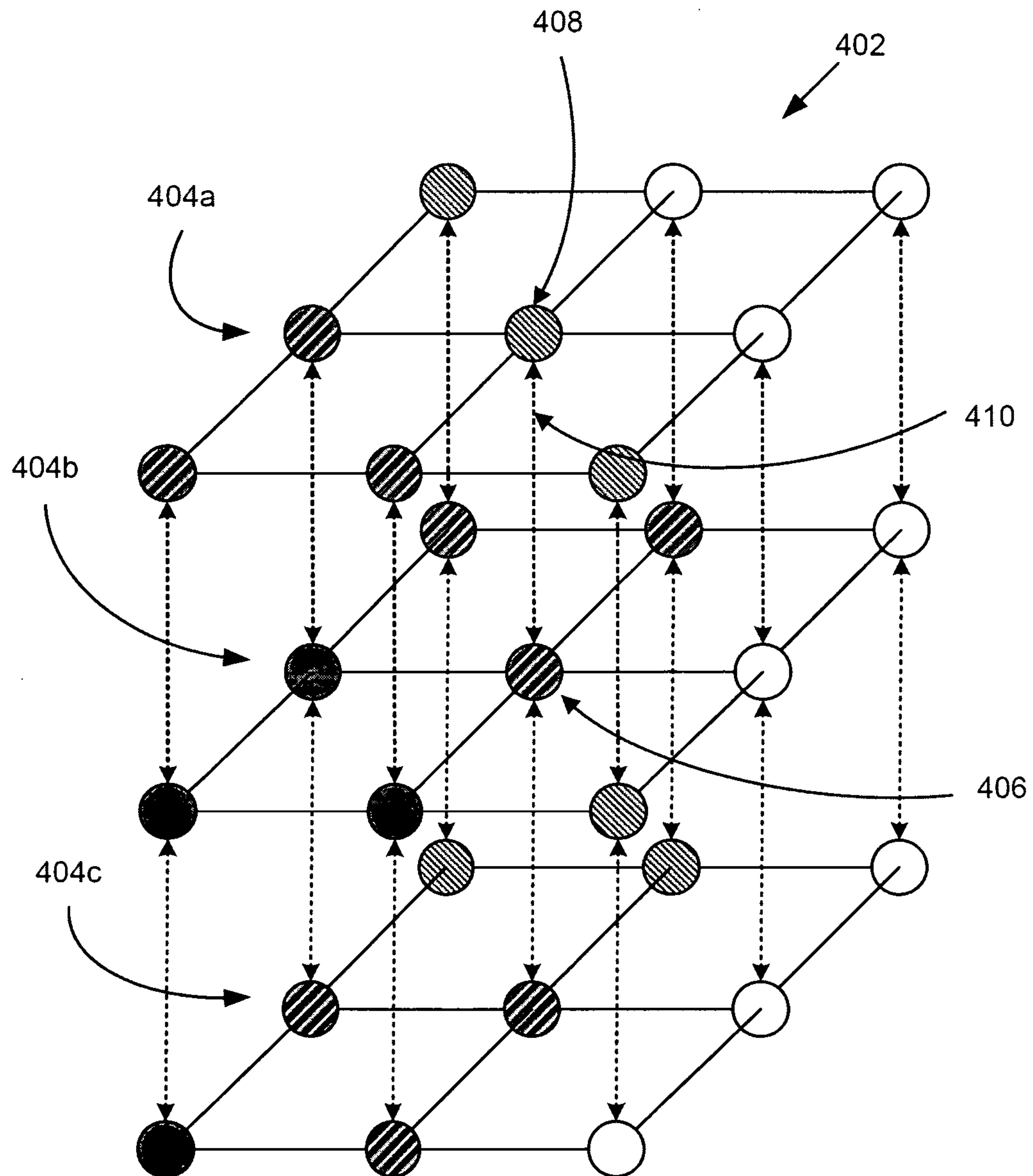
2/8

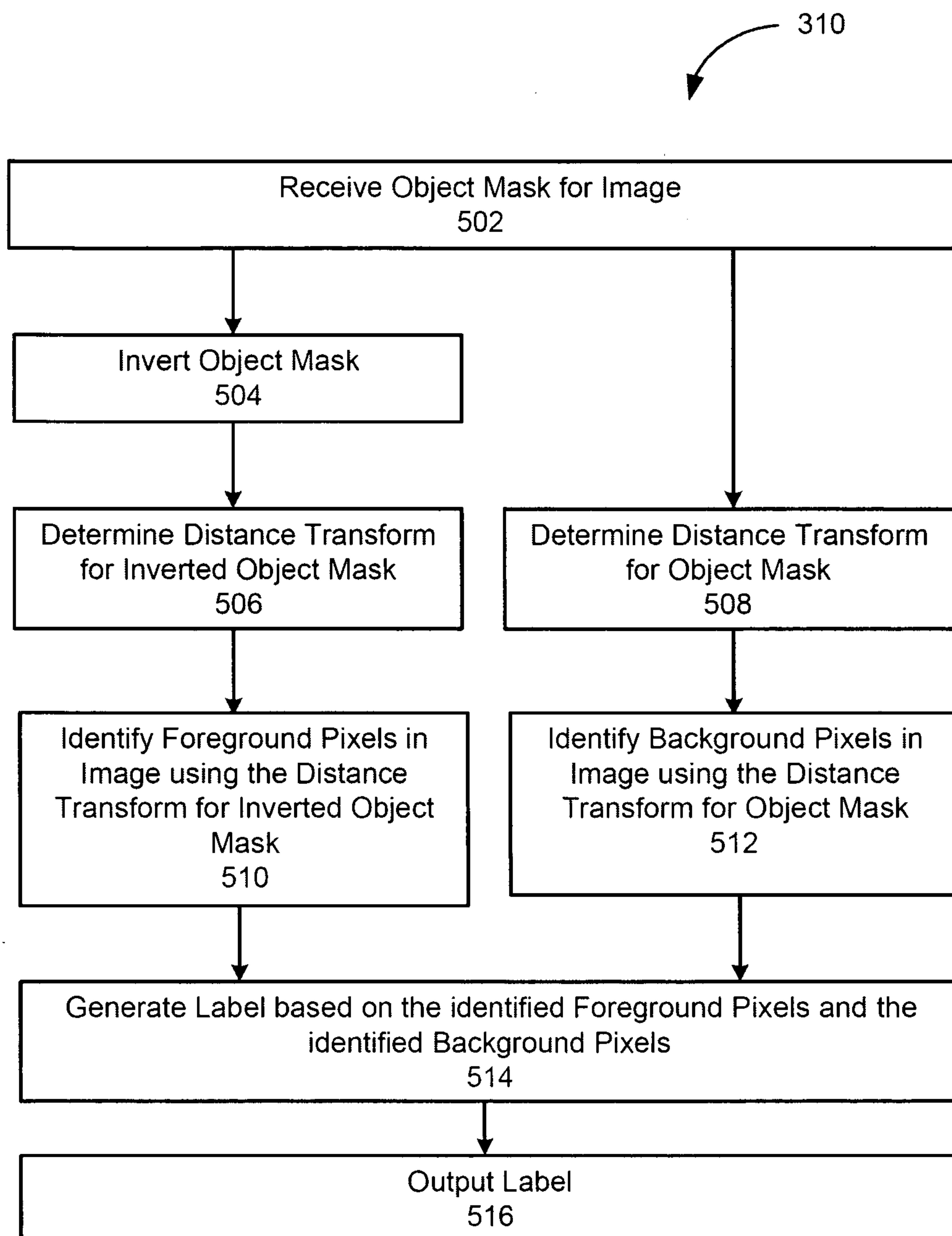
**Figure 2**

3/8

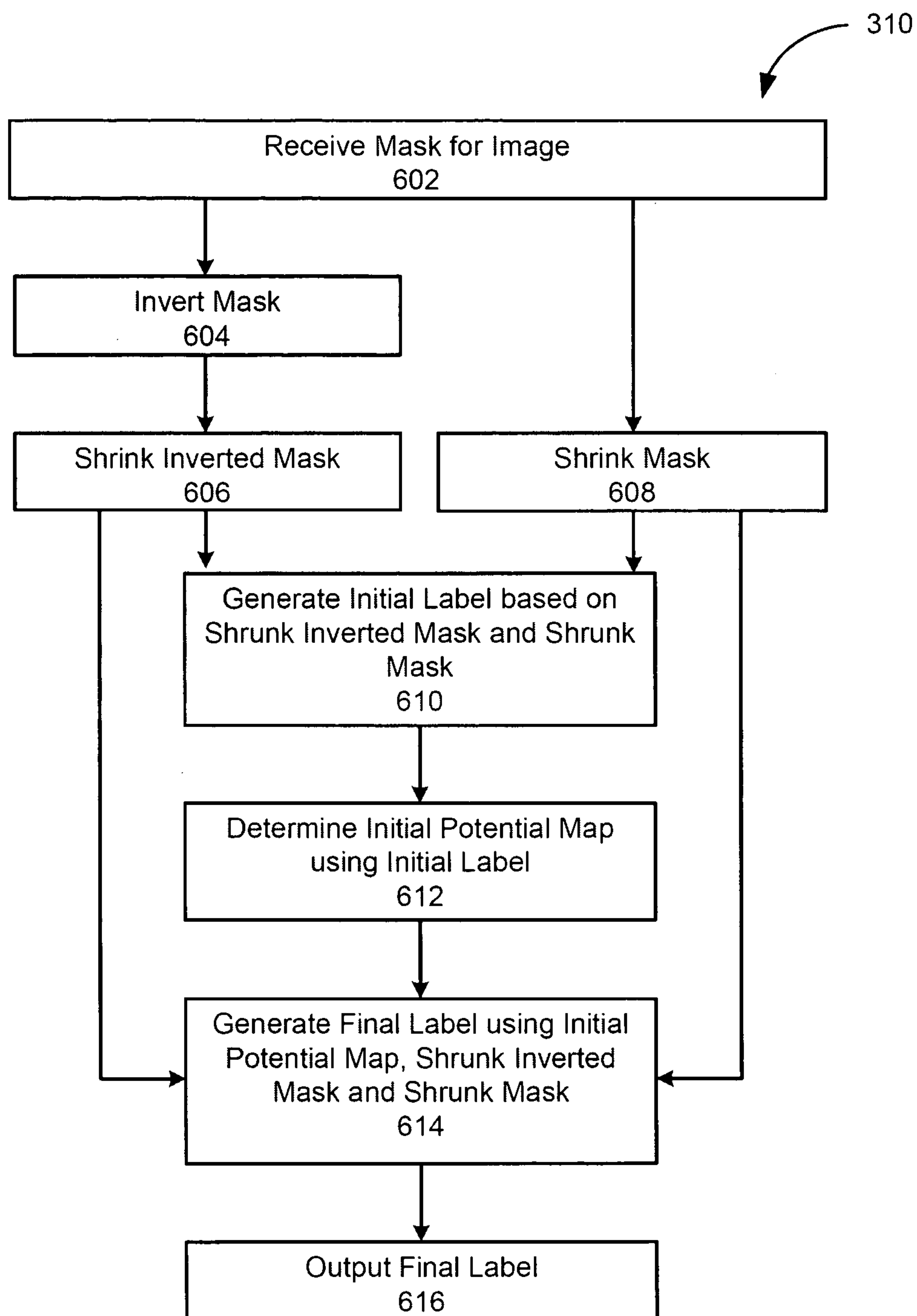
**Figure 3**

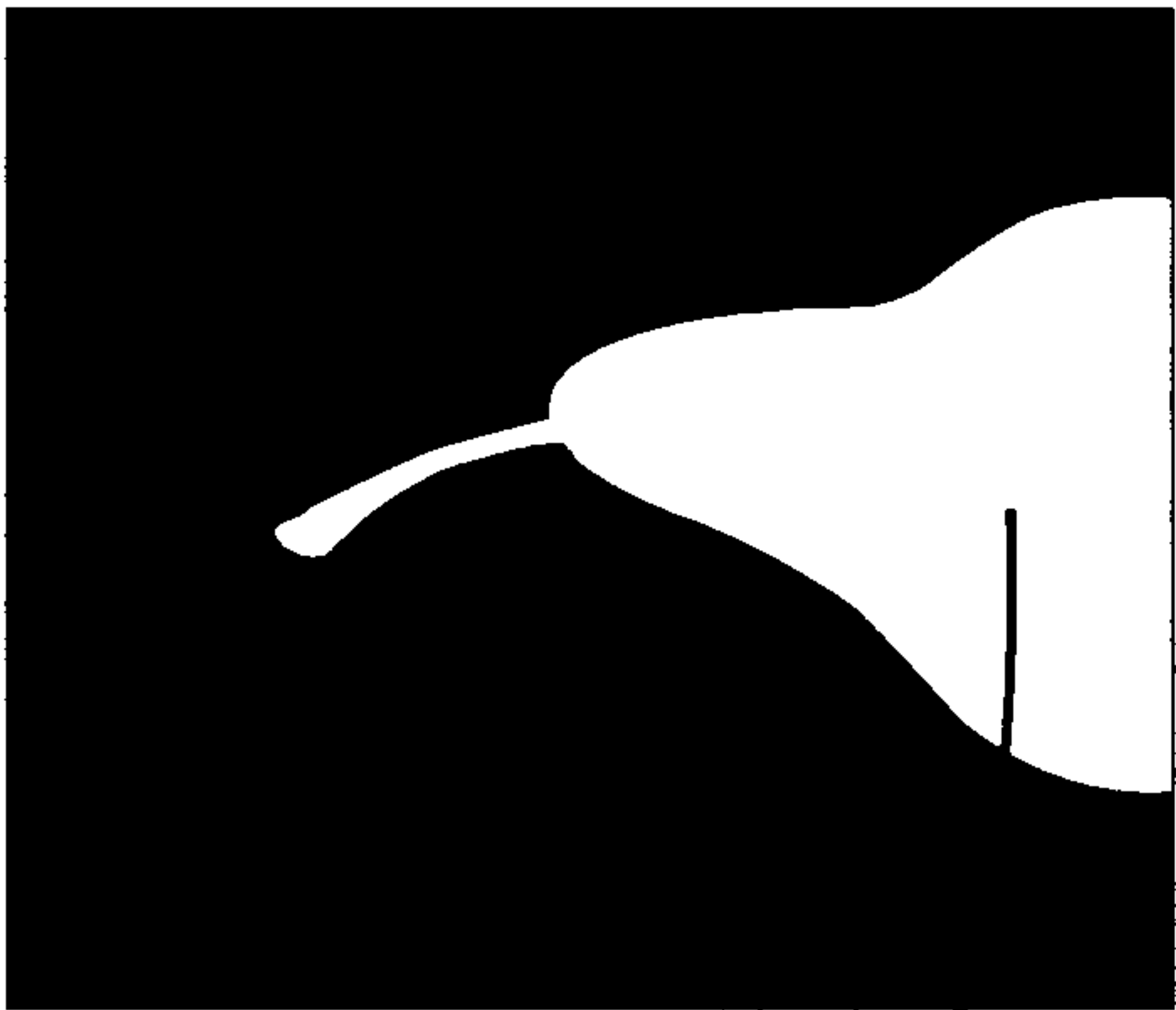
4/8

**Figure 4**

**Figure 5**

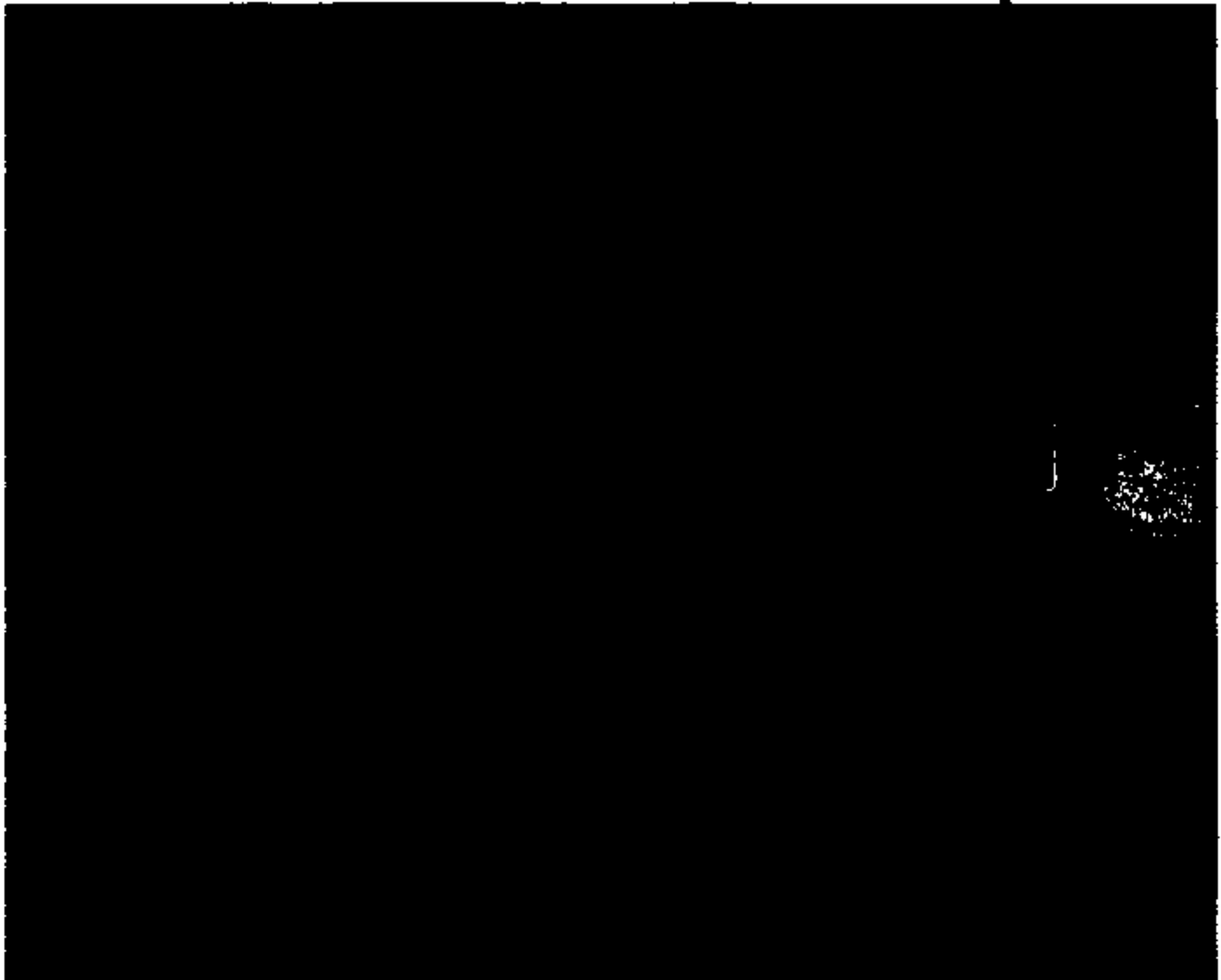
6/8

**Figure 6**



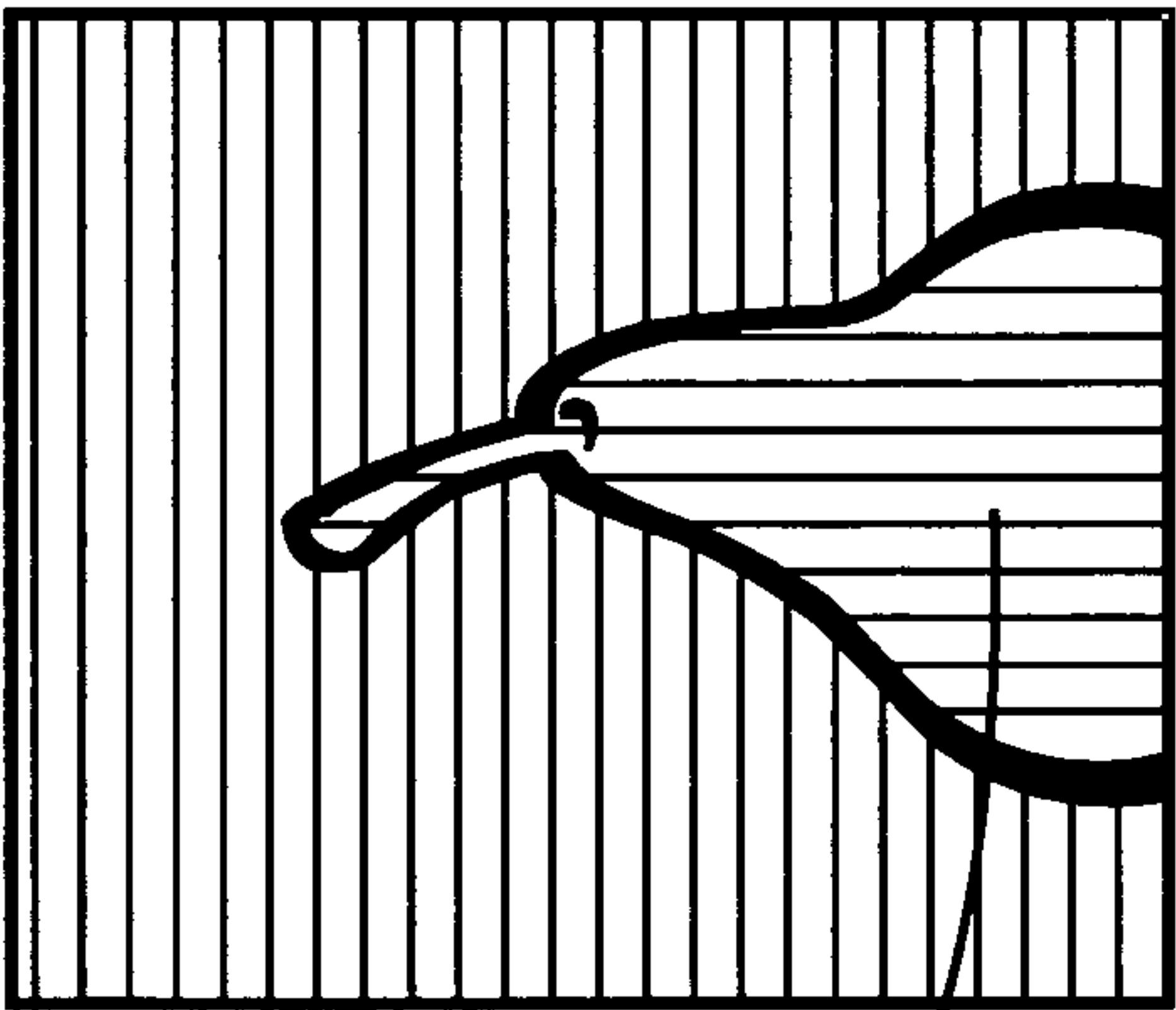
702

Figure 7A



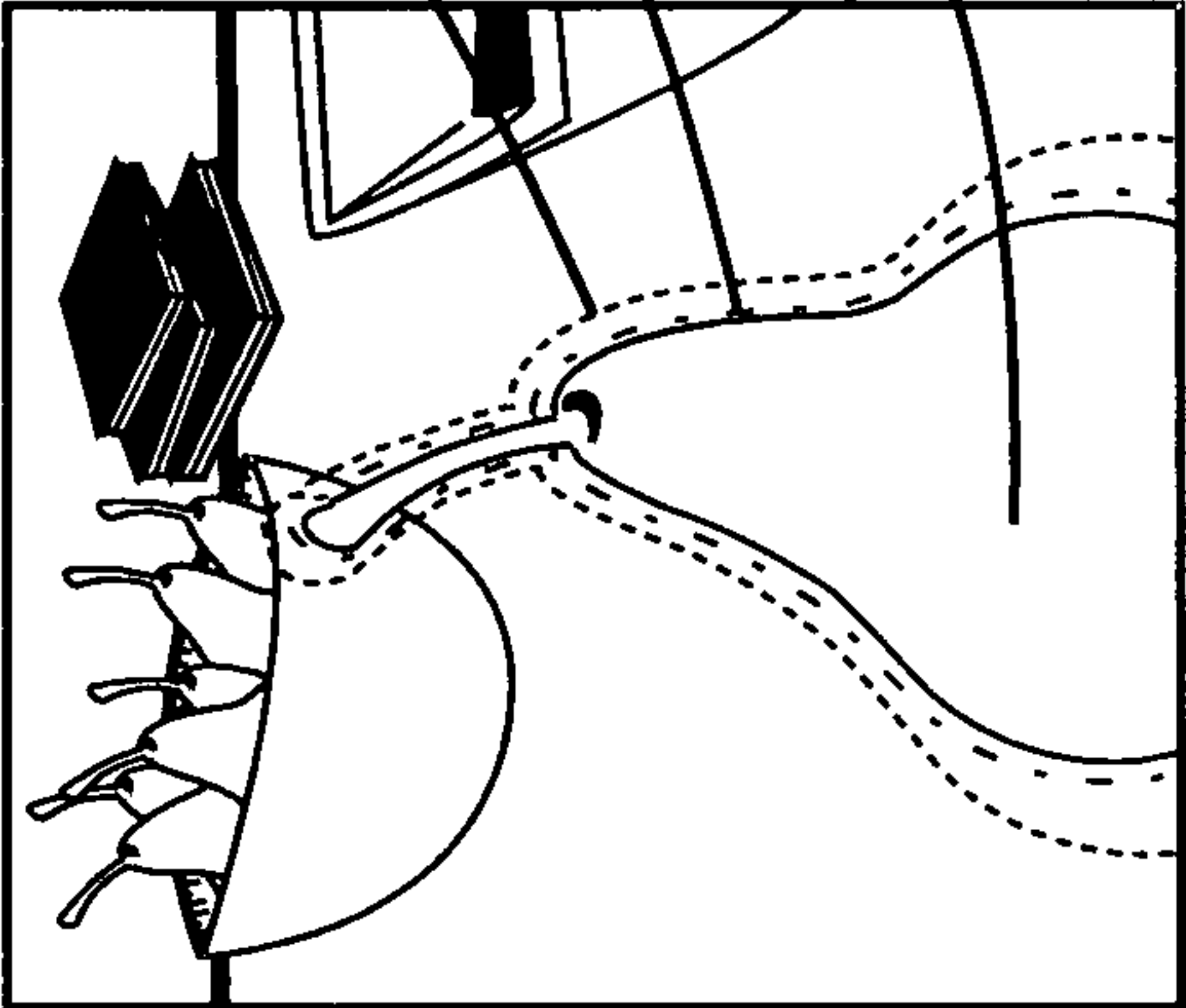
702

Figure 7B



702

Figure 7C



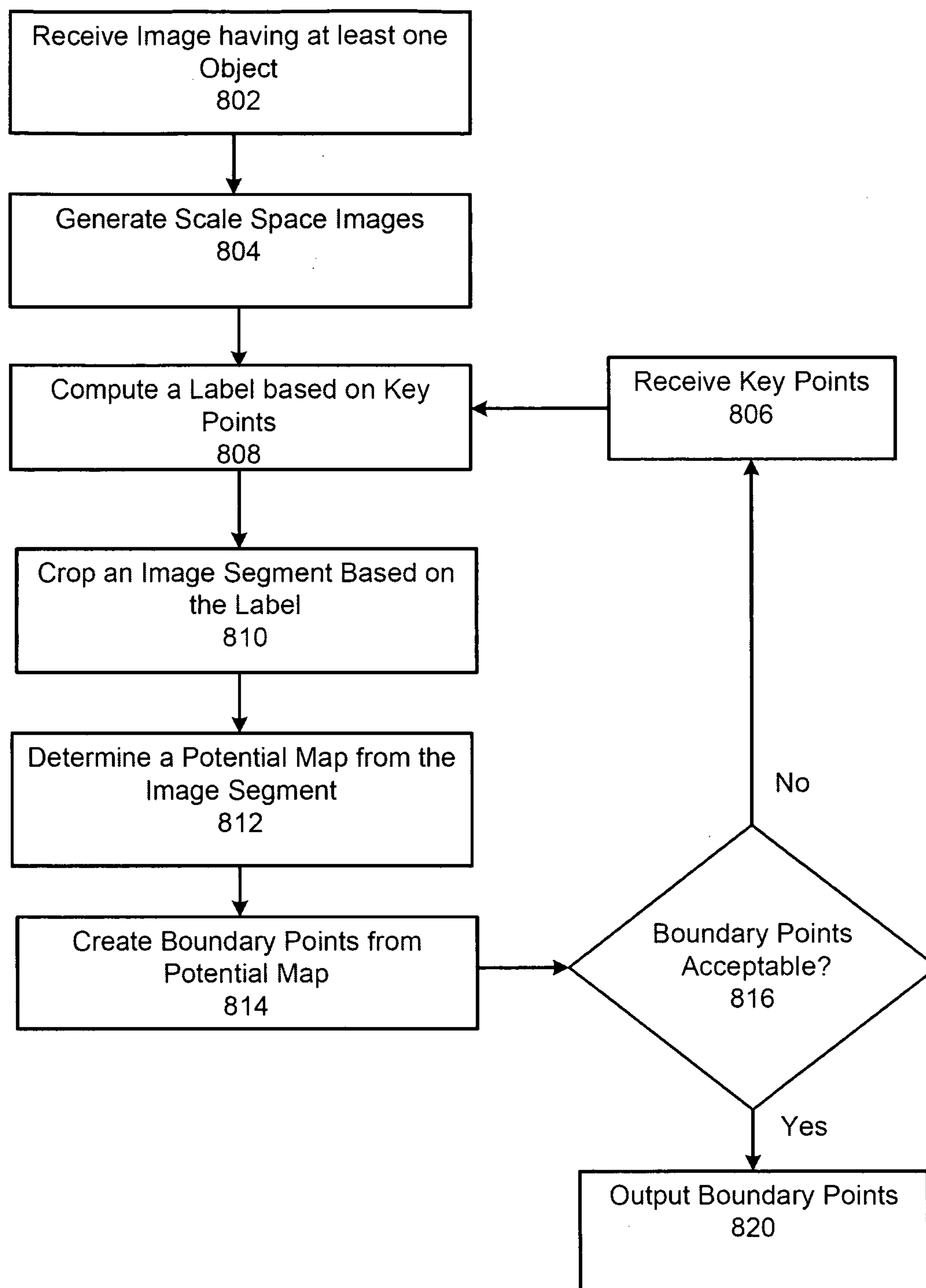
706

704

702

Figure 7D

8/8

**Figure 8**

