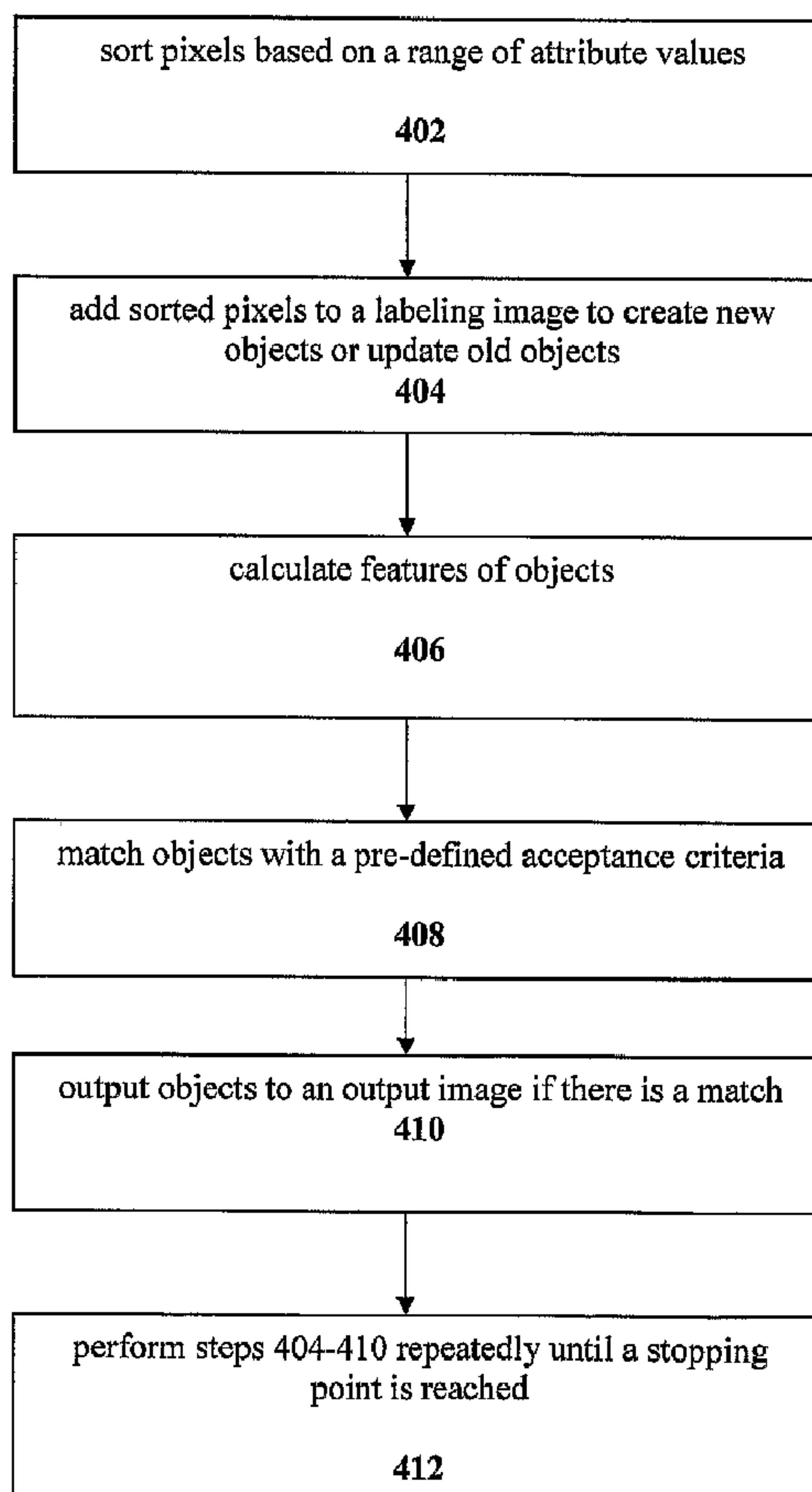




(86) Date de dépôt PCT/PCT Filing Date: 2006/01/10  
 (87) Date publication PCT/PCT Publication Date: 2006/07/20  
 (45) Date de délivrance/Issue Date: 2014/07/22  
 (85) Entrée phase nationale/National Entry: 2007/06/19  
 (86) N° demande PCT/PCT Application No.: US 2006/000726  
 (87) N° publication PCT/PCT Publication No.: 2006/076312  
 (30) Priorités/Priorities: 2005/01/10 (US60/642,110);  
 2006/01/09 (US11/328,354)

(51) Cl.Int./Int.Cl. *G06T 5/00* (2006.01)  
 (72) Inventeur/Inventor:  
 ZAHNISER, MICHAEL, US  
 (73) Propriétaire/Owner:  
 CYTYC CORPORATION, US  
 (74) Agent: RICHES, MCKENZIE & HERBERT LLP

(54) Titre : PROCÉDE DESTINÉ A UNE SEGMENTATION D'IMAGES AMÉLIORÉE  
 (54) Title: METHOD FOR IMPROVED IMAGE SEGMENTATION



(57) Abrégé/Abstract:

An improved image segmentation algorithm is provided that identifies each object in an image. Pixels of the image are sorted based on a range of attribute values of the image. These pixels are then added to a labeling image one by one starting with an extreme

(57) **Abrégé(suite)/Abstract(continued):**

point in the range of attribute values. Features are calculated for each object located and these features are matched with pre-defined acceptance criteria. If there is a match, the object is output to an output image. The steps of adding pixels to the image, evaluating the features of the resulting objects and outputting the objects are repeated until a stopping point is reached.

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

(19) World Intellectual Property Organization  
International Bureau



(43) International Publication Date  
20 July 2006 (20.07.2006)

PCT

(10) International Publication Number  
**WO 2006/076312 A3**

(51) International Patent Classification:  
**G06T 5/00** (2006.01)

[US/US]; 33 Sheridan Road, Wellesley, Massachusetts  
02481 (US).

(21) International Application Number:  
PCT/US2006/000726

(74) Agent: **BURSE, David, T.**; Vista IP Law Group LLP,  
12930 Saratoga Avenue, Suite D-2, Saratoga, CA 95070  
(US).

(22) International Filing Date: 10 January 2006 (10.01.2006)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:  
60/642,110 10 January 2005 (10.01.2005) US  
11/328,354 9 January 2006 (09.01.2006) US

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

(71) Applicant (*for all designated States except US*): **CYTYC CORPORATION** [US/US]; 250 Campus Drive, Marlborough, Massachusetts 01752 (US).

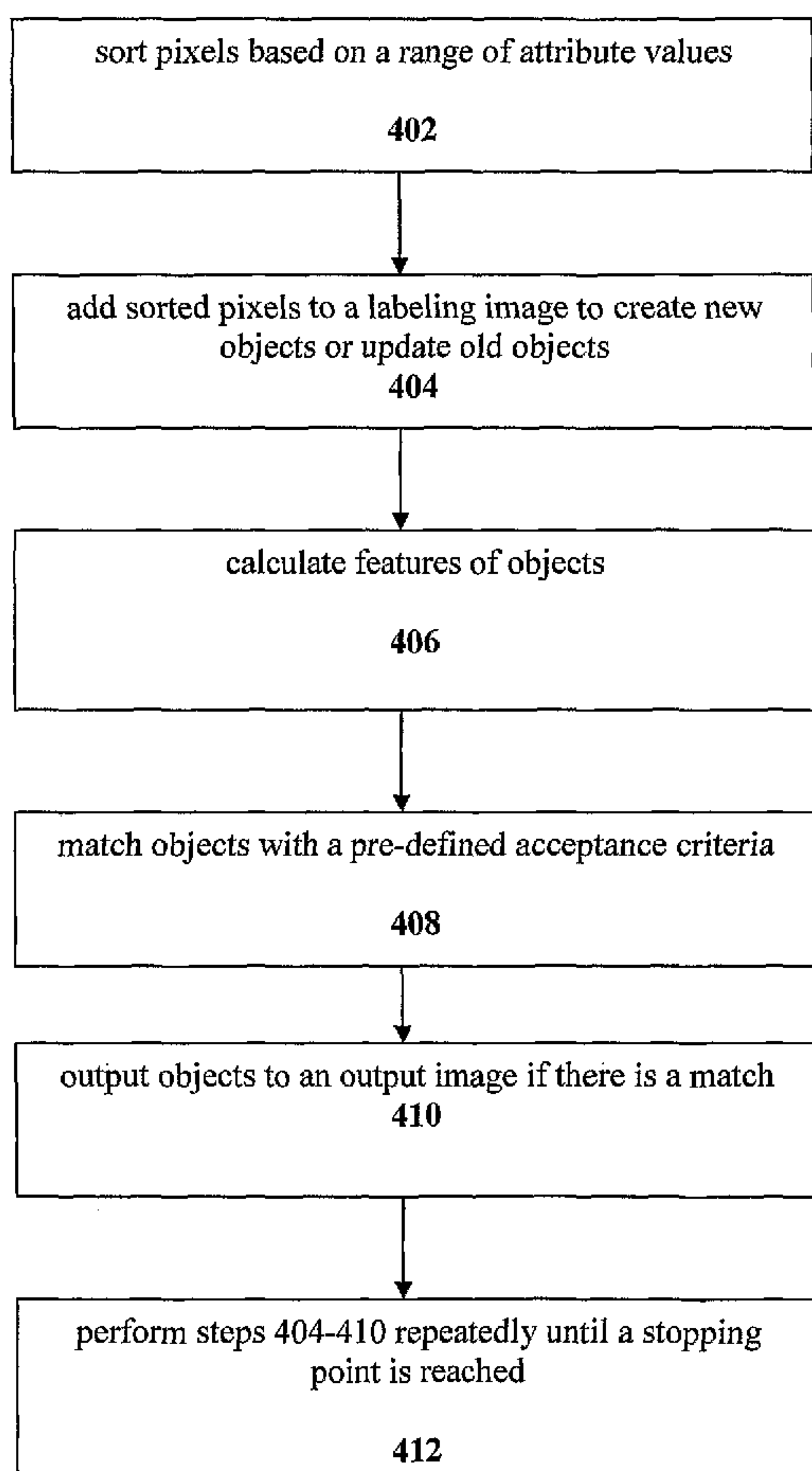
(72) Inventor; and

(75) Inventor/Applicant (*for US only*): **ZAHNISER, Michael**

(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),

[Continued on next page]

(54) Title: METHOD FOR IMPROVED IMAGE SEGMENTATION



(57) Abstract: An improved image segmentation algorithm is provided that identifies each object in an image. Pixels of the image are sorted based on a range of attribute values of the image. These pixels are then added to a labeling image one by one starting with an extreme point in the range of attribute values. Features are calculated for each object located and these features are matched with pre-defined acceptance criteria. If there is a match, the object is output to an output image. The steps of adding pixels to the image, evaluating the features of the resulting objects and outputting the objects are repeated until a stopping point is reached.

WO 2006/076312 A3

**WO 2006/076312 A3**



European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

**Published:**

- *with international search report*
- *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

**(88) Date of publication of the international search report:**  
21 September 2006

*For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.*

## METHOD FOR IMPROVED IMAGE SEGMENTATION

### BACKGROUND OF THE INVENTION

#### Field of Invention

The present invention relates generally to the field of image analysis. More specifically, the present invention is related to a novel image segmentation method.

#### Discussion of Prior Art

In pathology or cytology, it is often desirable to locate and measure cells or nuclei using an automated or semi-automated instrument. Such instruments may be used for research, or for screening. An example of the latter is the screening for cervical cancer using the Papanicolou test (or Pap test). These instruments acquire and analyze digital images to locate cells of interest or to classify slides as being normal or suspect.

In the analysis of objects in digital images it is essential that the objects be distinguished from the background of the image. To characterize cells or objects, the objects must first be located. The process of locating objects within the digital image is known as "segmentation." A variety of techniques are used in the segmentation process to locate the objects of interest so that subsequent computer analysis can characterize the objects. For example, segmentation of an image containing cells might allow the cell's nucleus and/or cytoplasm to be located.

A traditional approach to the task of locating and classifying objects within an image involves several stages: first - segmenting the image to create a binary mask of the objects; then - labeling the objects in this mask, with each connected set of pixels assigned a different label; and finally - measuring various features of the labeled objects.

One of the techniques used for segmenting images is "thresholding." In this technique, a threshold value of image brightness is chosen and each pixel in the image is then compared with this threshold value. Pixels with a brightness value above this threshold

are considered background pixels; pixels with values below the threshold are considered object pixels. The threshold value for locating objects may be chosen based on an image histogram, which is a frequency distribution of the darkness values found within an image. A thresholding algorithm may find a single threshold value using these histograms. For instance, the threshold value might be half-way between the darkest and lightest pixels. Alternatively, the threshold value might be chosen as an inflection point between the abundant "background" pixels and the more rare "object" pixels. Finding an ideal threshold for each object in an image is a difficult task. Often a single threshold value is not optimal for multiple objects with varying darkness values within an entire image.

Once the threshold value is chosen and the thresholding process is completed, the "object" pixels can form a binary mask of the objects in the image. A boundary around the mask might be used to represent each object. The boundary might or might not reflect the object accurately. Many methods have been developed to refine the boundary once it is located. Such methods may use darkness information near the boundary, or constraints such as gradient, curvature, "closeness to a circle," etc. to refine boundaries.

Currently known techniques for image segmentation are often complex and time consuming. These techniques do not always yield high accuracy in the segmentation process, particularly if there is little contrast between the object to be located and the background surrounding it. Consequently, current segmentation algorithms often fail to locate objects properly. In cell image analysis, for example, a cell nucleus might be incorrectly segmented because the located boundary is too large or too small. This can result in false positive events (the instrument incorrectly calls a normal object suspicious) or false negative events (the instrument misses a true suspicious object).

There is a need for improved segmentation for automated imaging and automated imaging devices, in particular for the accurate identification of object boundaries.

Whatever the precise merits, features, and advantages of currently known segmentation techniques, none of them achieve or fulfill the purposes of the present invention.

## SUMMARY OF THE INVENTION

The present invention provides for a method to identify each object in an image, the method comprising steps of: (a) sorting pixels based on a range of attribute values of the image; (b) adding the sorted pixels, one by one, to a "labeling image" to locate objects in the

labeling image, starting with an extreme point in the range of attribute values; (c) outputting the objects onto an output image if features of the objects match a pre-defined acceptance criteria; and (d) performing steps b and c repeatedly until a stopping point is reached, the stopping point representing another extreme point in the range of attribute values.

The present invention provides for an article of manufacture comprising a computer readable medium having computer readable program code embodied therein which identifies each object in an image, the medium comprising: (a) computer readable program code sorting pixels based on a range of attribute values of the image; (b) computer readable program code adding the sorted pixels, one by one, to a labeling image to locate objects in the labeling image, starting with an extreme point in the range of attribute values; (c) computer readable program code outputting the objects onto an output image if features of the objects match a pre-defined acceptance criteria; and (d) computer readable program code performing steps b and c repeatedly until a stopping point is reached, the stopping point representing another extreme point in the range of attribute values.

The present invention provides for a method to identify each object in an image under a plurality of threshold values, the method comprising steps of: (a) sorting pixels in the image based on a range of attribute values of the pixels, wherein the range of attribute values correspond to the plurality of threshold values; (b) adding pixels, one by one, to a labeling image to create new objects or update old objects, starting with an extreme point in the range of attribute values; (c) calculating features of the created new objects and the updated old objects; (d) matching the calculated features of the created new objects and the updated old objects with a pre-defined criteria; (e) outputting the created new objects and the updated old objects on an output image if an acceptance criteria is satisfied for the features; and (f) performing step b-e repeatedly until a stopping point is reached, wherein the stopping point chosen from any of the following: another extreme point in the range of values, a point representing background pixels values in the range of attribute values or a point representing pixel values not related to the new objects and the updated old objects.

3a

In another aspect, the present invention resides in a computer-implemented method to identify each object in an image, said method comprising steps of: a. sorting pixels based on a range of pixel values of a single attribute of said image; b. adding one of said sorted pixels to a labeling image to locate an object in said labeling image; c. outputting said object onto an output image if features of said object match a pre-defined acceptance criteria; and d. starting at one of a lowest or highest end in said range of pixel values, performing steps b and c repeatedly utilizing a computer until the other of said lowest or highest end in said range of pixel values is reached.

In a further aspect, the present invention resides in a computer readable medium having computer readable program code embodied therein for execution by a processor to perform a method comprising: a. sorting pixels based on a range of pixel values of a single attribute of said image; b. adding one of said sorted pixels to a labeling image to locate an object in said labeling image; c. outputting said object onto an output image if features of said object match a pre-defined acceptance criteria; and d. starting at one of a lowest or highest end in said range of attribute values, performing steps b and c repeatedly until the other of said lowest or highest end in said range of attribute values is reached.

In a further aspect, the present invention resides in a method to identify each object in an image under a plurality of threshold values, said method comprising steps of: a. sorting pixels in said image based on a range of pixel values of a single attribute of said image, wherein said range of pixel values correspond to said plurality of threshold values; b. adding one of said pixels to a labeling image to create a new object or update an old object; c. calculating features of said created new object or said updated old object; d. matching said calculated features of said created new object or said updated old object with a pre-defined criteria; e. outputting said created new object or said updated old object on an output image if an acceptance criteria is satisfied for said features; and f. starting at one of a lowest or highest end in said range of pixel values, performing steps b through e repeatedly utilizing a computer until a stopping point is reached, said stopping point being selected from the group consisting of: the other of said lowest or highest end in said range of pixel values, a point

3b

representing background pixels values in said range of pixel values, and a point in said range of pixel values representing pixel values not related to said new object or said updated old object.

In a further aspect, the present invention resides in an automated imaging device for identifying an object in an image, wherein the device includes a processor configured to process pixel data representing the image by performing the steps of: a. sorting pixels based on a range of pixel values of a single attribute of the image; b. adding one of the sorted pixels to a labeling image to locate an object in the labeling image; c. outputting the object onto an output image if features of the object match a pre-defined acceptance criteria; and d. starting at one of a highest or lowest end in the range of pixel values, performing steps b and c repeatedly utilizing a computer until the other of said lowest or highest end in the range of pixel values is reached.

In a further aspect, the present invention resides in an automated imaging device for identifying an object in an image, comprising: a processor; and a computer readable medium associated with the processor, the computer readable medium having computer readable program code embodied therein that, when executed by the processor, causes the processor to process pixel data representing the image, including performing the steps of: a. sorting pixels based on a range of pixel values of a single attribute of the image; b. adding one of the sorted pixels to a labeling image to locate an object in the labeling image; c. outputting the object onto an output image if features of the object match a pre-defined acceptance criteria; and d. starting at one of a lowest or highest end in the range of pixel values, performing steps b and c repeatedly until the other of said lowest or highest end in the range of pixel values is reached.

In still a further aspect, the present invention resides in an automated imaging device for identifying an object in an image, wherein the device includes a processor configured to process pixel data representing the image by performing the steps of: a. sorting pixels in the image based on a range of pixel values of a single attribute of the image, wherein the range

3c

of pixel values correspond to the plurality of threshold values; b. adding one of the pixels to a labeling image to create a new object or update an old object; c. calculating features of the created new object or the updated old object; d. matching the calculated features of the created new object or the updated old object with a pre-defined criteria; e. outputting the created new object or the updated old object on an output image if an acceptance criteria is satisfied for the features; and f. starting at one of a highest or lowest end in the range of pixel values, performing steps b through e repeatedly utilizing a computer until a stopping point is reached, the stopping point being selected from the group consisting of: the other of said highest or lowest end in the range of pixel values, a point representing background pixels values in the range of pixel values, and a point in the range of pixel values representing pixel values not related to the new object or the updated old object.

In a further aspect, the present invention resides in a method to identify objects in an image, said method comprising steps of: a. sorting pixels based on a range of pixel values of a single attribute of said image; b. generating a labeling image comprising starting with a blank labeling image and placing said sorted pixels, one by one, on the blank labeling image at a location in the labeling image corresponding to the location of said pixels in said image, each pixel being assigned a value corresponding to a new object, an existing object, or a merged object, wherein placed pixels that do not touch an existing object correspond to new objects, placed pixels that touch an existing object are assigned a value corresponding to that particular object, and placed pixels that join two existing objects are assigned a value corresponding to the merged objects; c. updating features of the objects generated in the labeling image as each sorted pixel is placed in the labeling image; d. generating an output image comprising boundary information on the objects if said features of said objects match a predefined acceptance criteria; and e. starting at one of a lowest or highest end in said range of pixel values, performing steps b, c, and d repeatedly until a stopping point is reached, said stopping point representing the other of said lowest or highest end in said range of pixel values.

3d

In a further aspect, the present invention resides in an article of manufacture comprising a computer readable medium having computer readable program code embodied therein which identifies objects in an image, said medium comprising: a. computer readable program code sorting pixels based on a range of pixel values of a single attribute of said image; b. computer readable program code configured to place said sorted pixels, one by one, on an initially blank labeling image, each pixel being assigned a value corresponding to a new object, an existing object, or a merged object, wherein placed pixels that do not touch an existing object correspond to new objects, placed pixels that touch an existing object are assigned a value corresponding to that particular object, and placed pixels that join two existing objects are assigned a value corresponding to the merged objects; c. computer readable program code configured to update features of the objects generated in the labeling image as each sorted pixel is placed on the labeling image; d. computer readable program code configured to generate an output image comprising boundary information on the objects if features of said objects match a predefined acceptance criteria; and e. computer readable program code performing steps (b) through (d) repeatedly, starting at one of a lowest or highest end in said range of pixel values, until a stopping point is reached, said stopping point representing the other of said lowest or highest end in said range of pixel values.

#### BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 illustrates a brightness "contour map" superimposed on cells in a cell group.

Figure 2 illustrates steps of the image segmentation algorithm, as per one embodiment of the present invention.

Figures 3a-3n collectively illustrate the processing during addition of pixels into a labeling image, as per one embodiment of the present invention.

5 Figure 4 illustrates steps of the image segmentation algorithm, as per a preferred embodiment of the present invention.

Figure 5 depicts the growth of objects in an image at five different threshold values, as per one embodiment of the present invention.

10 Figure 6 illustrates the output image showing objects that were located at different threshold values where only the best state for an object is retained, as per one embodiment of the present invention.

#### DESCRIPTION OF THE PREFERRED EMBODIMENTS

15 While this invention is illustrated and described in a preferred embodiment, the invention may be applied and produced in many different configurations. There is depicted in the drawings, and will herein be described in detail, a preferred embodiment of the invention, with the understanding that the present disclosure is to be considered as an exemplification of the principles of the invention and the associated functional specifications for its construction and is not intended to limit the invention to the  
20 embodiment illustrated. Those skilled in the art will envision many other possible variations within the scope of the present invention.

An improved automated image segmentation technique is described herein for identification of object boundaries in a digital two dimensional image. While the image segmentation technique described herein identifies nuclei, the technique itself can be  
25 applied to identifying any object in a digital image, such as cytoplasm or tissue structures in biological applications, or objects for non-biological applications such as different components on a circuit board, or man-made and natural features on a satellite image. This technique could be extended to a three dimensional image, such as those produced by X-Ray, CAT (Computed Axial Tomography) scan or MRI (Magnetic Resonance Imaging)  
30 devices. Three dimensional pixel elements in such three dimensional images are known as "voxels." Clusters of voxels might represent an organ, or a tumor in three dimensions.

A digital two dimensional image may be thought of as a three-dimensional surface, where the height dimension represents the grey value (i.e. brightness) of each pixel. In a contour map of that surface, identifying the nuclei might be done by finding the contours that are within a certain size range and round enough to be nuclei. If one contour is contained within another, the "better" of the two should be chosen. Figure 1 shows such a contour map superimposed on cells in a cell group. The objects of interest on the image are sets of connected pixels

Generating these contours, which is equivalent to generating all objects under all possible threshold values, is a mathematically complex and time consuming task. A naive approach would be simply to segment the image under each threshold, label the objects, measure their features, and keep the objects with the best features, eliminating objects which are subsets or supersets of "better" objects. This approach would be far too computationally expensive.

Figure 2 illustrates an image segmentation algorithm as per one embodiment of the present invention that identifies each object found in an image by performing the following steps. First the algorithm sorts pixels based on a range of attribute values of the image (step 202). It then adds the sorted pixels one by one to a "labeling image" for the purpose of locating objects in the labeling image, starting with an extreme point in the range of attribute values (step 204). A labeling image is an image wherein the pixels of each distinct object are assigned a unique value, or "label." Also, the extreme point used for the starting value may be at the lowest or the highest end of the range of attribute values. The pixels can be sorted based on attributes such as, but not limited to, brightness, hue, gradient, etc. Features of the located objects are then compared to a pre-defined criteria. If an object matches the criteria, the algorithm outputs the object to an output image, as shown in step 206. A stopping point is also defined for the image. This stopping point represents a point in the range of the attribute values at which the algorithm should stop performing steps 204 and 206 repeatedly (step 208). This stopping point may be the other extreme point in the range of attribute values. In one embodiment, this stopping point is a point representing background pixel values in the range of attribute values. In another embodiment, the stopping point is a point representing pixel values not related to the objects being located.

Figure 4 illustrates an algorithm, as per a preferred embodiment of the present invention. In step 402, pixels in an image are sorted based on a range of attribute values of

the image. The range of attribute values corresponds to a plurality of threshold values in the image. The threshold values are determined from a histogram of the attribute values being used. The pixels can be sorted based on attributes such as, but not limited to, brightness, hue, gradient, etc. Also, the indexing on a histogram could be done by brightness, hue, gradient, etc. Instead of segmenting the image at each threshold value, sorted pixels are added to a blank labeling image one by one, starting with an extreme point in the range of attribute values as shown in step 404, wherein new objects are created or old objects are updated in the process. If the pixel added is not adjacent to another pixel or object already placed on the labeling image (old object), a new object is created (see figures 3a, 3b and 3e). If the added pixel is adjacent to an old object, the pixel is combined with the old object to update the old object (see figures 3c, 3d, 3f, 3g, 3h, 3i, 3j, 3k, 3m and 3n). Further, if the added pixel joins two old objects, the two old objects are updated and merged into one object (see figure 3l). In step 406, features of these new or updated objects are calculated. These features are then matched with pre-defined acceptance criteria, in step 408. If there is a match (i.e. acceptance criteria are satisfied), the new or updated object (representing a located nucleus), is output onto the output image in step 410. The algorithm performs the steps 404-410 repeatedly until a stopping point is reached (step 412). Note that the objects may be output to the output image at any time during the processing of a threshold value (i.e., attribute value). Again, this stopping point may be the other extreme point in the range of attribute values. In one embodiment, this stopping point is a point representing background pixel values in the range of attribute values. In another embodiment, the stopping point is a point representing pixel values not related to the objects being located.

The algorithm tracks the objects created or updated in the steps described earlier, along with their features. The objects are also assigned labels (for example, see figures 3a and 3b). Whenever a pixel is added that touches an existing object, that object's label is assigned to it and the features of that object are updated to reflect the addition of the new pixel (for example, see figures 3d and 3g). When a pixel is added that joins two objects, the labels and features of the two are merged into one (for example, see figure 3l). Figures 3a-3n show how pixels are added into an originally blank mask/labeling image one at a time and labels are assigned as needed to each pixel.

As a result of the way the algorithm is designed, there is a record of the mask of the image during the processing of the image at any given time under a particular threshold, as

well as the labels and features of all objects found at that particular threshold. But instead of generating those data anew for each threshold value, the algorithm builds on the data from the previous threshold. This greatly increases the algorithm's efficiency.

In one embodiment of the present invention, an object is output to the output image only if its current features calculated at a current threshold are a better match to the acceptance criteria than its features calculated in all previous thresholds. In order to determine a better match, data from the previous best match is used. To compute an optimal set of objects that best matches the acceptance criteria out of all objects at all thresholds, a method called dynamic programming is used. The program passes through the thresholds one at a time, keeping track of the best set of objects that can be formed using only the objects located so far. Each object "remembers" the best state it has attained so far. Figure 5 depicts the growth of objects in an image at five different threshold values. Individual objects that were separate at one threshold may be joined at another threshold. Each object in the figure "points to" its previous best state. Objects with no arrows point to themselves (not shown). This means that the current state of these objects is better than any of their previous states. Thus, the new state of an object only needs to be compared to the state to which its arrow points, since this state represents the best state achieved so far. If two objects merge at a given threshold, the arrow for the new, merged object points to the better of the two objects that form the merged object.

While passing through every threshold, the program maintains an output image that contains the best objects from the set of objects formed so far. Whenever an object grows to a new state better than its previous recorded best state, the updated object is drawn into the output image. Figure 6 shows the output image as it is growing. First it contains the objects from a first threshold. Then it contains the best objects from the first and second threshold, then the best objects from the first three thresholds, and so on. The final output image contains all of the objects that were marked as best objects and not later rejected in favor of a better one (compare to Figure 5).

Thus, as per one embodiment of the algorithm, when an object matches the acceptance criteria, instead of being directly output to an output image, it may be conditionally output based upon a comparison between the object's current state and its previous state or states.

While the present invention's algorithm described herein, identifies cell nuclei based on features such as area and perimeter and accepts objects that fit the circularity acceptance criteria (perimeter squared divided by area), other features and criteria can be used to identify objects. Acceptance criteria can be tuned to the particular imaging application (biological or non-biological). When the objects are identified based on criteria other than circularity, other features could be used instead. Moment of inertia, eccentricity, longer and shorter elliptical axes, best fit to an ellipse, contrast with the surrounding pixels, grey value or optical density mean or standard deviation, and texture features are a few other examples of features that could be easily measured and used to specify acceptance criteria for the best segmentation. Other acceptance criteria may also be used such as size, shape, texture, color, density, contrast etc. Multiple criteria sets may allow different kinds of objects such as nuclei, nuclei of different shapes, cytoplasms, nuclear or cytoplasmic "inclusions" and clustered cells, in a single image to be located at the same time. However, it should be understood that the present invention is not limited to the use of these features or acceptance criteria.

Updating the perimeter measurement of an object when a pixel is added to it can be done using only the four pixels immediately adjacent to that pixel if a reasonable approximation of the perimeter is all that is required. Therefore, each time a pixel is "placed," only that pixel and its four nearest neighbors need be examined to determine how the object features should be updated. The only situation where more than these four neighbors are influenced is when two objects are merged and one of the two must be assigned the label of the other. However, optimizations in the merging process allow the algorithm's running time to approach a running time linear in proportion to the number of pixels. Thus, this measurement of perimeter shows how object features can be calculated from previous objects, i.e., it shows that the algorithm builds on the data from the previous threshold and increases the algorithm's efficiency.

As another example, ellipticity (comparison of the measured object to the ellipse defined by the moment of inertia matrix) may be used as a feature to identify nuclei. Segmentation based on ellipticity may better segmentation results, since it would distinguish elongated nuclei from irregularly shaped artifacts.

It can thus be seen that the algorithm of the present invention provides for various improvements: reduction in processing time for segmentation in images, elimination of

time consuming pre-processing to locate potential objects of interest and then establishing regions of interest to more accurately define an object in a secondary process, handling of images with varying darkness and contrast to minimize false negative cases, handling images with abnormal clusters to minimize false negative cases, identifying multiple objects  
5 in a single image at the same time using multiple acceptance criteria.

It should be noted that the algorithm of the present invention differs from other published iterative image analysis techniques such as region growing algorithms and active contours (also called "snakes"). Region growing algorithms work by first dividing an image into many separate regions (individual pixels, small groups of pixels, contiguous pixels with  
10 the same grey level, etc.). Next, regions "grow" by merging with other regions that touch them and have similar features. This is usually an iterative process that stops once no more merges of similar regions can be made. The goal of the technique is for the final set of regions to correspond to the objects being segmented.

In the segmentation algorithm of the present invention, the way that the objects will  
15 "grow" is predetermined: there is only one path that the growth can take and the algorithm simply "watches" the features of the objects changing as they travel along that path. The features that are measured do not influence which pixel is added next or which regions are merged. After a single pass through the image, the "best objects" observed in that path are reported.

20 In the region growing technique, the features of the regions determine which regions are grown or merged. Instead of following a linear path, the region growing technique is like is a search through a tree with many branches. Object features are used to determine which branch to take. Usually multiple passes are made through the image, and the final state of the regions is reported. Region growing algorithms do not store the best states attained by  
25 the objects; rather, the algorithm may have to "back up" to return to a better state.

"Active contours," also called "snakes," are object outlines represented either by polygons or by parametric equations, overlaid on the image. In an active contour image analysis technique, these lines "evolve" to improve their own shape and their  
30 correspondence with the underlying image. Again, this is an iterative process ending when the outlines cannot be improved by further evolution.

The present invention operates on individual pixels and performs operations using only the immediate neighbors of each pixel. In the active contours technique, the contours are represented, not by pixels, but by mathematical functions. The differences between the

present algorithm and the active contours techniques are the same as the differences noted with the region growing technique.

A key difference that distinguishes the algorithm of the present invention from other segmentation algorithms is that other algorithms involve an iterative process seeking the best path through a tree or graph, using the object features to choose which direction to move next. Sometimes this requires backtracking as the algorithm works to end up with the best possible result. The present algorithm follows a linear, predetermined path, passing by and remembering all the "best" objects on its way.

Additionally, the present invention provides for an article of manufacture comprising computer readable program code contained within implementing one or more modules to segment images and identify objects. Furthermore, the present invention includes a computer program code-based product, which is a storage medium having program code stored therein that can be used to instruct a computer to perform any of the methods associated with the present invention. The computer storage medium includes any of, but is not limited to, the following: CD-ROM, DVD, magnetic tape, optical disc, hard drive, floppy disk, ferroelectric memory, flash memory, ferromagnetic memory, optical storage, charge coupled devices, magnetic or optical cards, smart cards, EEPROM, EPROM, RAM, ROM, DRAM, SRAM, SDRAM, or any other appropriate static or dynamic memory or data storage devices.

Implemented in computer program code based products are software modules for:

- a) sorting pixels based on a range of attribute values of an image;
- b) adding the sorted pixels, one by one, to a labeling image to locate objects in the labeling image, starting with an extreme point in the range of attribute values;
- c) outputting the objects onto an output image if features of the objects match a pre-defined acceptance criteria; and
- d) performing steps (b) and (c) repeatedly until a stopping point is reached, said stopping point representing another extreme point in said range of attribute values.

CONCLUSION

A system and method has been shown in the above embodiments for the effective implementation of a method for improved image segmentation. While various preferred embodiments have been shown and described, it will be understood that the scope of the claims should not be limited by the preferred embodiments set forth in the examples, but should be given the broadest interpretation consistent with the description as a whole. Also, the present invention algorithm should not be limited by the type of image (biological, non-biological, two dimensional or three dimensional, etc.), the number of threshold values in an image, the type of objects to be identified in an image, or features and acceptance criteria used to identify objects.

The invention claimed is:

1. A computer-implemented method to identify each object in an image, said method comprising steps of:
  - a. sorting pixels based on a range of pixel values of a single attribute of said image;
  - b. adding one of said sorted pixels to a labeling image to locate an object in said labeling image;
  - c. outputting said object onto an output image if features of said object match a pre-defined acceptance criteria; and
  - d. starting at one of a lowest or highest end in said range of pixel values, performing steps b and c repeatedly utilizing a computer until the other of said lowest or highest end in said range of pixel values is reached.
2. The computer-implemented method to identify each object in the image, as per claim 1, wherein said pixels are sorted based on any of the following attribute values: brightness, hue, or gradient.
3. The computer-implemented method to identify each object in the image, as per claim 1, wherein said object is located by adding said one of said sorted pixels to the labeling image to create a new object or an updated old object.
4. The computer-implemented method to identify each object in the image, as per claim 3, wherein said new object is created if said added pixel is not adjacent to another pixel or object already placed on said labeling image.
5. The computer-implemented method to identify each object in the image, as per claim 3, wherein said updated old object is created if said added pixel is adjacent to already existing said old object or if said added pixel joins two old objects already existing on said labeling image.

6. The computer-implemented method to identify each object in the image, as per claim 1, wherein said object is assigned a label, wherein labels are updated for each added pixel.

7. The computer-implemented method to identify each object in the image, as per claim 6, wherein said label of said object builds on labels from a previously added pixel.

8. The computer-implemented method to identify each object in the image, as per claim 1, wherein said features of said object are calculated from features of an object located when a pixel was previously added.

9. The computer-implemented method to identify each object in the image, as per claim 1, wherein said object is output to said output image only if said object's features calculated when said sorted pixel is added are a better match to said acceptance criteria than said object's features calculated for all previously added pixels.

10. The computer-implemented method to identify each object in the image, as per claim 1, wherein said method is performed on two-dimensional or three-dimensional images from biological or non-biological applications.

11. The computer-implemented method to identify each object in the image, as per claim 10, wherein said object in said biological applications is any of the following: a cell, a nucleus, cytoplasm, a nuclear or cytoplasmic inclusion, clustered cells, or an object within images produced by an imaging device.

12. The computer-implemented method to identify each object in the image, as per claim 11, wherein the imaging device is one of an x-ray machine, a CAT scan machine or a magnetic resonance imaging device.

13. The computer-implemented method to identify each object in the image, as per claim 10, wherein said object in said non-biological applications is any of the following: a component on a circuit board, or a man-made or natural object in a satellite image.

14. The computer-implemented method to identify each object in the image, as per claim 1, wherein said features are any of the following: perimeter, area, object moment of inertia, eccentricity, elliptical axes, best fit to an ellipse, contrast, grey value, optical density mean, standard deviation, or texture.

15. The computer-implemented method to identify each object in the image, as per claim 1, wherein said acceptance criteria are any of the following: size, shape, texture, color, or density.

16. The computer-implemented method to identify each object in the image, as per claim 1, wherein said method uses multiple acceptance criteria to identify different kinds of objects in said labeling image simultaneously.

17. The computer-implemented method to identify each object in the image, as per claim 1, wherein steps b and c are performed repeatedly until a point representing background pixels values in said range of pixel values, or a point in said range of pixel values representing pixel values not related to said objects being located, is reached.

18. A computer readable medium having computer readable program code embodied therein for execution by a processor to perform a method comprising:

- a. sorting pixels based on a range of pixel values of a single attribute of said image;
- b. adding one of said sorted pixels to a labeling image to locate an object in said labeling image;
- c. outputting said object onto an output image if features of said object match a pre-defined acceptance criteria; and

d. starting at one of a lowest or highest end in said range of attribute values, performing steps b and c repeatedly until the other of said lowest or highest end in said range of attribute values is reached.

19. The computer readable medium having computer readable program code embodied therein for execution by the processor to perform the method, as per claim 18, wherein steps b and c are performed repeatedly until a point representing background pixels values in said range of pixel values, or a point in said range of pixel values representing pixel values not related to said objects being located, is reached.

20. A method to identify each object in an image under a plurality of threshold values, said method comprising steps of:

a. sorting pixels in said image based on a range of pixel values of a single attribute of said image, wherein said range of pixel values correspond to said plurality of threshold values;

b. adding one of said pixels to a labeling image to create a new object or update an old object;

c. calculating features of said created new object or said updated old object;

d. matching said calculated features of said created new object or said updated old object with a pre-defined criteria;

e. outputting said created new object or said updated old object on an output image if an acceptance criteria is satisfied for said features; and

f. starting at one of a lowest or highest end in said range of pixel values, performing steps b through e repeatedly utilizing a computer until a stopping point is reached, said stopping point being selected from the group consisting of: the other of said lowest or highest end in said range of pixel values, a point representing background pixels values in said range of pixel values, and a point in said range of pixel values representing pixel values not related to said new object or said updated old object.

21. The method to identify each object in the image under the plurality of threshold values,

as per claim 20, wherein said pixels are sorted based on any of the following attribute values: brightness, hue, or gradient.

22. The method to identify each object in the image under the plurality of threshold values, as per claim 20, wherein said plurality of threshold values is brightness, hue, or gradient.

23. The method to identify each object in the image under the plurality of threshold values, as per claim 20, wherein said new object is created if said added pixel is not adjacent to another pixel or object already placed on said labeling image.

24. The method to identify each object in the image under the plurality of threshold values, as per claim 20, wherein said updated old object is created if said added pixel is adjacent to already existing said old object or if said added pixel joins two old objects already existing on said labeling image.

25. The method to identify each object in the image under the plurality of threshold values, as per claim 20, wherein said method is performed on two-dimensional or three-dimensional images from biological or non-biological applications.

26. The method to identify each object in the image under the plurality of threshold values, as per claim 25, wherein said object in said biological applications is any of the following: a cell, a nucleus, cytoplasm, a nuclear or cytoplasmic inclusion, clustered cells or an object within images produced by an imaging device.

27. The method to identify each object in the image under the plurality of threshold values, as per claim 26, wherein the imaging device is one of an x-ray machine, a CAT scan machine or a magnetic resonance imaging device.

28. The method to identify each object in the image under the plurality of threshold values, as per claim 25, wherein said object in said non-biological applications is any of the

following: a component on a circuit board, or a man-made or natural object in a satellite image.

29. The method to identify each object in the image under the plurality of threshold values, as per claim 25, wherein said features are any of the following: perimeter, area, object moment of inertia, eccentricity, elliptical axes, best fit to an ellipse, contrast, grey value, optical density mean, standard deviation, or texture.

30. The method to identify each object in the image under the plurality of threshold values, as per claim 20, wherein said acceptance criteria are any of the following: size, shape, texture, color, or density.

31. The method to identify each object in the image under the plurality of threshold values, as per claim 20, wherein said method uses multiple acceptance criteria to identify different kinds of objects in said labeling image simultaneously.

32. The method to identify each object in the image under the plurality of threshold values, as per claim 20, wherein said created new object or said updated old object is assigned a label, wherein labels are updated for each added pixel.

33. The method to identify each object in the image under the plurality of threshold values, as per claim 32, wherein said labels build on labels from a previously added pixel.

34. The method to identify each object in the image under the plurality of threshold values, as per claim 20, wherein said features of said created new object or said updated old object are calculated from features of an object located when a pixel was previously added.

35. The method to identify each object in the image under the plurality of threshold values, as per claim 20, wherein said updated old object is output to said output image only if features of said updated old object calculated when said sorted pixel is added are a better

match to said acceptance criteria than features of said updated old object calculated for all previously added pixels.

36. An automated imaging device for identifying an object in an image, wherein the device includes a processor configured to process pixel data representing the image by performing the steps of:

- a. sorting pixels based on a range of pixel values of a single attribute of the image;
- b. adding one of the sorted pixels to a labeling image to locate an object in the labeling image;
- c. outputting the object onto an output image if features of the object match a pre-defined acceptance criteria; and
- d. starting at one of a highest or lowest end in the range of pixel values, performing steps b and c repeatedly utilizing a computer until the other of said lowest or highest end in the range of pixel values is reached.

37. The automated imaging device of claim 36, wherein the pixels are sorted based on any of the following attribute values: brightness, hue, or gradient.

38. The automated imaging device of claim 36, wherein the object is located in step b by adding a respective one of the sorted pixels to the labeling image to create a new object or an updated old object.

39. The automated imaging device of claim 36, wherein the new object is created if the respective added pixel is not adjacent to another pixel or object already placed on the labeling image.

40. The automated imaging device of claim 36, wherein the updated old object is created if the respective added pixel is adjacent to the already existing old object or if the added pixel joins two old objects already existing on the labeling image.

41. The automated imaging device of claim 36, wherein the object is assigned a label, and wherein labels are updated for each added pixel.

42. The automated imaging device of claim 41, wherein the label of the object builds on respective labels of one or more previously added pixels.

43. The automated imaging device of claim 36, wherein the features of the object are calculated from features of an object located when a pixel was previously added.

44. The automated imaging device of claim 36, wherein the object is output to the output image only if the object's features when calculated when the respective sorted pixel is added are a better match to the acceptance criteria than the object's features when calculated for all previously added pixels.

45. The automated imaging device of claim 36, wherein the processor is configured to process two-dimensional and three-dimensional images.

46. The automated imaging device of claim 36, wherein the object is any of the following: a cell, a nucleus, cytoplasm, a nuclear or cytoplasmic inclusion, clustered cells, or an object within images produced by an imaging device.

47. The automated imaging device of claim 46, wherein the imaging device is one of an x-ray machine, a CAT scan machine or a magnetic resonance imaging device.

48. The automated imaging device of claim 36, wherein the object is any of the following: a component on a circuit board, or a man-made or natural object in a satellite image.

49. The automated imaging device of claim 36, wherein the features are any of the following: perimeter, area, object moment of inertia, eccentricity, elliptical axes, best fit to an ellipse, contrast, grey value, optical density mean, standard deviation, or texture.

50. The automated imaging device of claim 36, wherein the acceptance criteria comprises any of the following: size, shape, texture, color, or density.

51. The automated imaging device of claim 36, wherein the processor employs multiple acceptance criteria to identify different kinds of objects in the labeling image simultaneously.

52. The automated imaging device of claim 36, wherein steps b and c are performed repeatedly by the processor until a point representing background pixels values in the range of pixel values, or a point in the range of pixel values representing pixel values not related to the objects being located, is reached.

53. An automated imaging device for identifying an object in an image, comprising:

a processor; and

a computer readable medium associated with the processor, the computer readable medium having computer readable program code embodied therein that, when executed by the processor, causes the processor to process pixel data representing the image, including performing the steps of:

a. sorting pixels based on a range of pixel values of a single attribute of the image;

b. adding one of the sorted pixels to a labeling image to locate an object in the

labeling image;

c. outputting the object onto an output image if features of the object match a pre-defined acceptance criteria; and

d. starting at one of a lowest or highest end in the range of pixel values, performing steps b and c repeatedly until the other of said lowest or highest end in the range of pixel values is reached.

54. The automated imaging device of claim 53, wherein the computer readable program code embodied in the computer readable medium, when executed by the processor, causes the processor to perform steps b and c repeatedly, until a point representing background

pixels values in the range of pixel values, or a point in the range of pixel values representing pixel values not related to the objects being located, is reached.

55. An automated imaging device for identifying an object in an image, wherein the device includes a processor configured to process pixel data representing the image by performing the steps of:

- a. sorting pixels in the image based on a range of pixel values of a single attribute of the image, wherein the range of pixel values correspond to the plurality of threshold values;
- b. adding one of the pixels to a labeling image to create a new object or update an old object;
- c. calculating features of the created new object or the updated old object;
- d. matching the calculated features of the created new object or the updated old object with a pre-defined criteria;
- e. outputting the created new object or the updated old object on an output image if an acceptance criteria is satisfied for the features; and
- f. starting at one of a highest or lowest end in the range of pixel values, performing steps b through e repeatedly utilizing a computer until a stopping point is reached, the stopping point being selected from the group consisting of: the other of said highest or lowest end in the range of pixel values, a point representing background pixels values in the range of pixel values, and a point in the range of pixel values representing pixel values not related to the new object or the updated old object.

56. The automated imaging device of claim 55, wherein the pixels are sorted based on any of the following attribute values: brightness, hue, or gradient.

57. The automated imaging device of claim 55, wherein the plurality of threshold values is brightness, hue, or gradient.

58. The automated imaging device of claim 55, wherein the new object is created if the added pixel is not adjacent to another pixel or object already placed on the labeling image.

59. The automated imaging device of claim 55, wherein the updated old object is created if the added pixel is adjacent to the already existing old object or if the added pixel joins two old objects already existing on the labeling image.

60. The automated imaging device of claim 55, wherein the processor is configured to process two-dimensional and three-dimensional images.

61. The automated imaging device of claim 55, wherein the object is any of the following: a cell, a nucleus, cytoplasm, a nuclear or cytoplasmic inclusion, clustered cells or an object within images produced by an imaging device.

62. The automated imaging device of claim 61, wherein the imaging device is one of an x-ray machine, a CAT scan machine or a magnetic resonance imaging device.

63. The automated imaging device of claim 55, wherein the object is any of the following: a component on a circuit board, or a man-made or natural object in a satellite image.

64. The automated imaging device of claim 55, wherein the features are any of the following: perimeter, area, object moment of inertia, eccentricity, elliptical axes, best fit to an ellipse, contrast, grey value, optical density mean, standard deviation, or texture.

65. The automated imaging device of claim 55, wherein the acceptance criteria comprises any of the following: size, shape, texture, color, or density.

66. The automated imaging device of claim 55, wherein the processor employs multiple acceptance criteria to identify different kinds of objects in the labeling image simultaneously.

67. The automated imaging device of claim 55, wherein the created new object or the updated old object is assigned a label, and wherein labels are updated for each added pixel.

68. The automated imaging device of claim 67, wherein the labels build on labels from a previously added pixel.

69. The automated imaging device of claim 55, wherein the features of the created new object or the updated old object are calculated from features of an object located when a pixel was previously added.

70. The automated imaging device of claim 55, wherein the updated old object is output to the output image only if features of the updated old object calculated when the sorted pixel is added are a better match to the acceptance criteria than features of the updated old object calculated for all previously added pixels.

71. A method to identify objects in an image, said method comprising steps of:

- a. sorting pixels based on a range of pixel values of a single attribute of said image;
- b. generating a labeling image comprising starting with a blank labeling image and placing said sorted pixels, one by one, on the blank labeling image at a location in the labeling image corresponding to the location of said pixels in said image, each pixel being assigned a value corresponding to a new object, an existing object, or a merged object, wherein placed pixels that do not touch an existing object correspond to new objects, placed pixels that touch an existing object are assigned a value corresponding to that particular object, and placed pixels that join two existing objects are assigned a value corresponding to the merged objects;
- c. updating features of the objects generated in the labeling image as each sorted pixel is placed in the labeling image;
- d. generating an output image comprising boundary information on the objects if said features of said objects match a predefined acceptance criteria; and
- e. starting at one of a lowest or highest end in said range of pixel values, performing steps b, c, and d repeatedly until a stopping point is reached, said stopping point representing the other of said lowest or highest end in said range of pixel values.

72. A method to identify objects in an image, as per claim 71, wherein said pixels are sorted based on any of the following attributes: brightness, hue, or gradient.

73. A method to identify objects in an image, as per claim 71, wherein said features of said objects are calculated based on the placed pixel and the four nearest neighbor pixels.

74. A method to identify objects in an image, as per claim 71, wherein said output image is generated only if said object's features calculated when said sorted pixel is placed are a better match to said acceptance criteria than said object's features calculated for all previously placed pixels.

75. A method to identify objects in an image, as per claim 71, wherein said method is performed on two-dimensional or three-dimensional images from biological or non-biological applications.

76. A method to identify objects in an image, as per claim 75, wherein said objects in said biological applications are any of the following: cells, nuclei, cytoplasms, nuclear or cytoplasmic inclusions, clustered cells, or objects within images produced by an imaging device.

77. A method to identify objects in an image, as per claim 76, wherein the imaging device is one of an x-ray machine, a CAT scan machine or a magnetic resonance imaging device.

78. A method to identify objects in an image, as per claim 75, wherein said objects in said non-biological applications are any of the following: components on a circuit board, or man-made or natural objects in a satellite image.

79. A method to identify objects in an image, as per claim 71, wherein said features are any of the following: perimeter, area, object moment of inertia, eccentricity, elliptical axes, best fit to an ellipse, contrast, grey value, optical density mean, standard deviation, or texture.

80. A method to identify objects in an image, as per claim 71, wherein said acceptance criteria are any of the following: size, shape, texture, color, or density.

81. A method to identify objects in an image, as per claim 71, wherein said method uses multiple acceptance criteria to identify different kinds of objects in said image simultaneously.

82. A method to identify objects in an image, as per claim 71, wherein said stopping point is a point representing background pixels values in said range of pixel values or a point representing pixel values not related to said objects being located.

83. An article of manufacture comprising a computer readable medium having computer readable program code embodied therein which identifies objects in an image, said medium comprising:

a. computer readable program code sorting pixels based on a range of pixel values of a single attribute of said image;

b. computer readable program code configured to place said sorted pixels, one by one, on an initially blank labeling image, each pixel being assigned a value corresponding to a new object, an existing object, or a merged object, wherein placed pixels that do not touch an existing object correspond to new objects, placed pixels that touch an existing object are assigned a value corresponding to that particular object, and placed pixels that join two existing objects are assigned a value corresponding to the merged objects;

c. computer readable program code configured to update features of the objects generated in the labeling image as each sorted pixel is placed on the labeling image;

d. computer readable program code configured to generate an output image comprising boundary information on the objects if features of said objects match a predefined acceptance criteria; and

e. computer readable program code performing steps (b) through (d) repeatedly, starting at one of a lowest or highest end in said range of pixel values, until a stopping point

is reached, said stopping point representing the other of said lowest or highest end in said range of pixel values.

84. An article of manufacture comprising a computer readable medium having computer readable program code embodied therein which identifies objects in an image, as per claim 83, wherein said stopping point is a point representing background pixels values in said range of pixel values or a point representing pixel values not related to said objects being located.

1/7

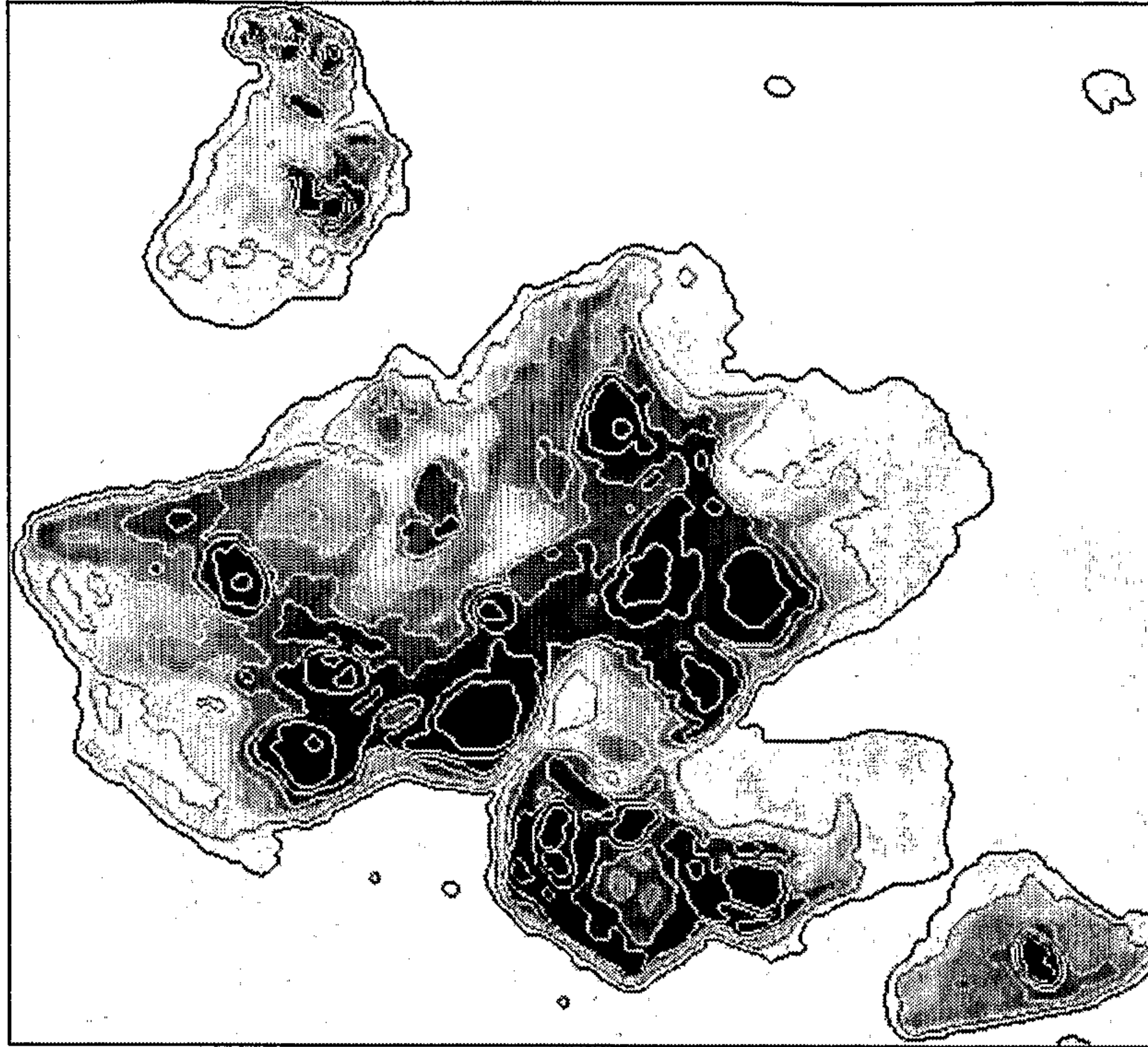


Figure 1

2/7

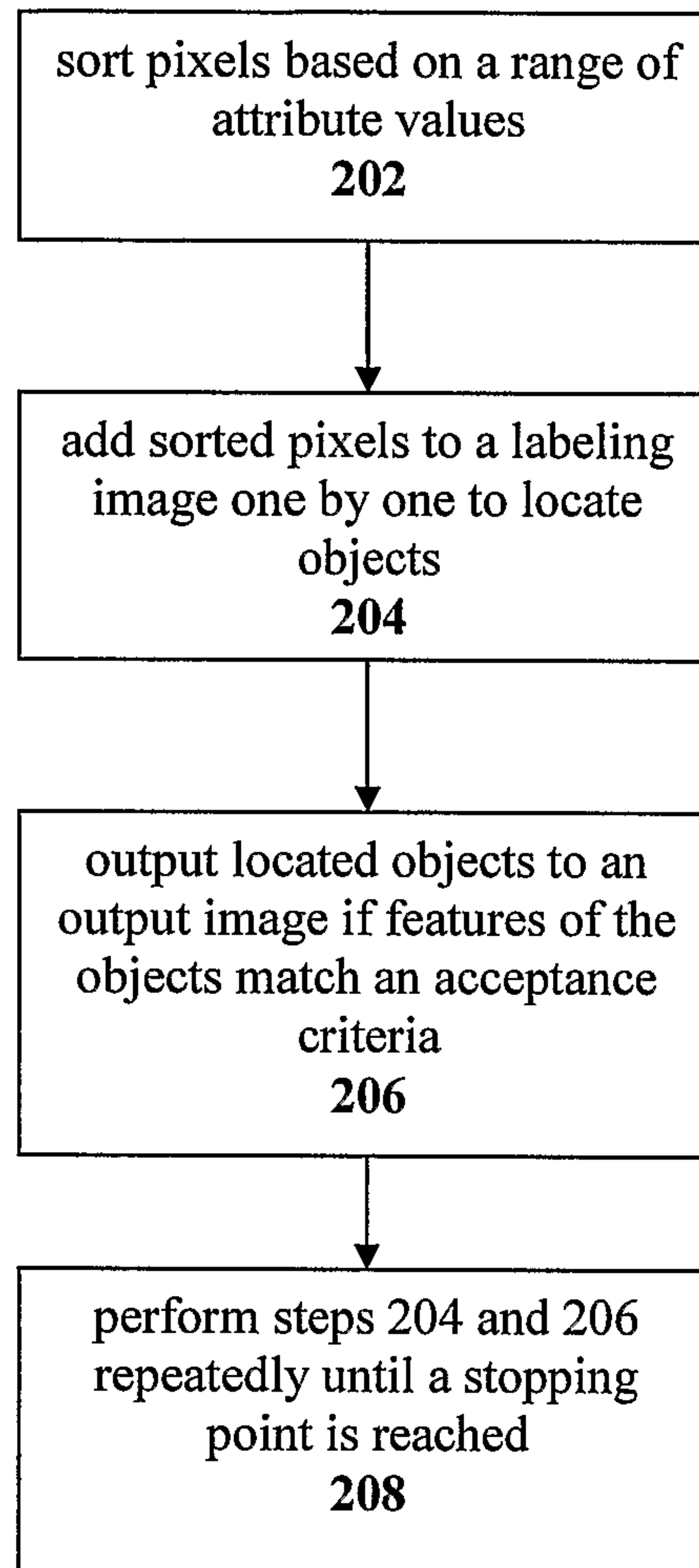


Figure 2

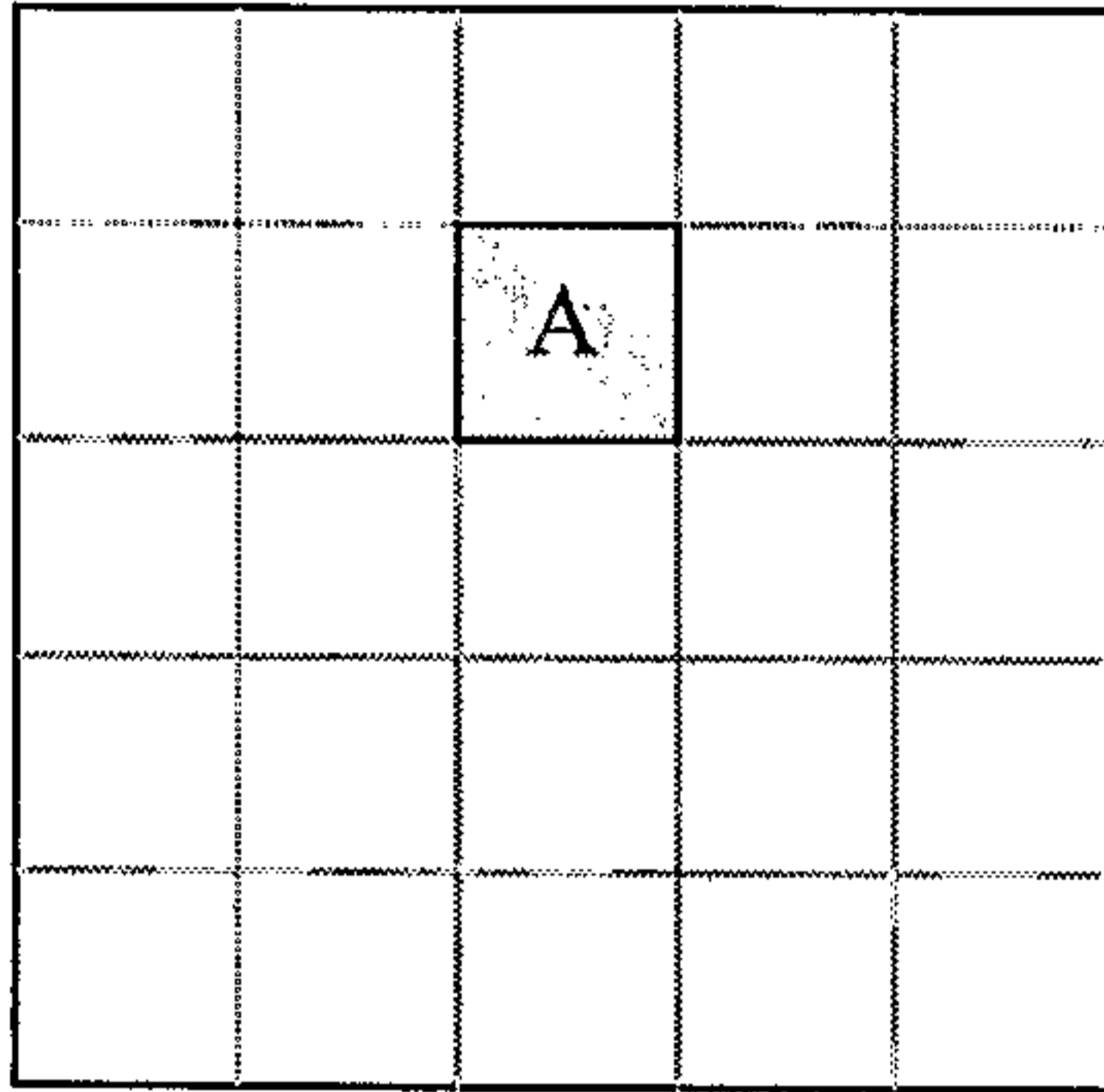


Figure 3a

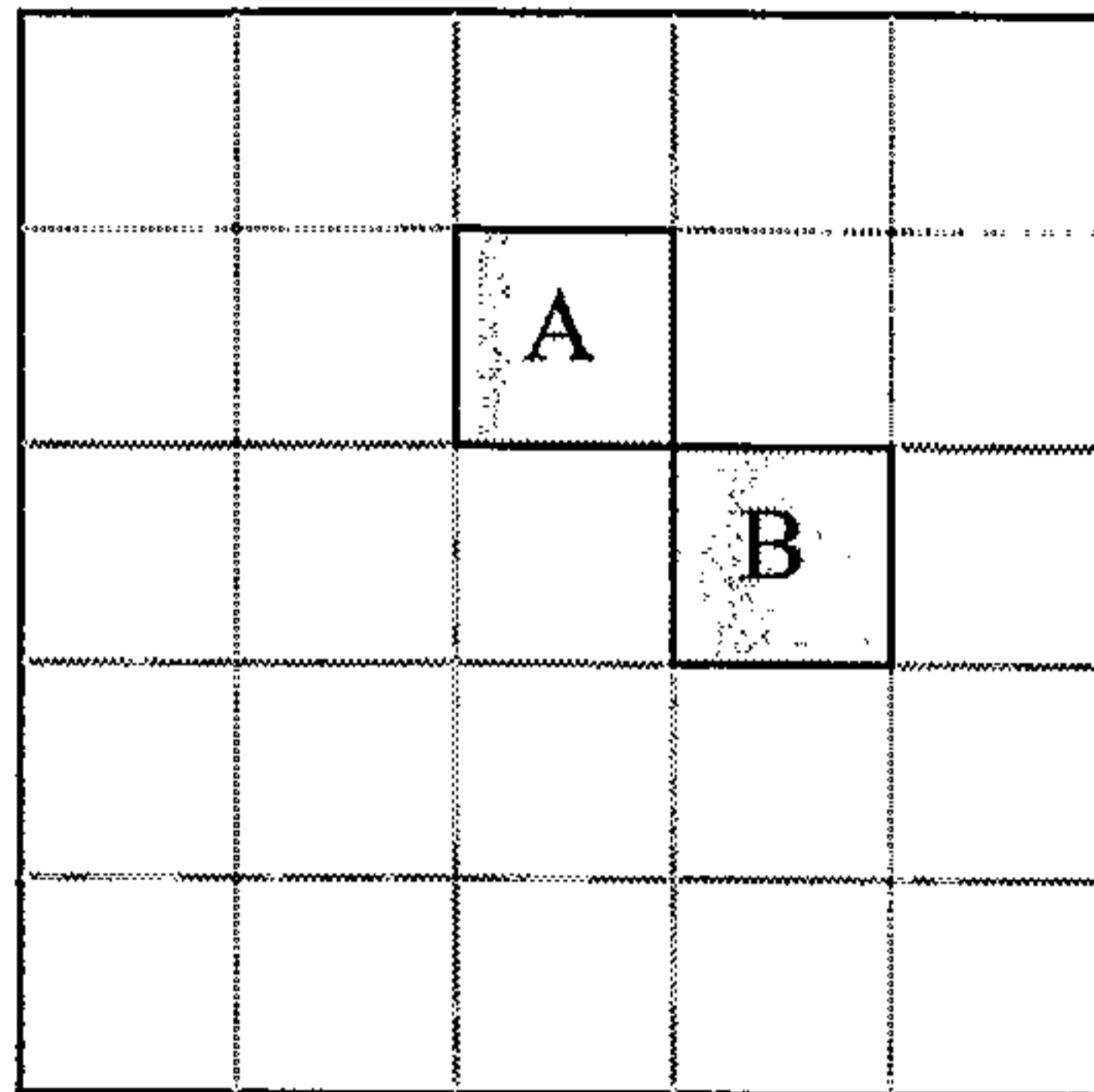


Figure 3b

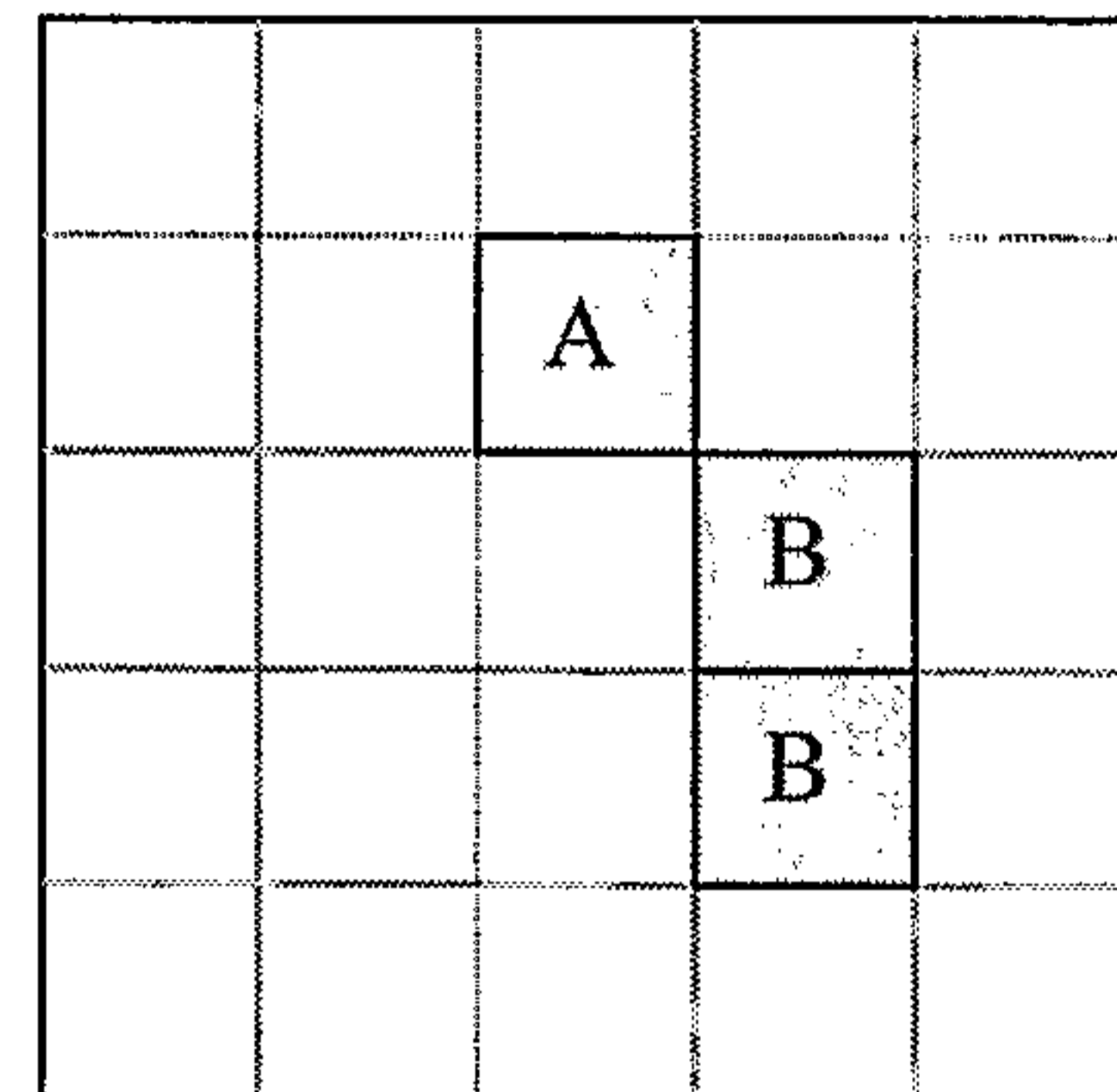


Figure 3c

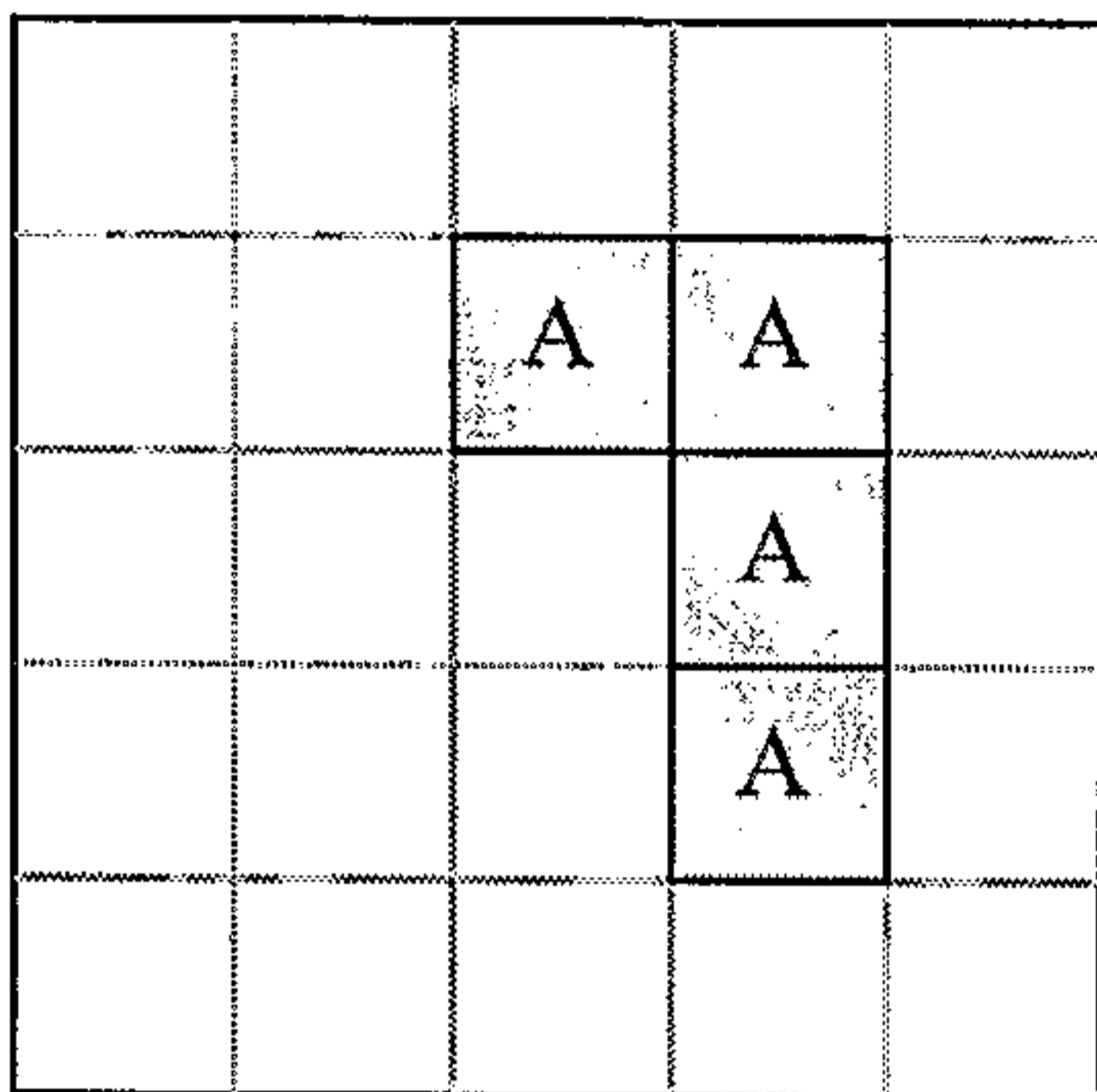


Figure 3d

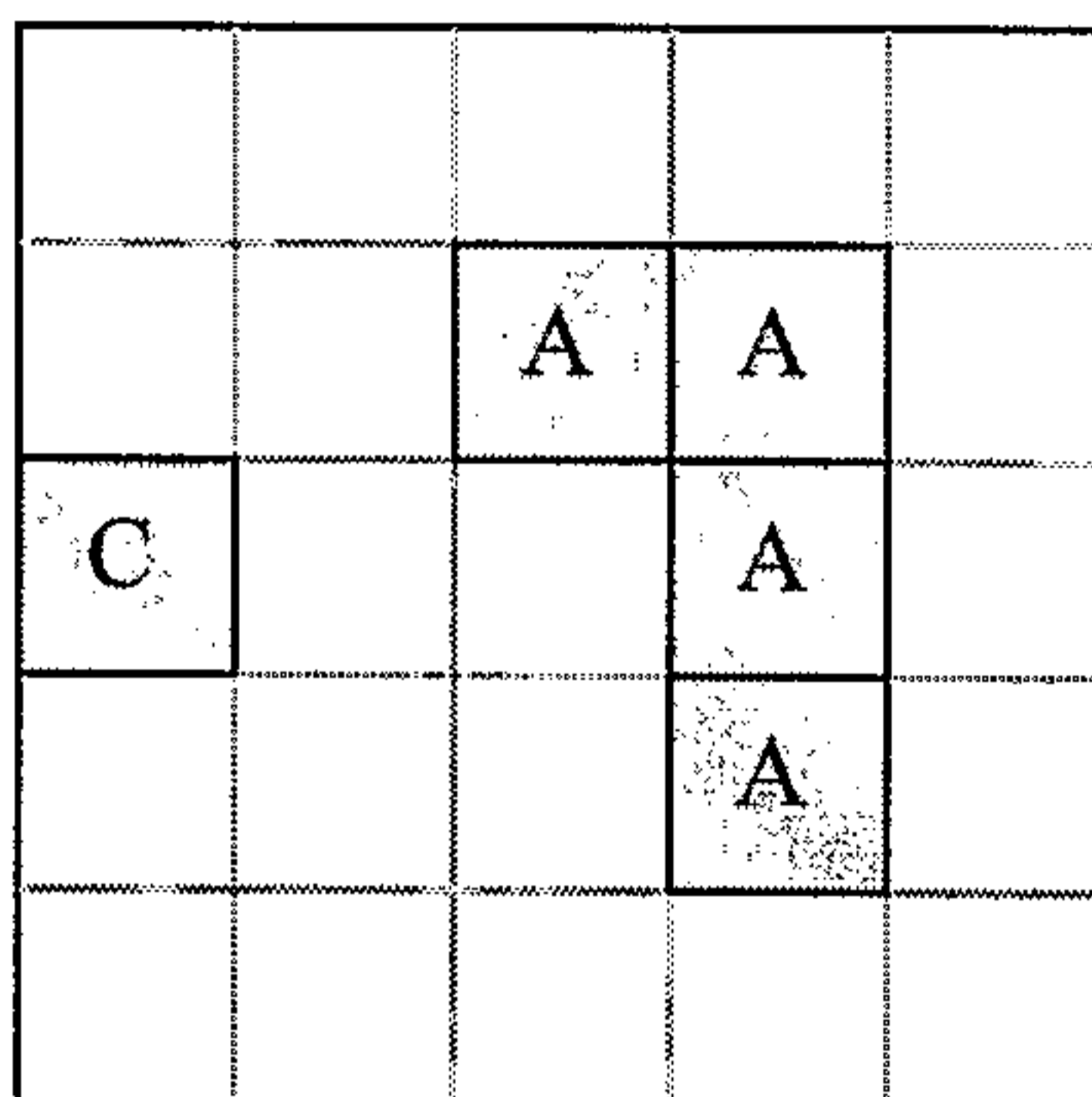


Figure 3e

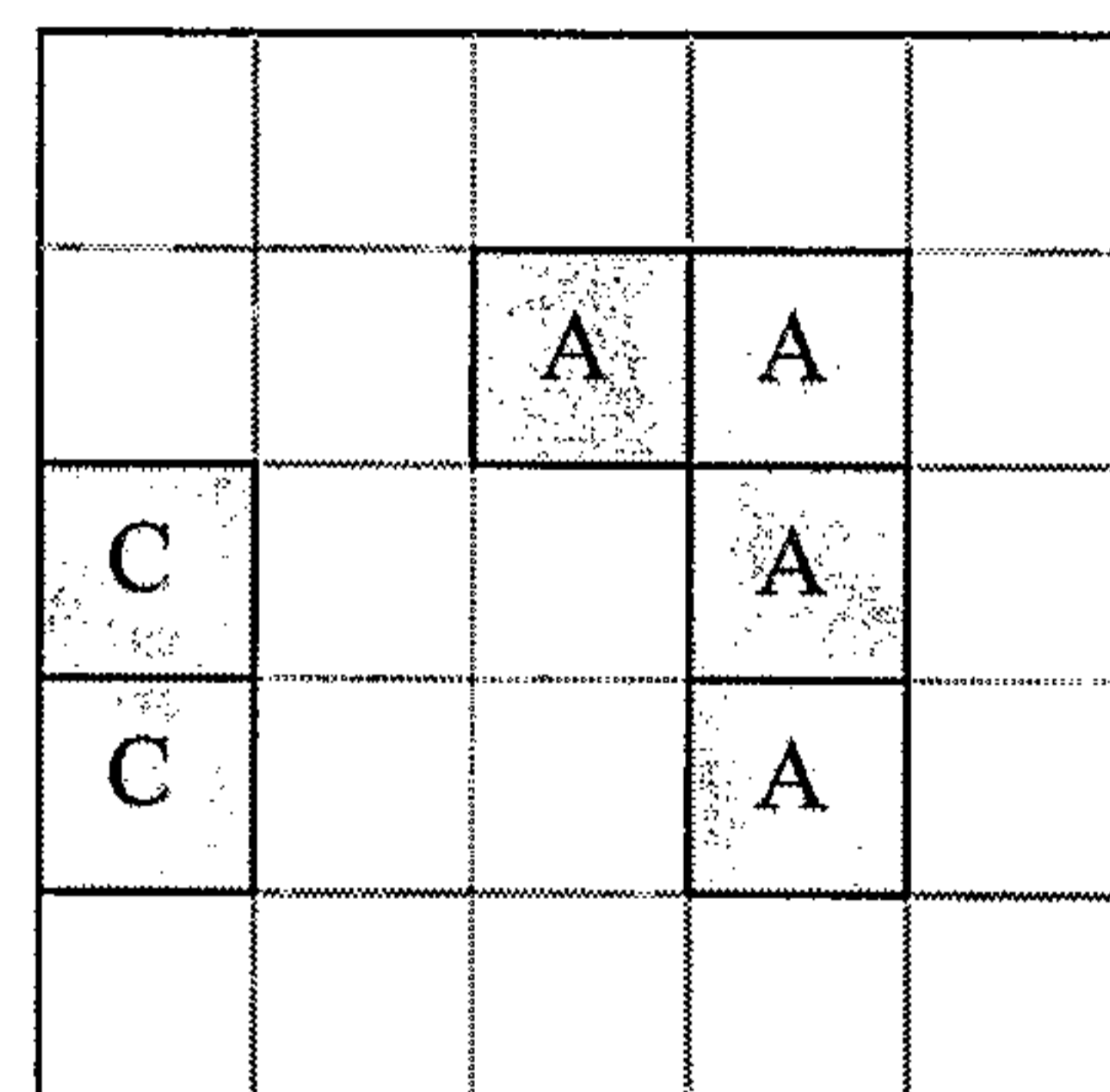


Figure 3f

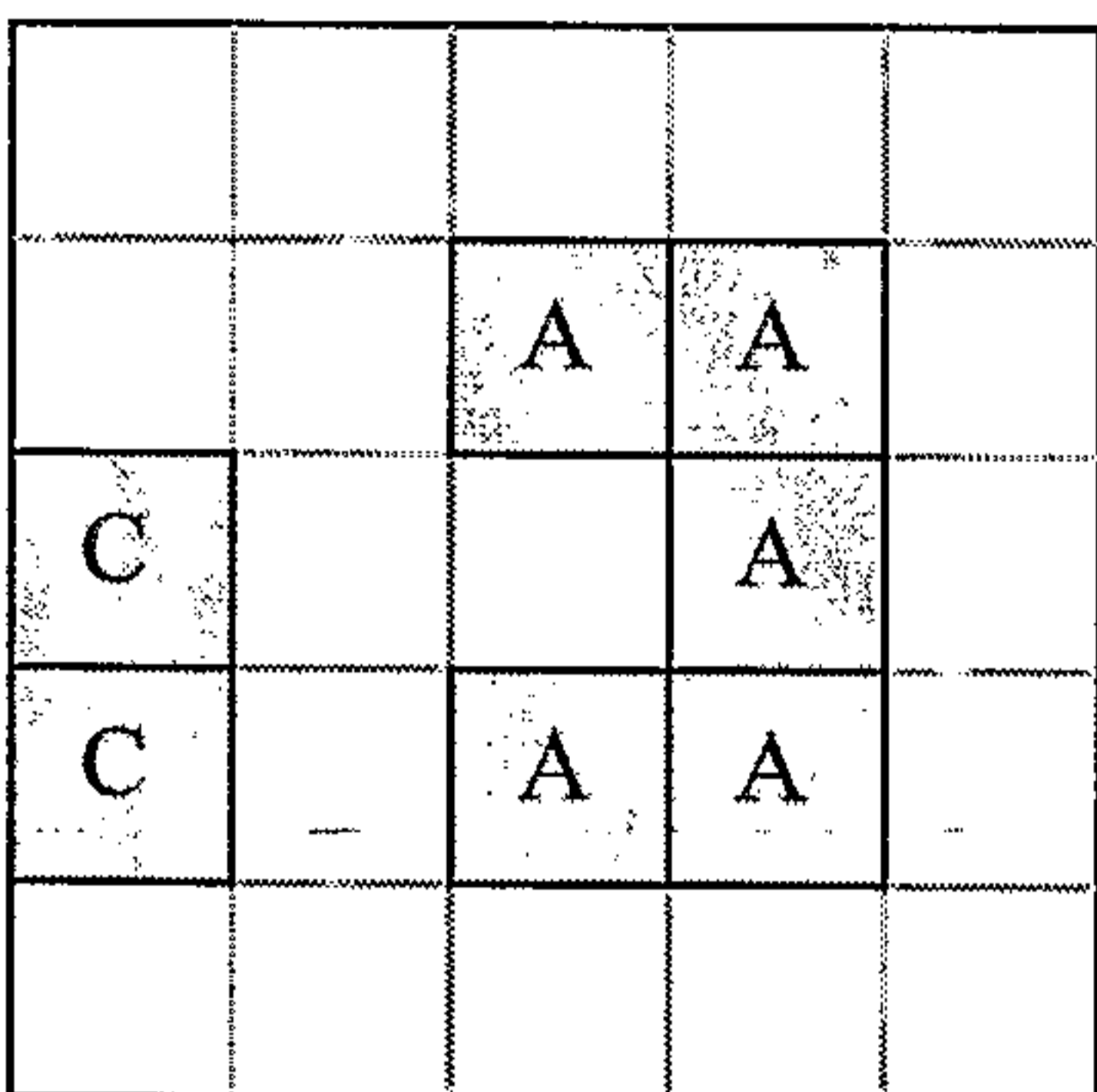


Figure 3g

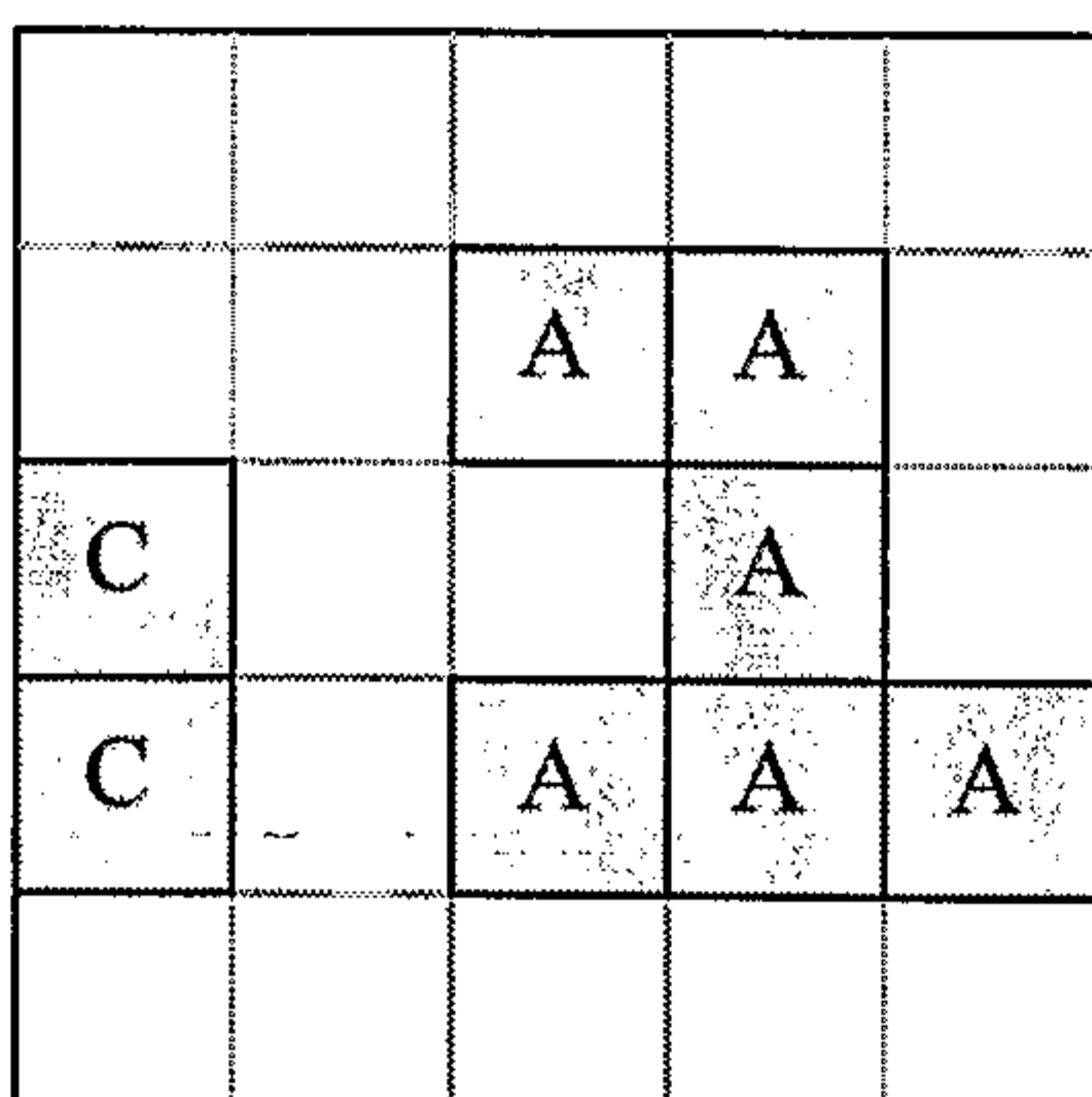


Figure 3h

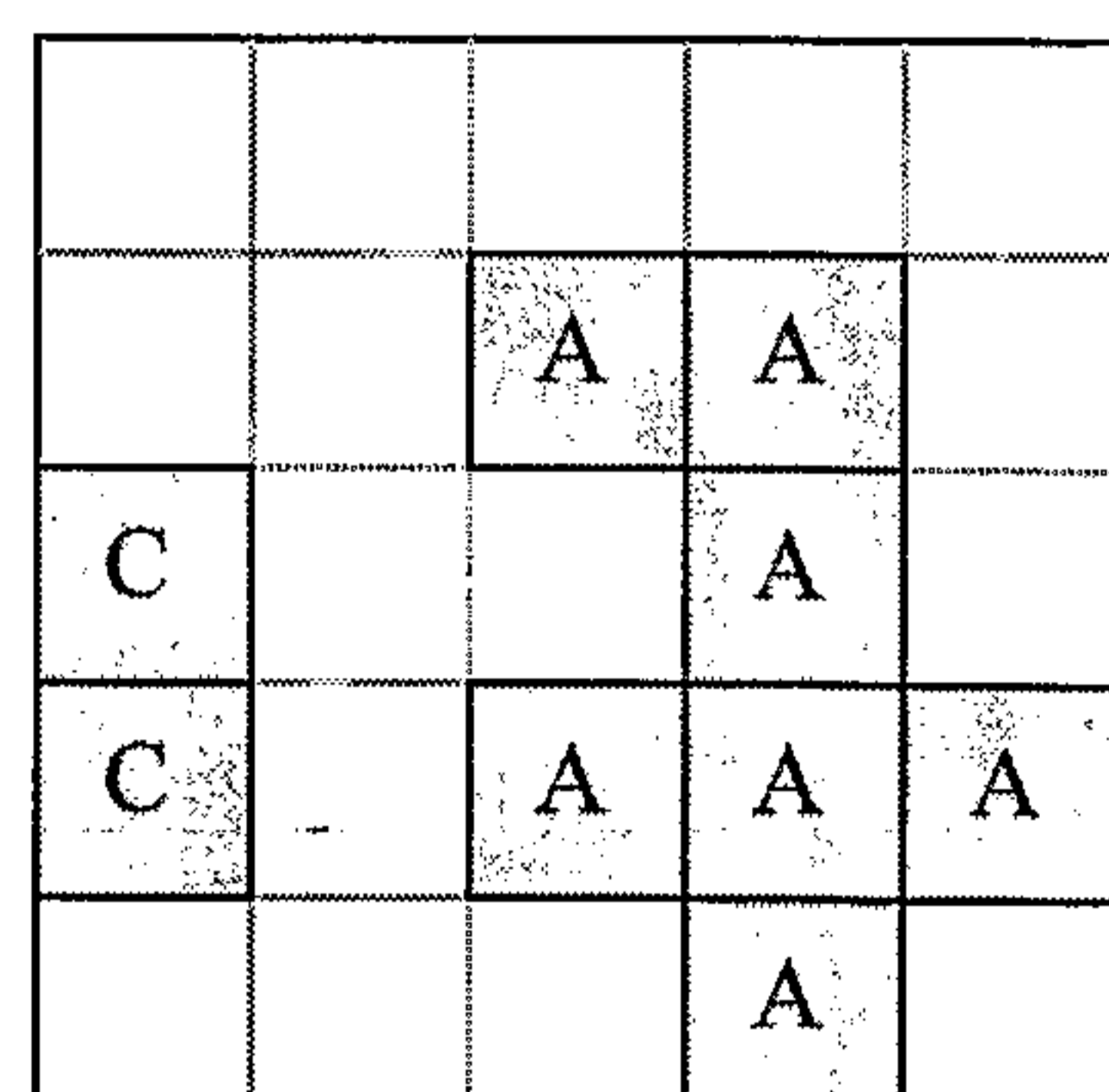


Figure 3i

4/7

		A		
		A	A	
C			A	
C		A	A	A
			A	

Figure 3j

		A		
C		A	A	
C			A	
C		A	A	A
			A	

Figure 3k

		A		
A	A	A	A	
A			A	
A		A	A	A
			A	

Figure 3l

		A		
A	A	A	A	
A		A	A	
A		A	A	A
			A	

Figure 3m

		A		
A	A	A	A	
A		A	A	A
A		A	A	A
			A	

Figure 3n

5/7

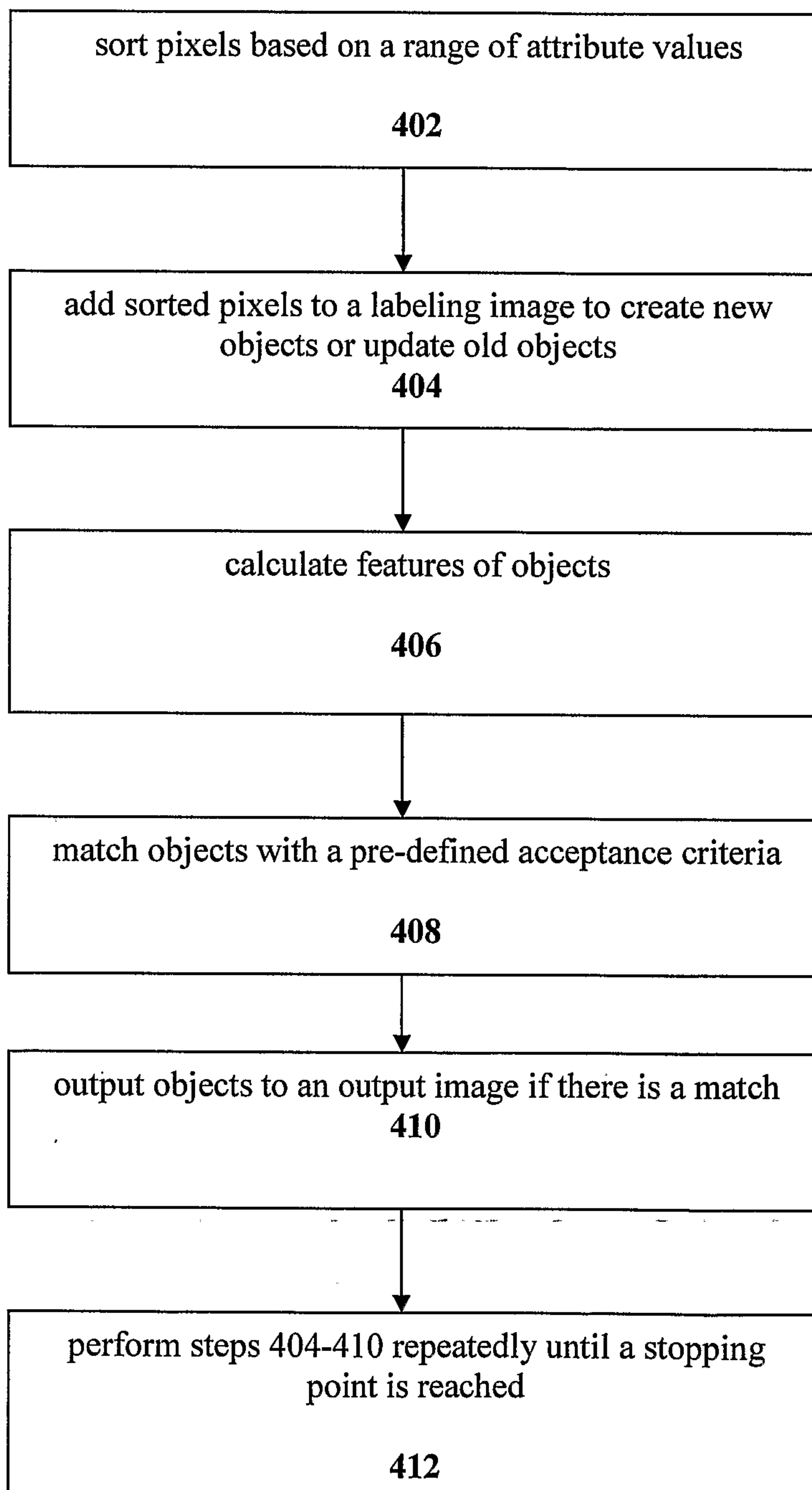


Figure 4

6/7

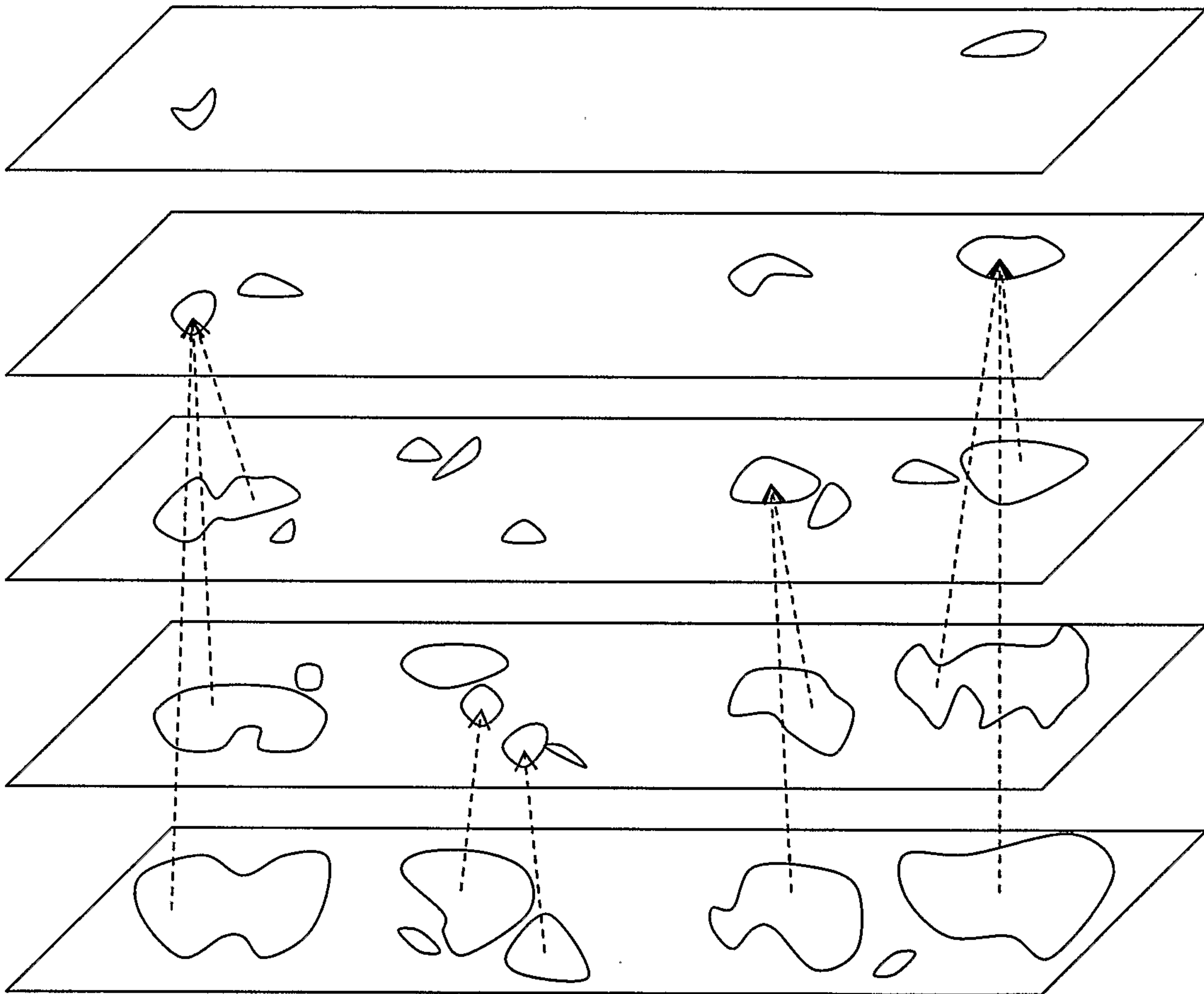


Figure 5

7/7

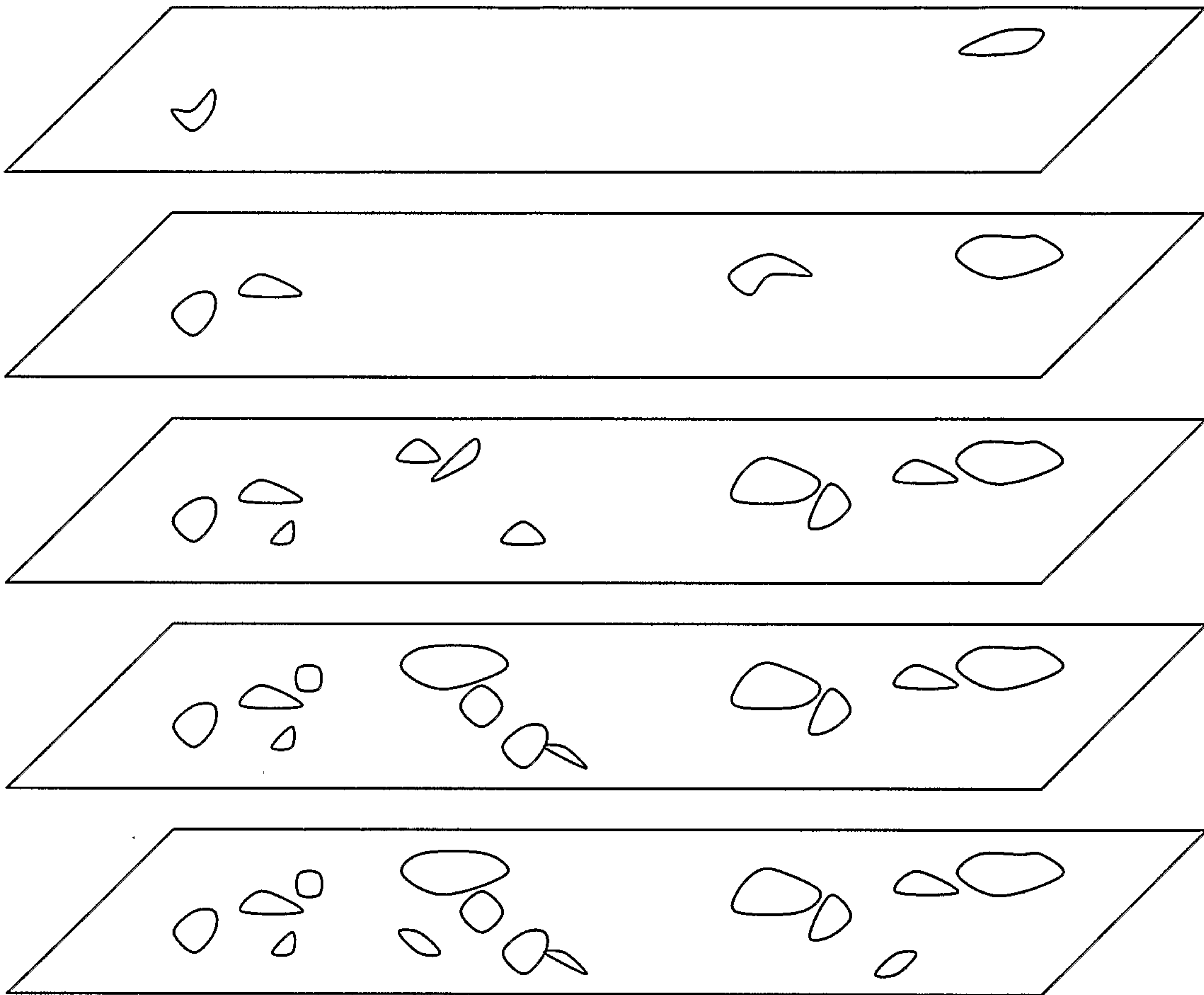


Figure 6

sort pixels based on a range of attribute values

402

add sorted pixels to a labeling image to create new objects or update old objects

404

calculate features of objects

406

match objects with a pre-defined acceptance criteria

408

output objects to an output image if there is a match

410

perform steps 404-410 repeatedly until a stopping point is reached

412