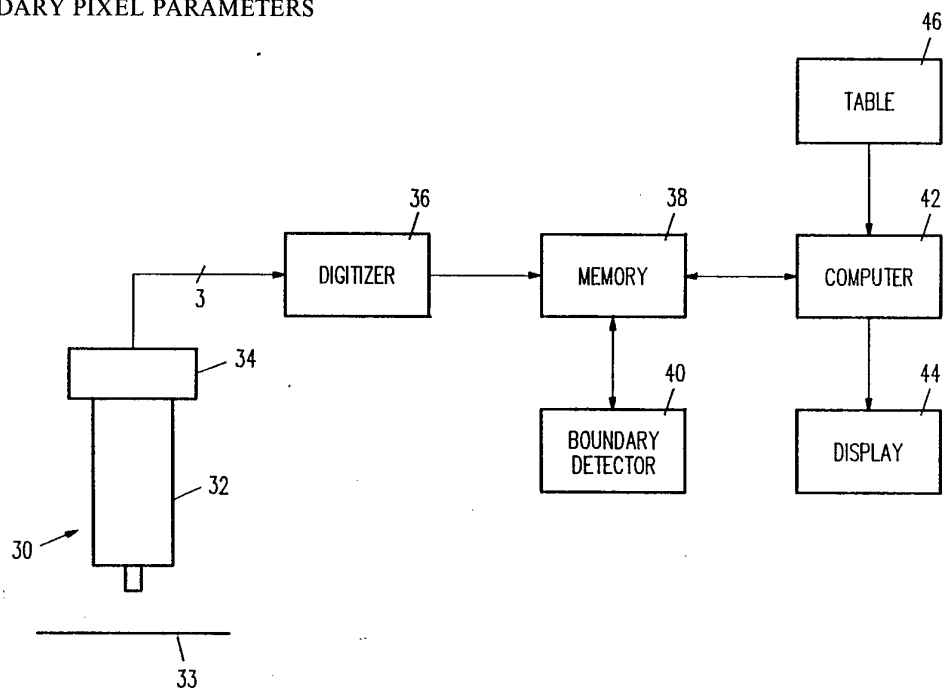




## INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

<b>(51) International Patent Classification <sup>5</sup> :</b>  <b>G06K 9/48</b>	<b>A1</b>	<b>(11) International Publication Number:</b> <b>WO 94/08315</b>  <b>(43) International Publication Date:</b> 14 April 1994 (14.04.94)
<b>(21) International Application Number:</b> PCT/US93/09159 <b>(22) International Filing Date:</b> 24 September 1993 (24.09.93)  <b>(30) Priority data:</b> 07/956,056                      2 October 1992 (02.10.92)      US  <b>(71) Applicant:</b> INTERNATIONAL REMOTE IMAGING SYSTEMS, INC. [US/US]; 9162 Eton Avenue, Chatsworth, CA 91311-5805 (US).  <b>(72) Inventor:</b> KASDAN, Harvey, Lee ; 5414 Sunnyslope Avenue, Van Nuys, CA 91401 (US).  <b>(74) Agents:</b> YIN, Ronald, L. et al.; Limbach & Limbach, 2001 Ferry Building, San Francisco, CA 94111 (US).		<b>(81) Designated States:</b> AU, CA, JP, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).  <b>Published</b> <i>With international search report.          Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i>

**(54) Title:** METHOD AND APPARATUS FOR IDENTIFYING AN OBJECT USING AN ORDERED SEQUENCE OF BOUNDARY PIXEL PARAMETERS



**(57) Abstract**

The present invention relates to a method and apparatus for identifying an examined object having a discernible boundary. An image of the object is formed (34) and segmented into a plurality of pixels (36). The boundary of the object in the image is detected (40) and the parameters of the boundary pixels form an ordered sequence:  $p_1 \dots p_k \dots p_N$ , where  $p_1$  is the parameter value of the first boundary pixel; where  $p_N$  is the parameter value of the last boundary pixel;  $N$  is the total number of boundary pixels and  $k$  is the index to the  $N$  pixels. The  $p_k$  data point is determined (42) such that  $k$  satisfies the relationship (I):  $i-1/M < k/N < i/M$ , where  $M$  is the total number of quantile partitions and  $i$  is the  $i$ th quantile partition. The object is identified based upon the  $p_k$  data point value, which is the  $i$ th quantile in  $M$  partitions developed from the above relationship.

**FOR THE PURPOSES OF INFORMATION ONLY**

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

<b>AT</b>	Austria	<b>FR</b>	France	<b>MR</b>	Mauritania
<b>AU</b>	Australia	<b>GA</b>	Gabon	<b>MW</b>	Malawi
<b>BB</b>	Barbados	<b>GB</b>	United Kingdom	<b>NE</b>	Niger
<b>BE</b>	Belgium	<b>GN</b>	Guinea	<b>NL</b>	Netherlands
<b>BF</b>	Burkina Faso	<b>GR</b>	Greece	<b>NO</b>	Norway
<b>BG</b>	Bulgaria	<b>HU</b>	Hungary	<b>NZ</b>	New Zealand
<b>BJ</b>	Benin	<b>IE</b>	Ireland	<b>PL</b>	Poland
<b>BR</b>	Brazil	<b>IT</b>	Italy	<b>PT</b>	Portugal
<b>BY</b>	Belarus	<b>JP</b>	Japan	<b>RO</b>	Romania
<b>CA</b>	Canada	<b>KP</b>	Democratic People's Republic of Korea	<b>RU</b>	Russian Federation
<b>CF</b>	Central African Republic	<b>KR</b>	Republic of Korea	<b>SD</b>	Sudan
<b>CG</b>	Congo	<b>KZ</b>	Kazakhstan	<b>SE</b>	Sweden
<b>CH</b>	Switzerland	<b>LI</b>	Liechtenstein	<b>SI</b>	Slovenia
<b>CI</b>	Côte d'Ivoire	<b>LK</b>	Sri Lanka	<b>SK</b>	Slovak Republic
<b>CM</b>	Cameroon	<b>LU</b>	Luxembourg	<b>SN</b>	Senegal
<b>CN</b>	China	<b>LV</b>	Latvia	<b>TD</b>	Chad
<b>CS</b>	Czechoslovakia	<b>MC</b>	Monaco	<b>TC</b>	Togo
<b>CZ</b>	Czech Republic	<b>MG</b>	Madagascar	<b>UA</b>	Ukraine
<b>DE</b>	Germany	<b>ML</b>	Mali	<b>US</b>	United States of America
<b>DK</b>	Denmark	<b>MN</b>	Mongolia	<b>UZ</b>	Uzbekistan
<b>ES</b>	Spain			<b>VN</b>	Viet Nam
<b>FI</b>	Finland				

-1-

METHOD AND APPARATUS FOR IDENTIFYING  
AN OBJECT USING AN ORDERED SEQUENCE  
OF BOUNDARY PIXEL PARAMETERS  
\*\*\*\*\*

Technical Field

5           The present invention relates to a method and an  
apparatus for identifying an object under  
examination, wherein an image of the object is  
formed. More particularly, the present invention  
relates to a method and an apparatus for identifying  
an object based upon the object having a discernable  
10           boundary and a novel feature termed "quantile".

Background Of The Invention

          Methods and apparatuses for identifying an  
object are well-known in the art. The art of  
identifying biological samples is replete with  
15           various techniques for identifying the type of  
biological samples under examination. See, for  
example, U.S. Patent No. 4,175,860. Formation of  
histograms of various parameters are also known. See  
for example U. S. Patent No. 3,851,156.

20           Heretofore, one of the known methods for  
identifying an object is to use the size of the  
object. Other parameters of the object which can be  
used to identify an object include color, internal  
optical density, and average intensity within the  
25           boundary of the object.

          In some cases, however, it is not possible to  
identify an object uniquely or to differentiate two  
objects if certain of their parameters yield the same  
result. Thus, for example, if two objects have the  
30           same size and have the same internal optical density  
and have the same average visible light intensity,  
then it is not possible to identify one of the  
objects or to differentiate these two objects, based

-2-

upon these parameters. In addition, some of these parameters are computationally intensive requiring considerable amount of time or computer resources. Thus, the present invention contemplates a novel method and apparatus for identifying an object or to differentiate two objects based upon a new parameter which is termed "quantile".

#### Summary Of The Invention

In the present invention, a method of identifying an object, under examination, which has a discernible boundary is disclosed. The method comprises the steps of forming an image of the object. The image is segmented to form a plurality of pixels. The boundary of the object under examination in the image is then detected. A measurement is made of a parameter of each pixel within the boundary detected. The parameter values of the pixels measured within the boundary detected are formed in an ordered sequence. The particular measured parameter value  $P_k$ , where  $k$  is an index of the ordered sequence, is determined such that the index  $k$  satisfies the following relationship:

$$\frac{i-1}{M} < \frac{k}{N} \leq \frac{i}{M}$$

$N$  - is the total number of pixels;

$M$  - is the total number of quantile partitions

$i$  - index corresponding to the  $i^{\text{th}}$  fraction of partition or  $i^{\text{th}}$  quantile

The object under examination is then identified based upon the  $P_k$  parameter value which is the  $i^{\text{th}}$  quantile within  $M$  partitions developed from the above relationship associated with the object.

-3-

The present invention also contemplates an apparatus to carry out the foregoing method.

#### Brief Description Of The Drawings

5 Figure 1 is a schematic graphical illustration of two objects "a" and "b" which have the same size and shape and the same average intensity but which are different.

10 Figure 2 is a graph showing a histogram of intensity of light versus the number of pixels in the boundary detected.

Figure 3 is a graph showing a cumulative histogram of intensity of light versus the cumulative number of pixels in the boundary detected, and with the quantile feature determined for each object.

15 Figure 4 is a table of pre-determined relationships or quantiles of various known objects.

Figure 5 is an ordered sequence of the values of the parameter measured for the pixels in the boundary.

20 Figure 6 is a block diagram of an apparatus suitable for carrying out the method of the present invention.

#### Detailed Description Of The Drawings

25 Referring to Figure 1, there is shown a schematic diagram of an image 10 of two objects: object "A" 12 and "B" 14. Although the description set forth hereinafter is with reference to the method of the present invention to distinguish object "A" from object "B", it can be seen that the method of  
30 the present invention is equally applicable to identifying a single object which is under examination.

-4-

Each of the objects 12 and 14 has a discernible boundary 18 and 20, respectively. As shown in Figure 1, each of the objects 12 and 14 has the same size and shape. Although for illustration purposes, objects 12 and 14 are shown as having the same size and shape, it is not necessary to the practice of the present invention. The purpose of illustrating the method of the present invention using objects 12 and 14 having the same size and shape is to show how the method of the present invention can be used to distinguish objects 12 and 14 under examination when other parameters, such as size and shape, cannot be used.

Objects 12 and 14, of course, represent images of real objects, such as cells, genes or other biological or non-biological objects. In the method of the present invention, an image 10 is taken of the objects 12 and 14. The image 10 can be formed by using an electronic video camera, such as a CCD (the specific details of an apparatus suitable to carry out the method of the present invention will be discussed hereinafter). The image 10 is then filtered in accordance with the different color filters that can be used to distinguish the type of color of the object. Each of the different color images is segmented, to form a plurality of pixels. Each pixel is then digitized. The boundary of each of the objects 12 and 14 is determined. This can be done, for example, by the method disclosed in U.S. Patent No. 4,538,299, which is incorporated herein by reference.

Once the boundary of each of the objects 12 and 14 is determined, one of the parameters of each pixel within the boundary is measured. One parameter can be the intensity of visible light. Thus, the

-5-

intensity of visible light at each pixel, positioned within the boundary positions 18 and 20, is measured.

One prior art method is to form a histogram of intensity versus the number of pixels. Referring to Figure 2, there is shown two histograms. The histogram of Figure 2a corresponds to the object 12 whereas the histogram of Figure 2b corresponds to the object 14. As can be seen from Figure 1, object 12 has a substantially uniform intensity (the dark image indicates brightness) throughout the entire region within the boundary 18. Thus, the histogram, as shown in Figure 2a, shows substantially  $N_1$  pixels each having an intensity of  $I_1$ . In contrast, object 14 has a small area 16 that has greater intensity than the uniform intensity  $I_1$  of object 12. The histogram, as shown in Figure 2b, shows that  $N_2$  number of pixels in the spot 16 have substantially an intensity value of  $I_2$ , with  $N_1 > N_2$  and  $I_2 > I_1$ .

Once the histogram of each object 12 and 14 has been formed, a cumulative histogram is then developed. The cumulative histogram is formed by summing the number of pixels that has at least a certain intensity value. If  $N = f(I)$ , where  $N$  is the number of pixels having an intensity level  $I$ , then  $\text{Cum } N = F(I)$  where  $\text{Cum } N$  is the cumulative number of pixels having an intensity level  $\leq I$ , and where

$$F(I) = \sum_{j=0}^I f(j)$$

The cumulative histograms of the histograms shown in Figure 2a and 2b, are shown in Figure 3.

The particular parameter value, such as intensity of visible light, associated with a cumulative fraction of all pixels, is termed a quantile. Thus, from Figure 3, the 50th quantile is

-6-

$I_3$  for object 12 and  $I_4$  for object 14. It should be noted that unlike conventional percentile representations, which are even increments of discrete percents, quantile representations are not limited to discrete, even increments. Thus, an object may be identified by its  $i$ th,  $j$ th, or  $k$ th quantile, where  $i$ ,  $j$ , and  $k$  do not necessarily occur in discrete, even increments.

Since the quantile relationship, shown in Figure 3, associated with each objects 12 and 14 is different, the objects 12 and 14 can be distinguished from one another, based upon this developed quantile relationship. Alternatively, to uniquely identify a particular object, such as object 14, the quantile relationship associated with the object 14 can be compared to a table of pre-determined relationship of other known objects whose quantile relationships have been pre-determined. The table of pre-determined quantile relationships of other known objects can be based upon experimental results. The comparison of the quantile relationship to the table of pre-determined quantile relationships would then serve to identify uniquely the type of the object that is under examination.

For the purpose of distinguishing the objects 12 and 14 from one another or to identify an unknown object by comparing the quantile relationship, it is not necessary to compare each ordered pair of numbers  $(Q, I)$  (where  $Q$  is the quantile number and  $I$  is intensity associated therewith), to each other (in the case of two particles) or to a table (in the case of attempting to uniquely identify an object). One method is to compare the associated intensity value of the object under examination at a particular quantile number to the associated intensity value of



-7-

another object at the same quantile number, or to a table of associated intensity values of identified objects at the same quantile number. Thus, as shown in Figure 3, the 50th quantile for the object 12 has an intensity value of  $I_3$  and the same 50th quantile for the object 14 has the value  $I_4$ . These intensity values can be compared to each other to distinguish one object from another. Alternatively, the 50th quantile of one of the objects can be compared to a table of intensity values of identified particles whose 50th quantile have been pre-determined. See Figure 4. By comparing the 50th quantile of the unknown object to the table, the unknown object can be identified.

Of course, if we are dealing with objects of biological particles having statistical variations in intensity, it may not be possible to determine precisely the intensity associated with a particular quantile for the same particle under all cases of examination. Thus, a table relationship may encompass a range of values of intensity for a particular quantile such that the object having that range of intensity values, can be uniquely identified. The range of values can be developed from representative samples identified from experimental results.

The foregoing example illustrates a method of distinguishing objects wherein the average intensity of the two objects 12 and 14 are indistinguishable. However, using the intensity of visible light and the quantiles developed therefor, the objects can be distinguished from one another. Parameters, other than intensity of light can also be used to distinguish the particles. Thus, another parameter suitable for use is differentiating by color

-8-

representation. Color can be represented by three primary colors of red, blue, and green. Alternatively, color can also be represented by hue, intensity, and saturation; or by cyan, magenta and yellow.

5

In one particular embodiment, the difference in color representation of the primary colors is used. The parameter of  $\log(a) - \log(b)$  may be employed, where a, and b are intensities of red and blue primary colors, respectively. Other possible combinations include: a being green and b being red; or a being green and b being blue.

10

The problem with the foregoing method of calculating quantiles is that a cumulative histogram must be formed. This necessitates the creation of a plurality of discrete "bins". A bin is defined by a cumulative fraction of all the pixels from a first fraction to a second fraction. As can be seen from Figure 3, the entire value of the cumulative histogram from 0 to  $I_T$  must be divided into a number of discrete bins. The  $i^{\text{th}}$  quantile is the value of I corresponding to the  $i^{\text{th}}$  cumulative fraction of the pixels. For parameters, such as  $\log(a) - \log(b)$ , as mentioned hereinabove, the range of values can be enormous (due to the nature of the logarithmic operation). Thus, the choice of the boundary of the bins may become difficult.

15

20

25

In the method of the present invention, quantiles can be calculated without calculating a cumulative histogram. In the method of the present invention, the values of the parameters measured for each of the pixels within the boundary detected are placed in an ordered sequence. This is shown in Figure 5. As previously discussed, for the example used in the method of the present invention, the

30

35

-9-

objects 12 and 14 are assumed to be the same size and shape. Thus, each of those objects 12 and 14 would be segmented into the same number of total pixels, N. The parameter, for example of intensity of visible light, for each of the pixels within the boundary is then placed in an ordered sequence.  $P_1$  is the value of the pixel having the lowest intensity of visible light.  $P_N$  is the value of the pixel having the largest value of the parameter, such as intensity visible light. The values of the parameter of the pixels measured are then formed in an ordered sequence. A value of  $P_k$  represents the parameter of the pixel between the smallest value and the largest value with k being the index to the N pixels.

Thereafter, k is chosen such that k satisfies the following relationship:

$$\frac{i-1}{M} < \frac{k}{N} \leq \frac{i}{M}$$

where M is the total number of quantile partitions and i is the  $i^{\text{th}}$  fraction of partition or  $i^{\text{th}}$  quantile. For the  $i^{\text{th}}$  chosen quantile, once k is determined, the intensity value of the data point  $P_k$  would be the intensity value associated with the  $i^{\text{th}}$  quantile, or simply the value of the  $i^{\text{th}}$  quantile. Thereafter, the object under examination can be distinguished from other objects based upon the  $i^{\text{th}}$  quantile.

Alternatively, the object under examination can be distinguished from all other objects based upon a table of predetermined values of corresponding  $i^{\text{th}}$  quantile. This can be accomplished even where the two objects do not have the same number of pixels. Thus, for example, object 1 has  $N_1$  pixels and Object 2 has  $N_2$  pixels. The parameters of the pixels are placed in an ordered sequence as follows:

$P_1 \dots P_{k1} \dots P_{N1}$  - object 1 having  $N_1$  pixels

$P_1 \dots P_{k2} \dots P_{N2}$  - object 2 having  $N_2$  pixels

where  $k$  is the index. For each object, the  $P_k$  data point is determined such that  $k$  satisfies the

5 following:

$$\frac{i-1}{M} < \frac{k}{\text{Total No. of pixels for the object}} \leq \frac{i}{M}$$

10  $M$  - total number of quantile partitions;  
 $i$  -  $i^{\text{th}}$  fraction of partition or  $i^{\text{th}}$  quantile partition

where for Objects 1 and 2  $i$  and  $M$  are chosen to be the same. The resultant value of  $P_k$  for each object is compared to one another to differentiate the  
 15 objects. Of course, as previously discussed, Objects 1 and 2 can be the same type of object even if  $P_{k1}$  does not equal  $P_{k2}$ . There can be a range of values for a particular quantile such that objects having  
 20 values within that range will still be the same type.

As can be seen from the foregoing, with the method of the present invention, no cumulative histogram calculation is made. As a result, there is no need to set the size of the bin to calculate the  
 25 cumulative histogram. A simple comparative relationship to determine the index of the data points is determined from which the value of the data point of the parameter measured is then obtained which corresponds to the quantile of interest.

30 An example of the method of the present invention can be seen with reference to the following Table 1, with an example of three particles: Particle 1, Particle 2, and Particle 3, each having 50 pixels of varying intensity of the same parameter.

-11-

<u>TABLE 1</u>				
	<u>Pixel#</u>	<u>Particle 1</u>	<u>Particle 2</u>	<u>Particle 3</u>
	1	6.55	3.77	2.64
	2	6.98	5.56	2.60
5	3	12.24	6.45	1.92
	4	9.38	4.39	1.82
	5	8.28	5.28	2.25
	6	12.66	6.50	2.46
	7	8.04	3.14	2.23
10	8	6.46	5.78	2.83
	9	10.12	4.18	2.15
	10	7.90	7.58	2.83
	11	8.94	5.45	1.98
	12	9.99	4.47	2.07
15	13	8.78	5.21	2.30
	14	9.16	4.55	1.86
	15	15.12	5.21	2.26
	16	13.26	3.47	1.90
	17	9.04	4.25	8.44
20	18	14.82	4.65	6.72
	19	11.36	4.25	4.98
	20	8.64	5.96	5.61
	21	30.53	6.16	6.36
	22	33.61	4.61	4.70
25	23	30.36	3.42	9.32
	24	30.21	4.38	9.30
	25	31.12	2.87	5.86
	26	27.71	6.49	6.96
	27	32.21	4.38	2.96
30	28	33.27	5.74	4.74
	29	34.31	6.84	8.42
	30	41.23	3.43	6.32
	31	26.76	6.03	6.72
	32	27.83	4.92	8.90
35	33	30.93	5.23	6.64
	34	29.12	6.59	4.02
	35	26.33	4.19	5.53
	36	30.59	4.87	8.84
	37	33.00	4.01	6.19
40	38	29.07	6.34	8.75
	39	26.01	3.96	4.17
	40	36.96	6.95	6.91
	41	34.02	91.44	91.44
	42	30.15	90.50	90.50
45	43	30.13	95.96	95.96
	44	21.58	92.91	92.91
	45	34.80	88.72	88.72
	46	21.75	97.00	97.00
	47	27.42	94.56	94.56
50	48	30.47	94.97	94.97
	49	28.36	94.35	94.35
	50	29.58	93.19	93.19

-12-

The next table, Table 2, shows various statistical values computed for the three particles.

		<u>TABLE 2</u>		
		<u>Particle 1</u>	<u>Particle 2</u>	<u>Particle 3</u>
5	Mean	22.182	22.682	22.518
	Standard Error	1.504	5.053	5.073
	Median	27.088	5.314	6.025
	Mode	#N/A	#N/A	#N/A
	Standard Deviation	10.637	35.729	35.872
10	Variance	113.141	1276.597	1286.779
	Kurtosis	-1.580	0.421	0.402
	Skewness	-0.219	1.547	1.535
	Range	34.763	94.129	95.341
	Minimum	6.464	2.872	1.660
15	Maximum	41.227	97.001	97.001
	Sum	1109.125	1134.118	1125.918
	Count	50	50	50

The next table, Table 3, shows the creation of a plurality (10) of "bins" with each bin containing a count of the number of pixels having a value between a first count to a second count. Thus, Bin #1 contains all the pixels having intensity values between 0 and 10. For particle #1, there would be 13 such pixels. For particles #2 and #3, there would be 40 and 40 respectively. The 10 bins are chosen to partition the maximum range of pixel intensity values into 10 fractions. Since the maximum range of intensity value is zero to 41.23, 97.00, and 97.00 for particles #1, #2 and #3 respectively, the 10 bins are chosen to delineate pixel intensity values of 0-10-20-... -90-100. The column entitled Cumulative Histogram is a cumulative count (shown as a %) of the total number of pixels having an intensity value between zero and the value associated with the Bin #. As can be seen from Table 3, particles #2 and #3 are indistinguishable based upon cumulative histogram. Further, as

-13-

previously discussed, for intensity values based upon  $\log()$ , the value can be large. Thus, the range of pixel intensity values will be large, rendering difficult the division of the intensity values into a manageable number of bins.

5

TABLE 3

Bin #	Intensity Value of Pixels	Particle 1		Particle 2		Particle 3	
		Histogram	Cum.Hist.	Histogram	Cum.Hist.	Histogram	Cum.Hist.
1	0-10	13	26.00%	40	80.00%	40	80.00%
2	10-20	7	40.00%	0	80.00%	0	80.00%
3	20-30	12	64.00%	0	80.00%	0	80.00%
4	30-40	17	98.00%	0	80.00%	0	80.00%
5	40-50	1	100.00%	0	80.00%	0	80.00%
6	50-60	0	100.00%	0	80.00%	0	80.00%
7	60-70	0	100.00%	0	80.00%	0	80.00%
8	70-80	0	100.00%	0	80.00%	0	80.00%
9	80-90	0	100.00%	1	82.00%	1	82.00%
10	90-100	0	100.00%	9	100.00%	9	100.00%



-15-

The following table, Table 4, shows the table of pixel intensity values (Table 1) rearranged in accordance with the method of the present invention. The pixel intensity values are ordered in a sequential order, from highest to lowest. (Of course, the ordered sequence can be from the lowest to the highest.) In this example,  $P_{50}$  for each of the three particles have the values 6.46, 2.87 and 1.66, respectively, while  $P_1$  is 41.23, 97.00 and 97.00 respectively.

TABLE 4

	Rank	Percent	Particle 1		Particle 2		Particle 3	
			Pixel#	Int.	Pixel#	Int.	Pixel#	Int.
	1	100.00%	30	41.23	46	97.00	46	97.00
5	2	97.95%	40	36.96	43	95.96	43	95.96
	3	95.91%	45	34.80	48	94.97	48	94.97
	4	93.87%	29	34.31	47	94.56	47	94.56
	5	91.83%	41	34.02	49	94.35	49	94.35
	6	89.79%	22	33.51	50	93.19	50	93.19
10	7	87.75%	28	33.27	44	92.91	44	92.91
	8	85.71%	37	33.00	41	91.44	41	91.44
	9	83.67%	27	32.31	42	90.50	42	90.50
	10	81.63%	25	31.12	45	88.72	45	88.72
	11	79.59%	33	30.93	10	7.58	23	9.32
15	12	77.55%	36	30.59	40	6.95	24	9.30
	13	75.51%	21	30.53	29	6.84	32	8.90
	14	73.46%	48	30.47	34	6.59	36	8.84
	15	71.42%	23	30.36	6	6.50	38	8.75
	16	69.38%	24	30.21	26	6.49	17	8.44
20	17	67.34%	42	30.15	3	6.45	29	8.42
	18	65.30%	43	30.13	21	6.16	26	6.96
	19	63.26%	50	29.58	31	6.03	40	6.91
	20	61.22%	34	29.12	20	5.96	18	6.72
	21	59.16%	38	29.07	8	5.78	31	6.72
25	22	57.14%	49	28.36	28	5.74	33	6.64
	23	55.10%	32	27.83	2	5.56	21	6.36
	24	53.06%	26	27.71	11	5.45	30	6.32
	25	61.02%	47	27.42	38	5.34	37	6.19
	26	48.97%	31	26.76	5	5.28	25	5.86
30	27	46.93%	25	26.33	33	5.23	20	5.81
	28	44.89%	29	26.01	15	5.21	35	5.53
	29	42.85%	46	21.75	13	5.21	19	4.98
	30	40.81%	44	21.58	32	4.92	28	4.74
	31	38.77%	15	15.12	36	4.87	22	4.70
35	32	36.73%	18	14.82	18	4.65	39	4.17
	33	34.69%	16	13.26	22	4.61	34	4.02
	34	32.65%	6	12.66	14	4.55	27	2.96
	35	30.61%	3	12.24	12	4.47	8	2.83
	36	28.57%	19	11.36	4	4.39	1	2.64
40	37	26.53%	9	10.12	27	4.38	2	2.60
	38	24.48%	12	9.99	24	4.38	6	2.46
	39	22.44%	4	9.38	19	4.25	13	2.30
	40	20.40%	14	9.16	17	4.25	15	2.25
	41	18.36%	17	9.04	35	4.19	5	2.25
45	42	16.32%	11	8.94	9	4.18	7	2.23
	43	14.28%	13	8.78	37	4.01	9	2.15
	44	12.24%	20	8.64	39	3.96	12	2.07
	45	10.20%	1	8.55	1	3.77	11	1.98
	46	8.16%	5	8.28	16	3.47	3	1.92
50	47	6.12%	7	8.04	30	3.43	16	1.90
	48	4.08%	10	7.90	23	3.42	14	1.86
	49	2.04%	2	6.98	7	3.14	4	1.82
	50	0.00%	8	6.46	25	2.87	10	1.66

-17-

where Rank is the order in sequence by the intensity of the pixel, Pixel# is the number of the pixel from Table 1, Intensity is the value of the intensity for the corresponding Pixel# from Table 1, and Percent is the percent of particles that would have intensity value shown under Intensity or less.

Thus, the pixel having the Rank of 3 for Particle 1 is Pixel# 45 from Table 1. Pixel# 45 for Particle 1 has an intensity value of 34.80. Further 95.91% of all pixels in Particle 1 have intensity values of 34.80 or less.

If the number of "bins" or fractions, M is chosen to be 10 (the same as the example based upon a cumulative histogram calculation), and N-number of pixels = 50, then

$$\frac{i-1}{10} < \frac{k_i}{50} \leq \frac{i}{10}$$

where i is the i<sup>th</sup> bin. If i=10

$$\frac{9}{10} < \frac{k_{10}}{50} \leq 1$$

$k_{10} = 46, 47, 48, 49, 50$ .

If we choose  $k_{10} = 50$ , then  $P_{k_{10}} = 6.46, 2.87, 1.66$  for particles #1, #2 and #3, respectively.

Since  $P_{k_{10}} = 2.87$  for particle #2 clearly differs from  $P_{k_{10}} = 1.66$  for particle #3, particles #2 and #3 can be differentiated, while other statistical based analysis such as histogram, and cumulative histogram are unable to differentiate the particles. Furthermore, although a single value was compared, clearly a range of values can be compared. Note that the same holds true for  $P_{k_9}$  ( $K_9 = 45$ ) = 8.55, 3.77, 1.98 and  $P_{k_8}$  ( $K_8 = 40$ ) = 9.16, 4.25, 2.25 for particles #1, #2 and #3 respectfully.

-18-

An apparatus 30 suitable for carrying out the method of the present invention is shown in Figure 6. The apparatus 30 comprises a microscope 32, which is focused on an examination area 33. The examination area can be a  
5 microscopic slide or a flow cell such as that disclosed in U.S. Patent No. 4,338,024. A camera 34 is attached to the microscope 32 and is adapted to image a portion of a suspension having a particle therein. The camera 34 is preferably of a raster scan type and may be a CCD camera  
10 model XC-711 manufactured by Sony. The camera 34 can form an electrical image of the field of view as seen through the microscope 32. Further, the camera 34 segments the image into a plurality of pixels, with an electrical signal corresponding to each pixel of the  
15 image. The camera 34 also outputs a plurality of signals (3) one for each of the colors (Red, Blue, and Green).

Each of the signals from the camera 34 is supplied to a digitizer 36, which digitizes the image intensity of each pixel into an electrical signal representing the  
20 grey scale value. Preferably, the pixel is digitized into a gray scale value between 0 and 255, inclusive.

From the digitizer, the digitized gray scale value is supplied to a memory 38, where the values are stored. The memory 36 can be a RAM. A boundary detector 40, such  
25 as that disclosed in U.S. Patent No. 4,538,299 operates on the image in the memory 38 and detects the boundary of the image detected. The result is also stored in the memory 38.

From the memory 38, a computer 42 operates on the  
30 image and measures a parameter, such as intensity of light, of each pixel within the boundary detected.

In the event the apparatus 30 seeks to determine the identity of a single particle under examination, then the particular quantile value derived by the computer 42 is  
35 compared to the table of stored values 46 of previously

-19-

identified particles. The table 46 can be a ROM or another memory. The result can be displayed on the display 44.

5 In the event the apparatus 30 seeks to differentiate a plurality of objects under examination, then a quantile is developed for the same parameter for each of the pixels in the boundary of each of the images of the different objects. From the quantile, the particles under examination can be differentiated from one another.  
10 The result can be displayed on the display 44.

The computer 42 carries out its tasks in accordance with a computer program. A copy of a program to compute quantiles written in the C language for execution under the MS-DOS operating system is as follows:

```

1 /*      cca.h()      Block Diagram Software      cca.h()
2 **
3 **      Programmer:   H. L. Kasdan
4 **
5 **      Copyright :   1987 by Harvey L. Kasdan
6 **
7 *****
8 **
9 **      NAME          cca.h - chain code analysis header file
10 **
11 **
12 **      SYNOPSIS
13 **
14 **      DESCRIPTION
15 **
16 **      SEE ALSO
17 **
18 **      DIAGNOSTICS
19 **
20 **      BUGS
21 **
22 **      REVISION HISTORY
23 **
24 *****
25 */
26 /*
27 *****
28 **      DEFINES
29 *****
30 */
31 #define MAXX 60
32 #define MAXY 60
33 /*
34 *****
35 **      STRUCTURE DEFINITION
36 *****

```

```

37 */
38 struct ccdef {
39     char ccang;
40     int x;
41     int y;
42 };
43
44 typedef char nBgroup[9];
45
46 typedef struct {
47     int px;
48     int py;
49 } point;
50
51 typedef point nBpoints[9];
52
53 /*
54 *****
55 **
56 *****
57 */
58 int ccabox0();
59 int ccabrko();
60 int ccaconsc();
61 int ccaconss();
62 int ccadcom0();
63 int ccadtp0();
64 int ccadilate();
65 int ccadiss1();
66 int ccadiss2();
67 int ccaenter();
68 int ccaerod1();
69 int ccaerode();
70 int ccafilin();
71 int ccafilil1();
72 int ccafilil2();
73 void ccafilil0();
74 int ccanBbf0();
75 int ccanBfol();
76
77

```

```

/* angular position of next point 0 - 7 */
/* x-position of current point */
/* y-position of current point */
/* chain code definition structure */

/* B-group of characters (pixels) */

/* x-coordinate */
/* y-coordinate */
/* point coordinates */

/* array of central point and 8 neighbors */

```

```

*****
SUBROUTINE DECLARATIONS
*****

```

```

1 /*      ndap.h()      Block Diagram Software      ndap.h()
2 **
3 **      Programmer:      H. L. Kasdan
4 **
5 **      Copyright :      (C) 1988 by
6 **                      International Remote Imaging Systems Inc.
7 **
8 **      *****
9 **
10 **      NAME      ndap.h - NIH data analysis header file
11 **
12 **
13 **      SYNOPSIS
14 **
15 **      DESCRIPTION
16 **
17 **      SEE ALSO
18 **
19 **      DIAGNOSTICS
20 **
21 **      BUGS
22 **
23 **      REVISION HISTORY
24 **
25 **      *****
26 #/
27 /#
28 *****
29 *****
30 *****

```

DEFINES



```

31 */
32 #define CAXMAX 50
33 #define CAYMAX 50
34 /*
35 *****
36 **
37 *****
38 */
39 struct image0 {
40     char fname[14];
41     int cellno;
42     long int ri0d;
43     long int gi0d;
44     long int bi0d;
45     long int ri0t;
46     long int gi0t;
47     long int bi0t;
48     int carea;
49     int window;
50     int thresh;
51     char ctype[12];
52     int cx0;
53     int cy0;
54     int cxmin;
55     int cymin;
56     int cxmax;
57     int cymax;
58     /* cell array x-maximum */
59     /* cell array y-maximum */
60     /* cell data file name and extension */
61     /* cell number */
62     /* red iod */
63     /* green iod */
64     /* blue iod */
65     /* red iot */
66     /* green iot */
67     /* blue iot */
68     /* cell area */
69     /* box window size */
70     /* edge trace threshold */
71     /* cell type */
72     /* cell trace starting x-position */
73     /* cell trace starting y-position */
74     /* cell trace minimum x-position */
75     /* cell trace minimum y-position */
76     /* cell trace maximum x-position */
77     /* cell trace maximum y-position */

```

```

57 int cymax;
58 int cpie;
59 char ccc[256];
60 unsigned char crimg[CAYMAX][CAXMAX];
61 unsigned char cgimg[CAYMAX][CAXMAX];
62 unsigned char cbimg[CAYMAX][CAXMAX];
63 unsigned char cnrimg[CAYMAX][CAXMAX];
64 unsigned char cngimg[CAYMAX][CAXMAX];
65 unsigned char cnbimg[CAYMAX][CAXMAX];
66 int lgmlr[CAYMAX][CAXMAX];
67 int lbmlg[CAYMAX][CAXMAX];
68 int lrmlb[CAYMAX][CAXMAX];
69 int qlgmlr[50];
70 int qlbmlg[50];
71 int qlrmlb[50];
72 int mcigmlr;
73 int mcibmlg;
74 int mclrmlb;
75 int sdcigmlr;
76 int sdcibmlg;
77 int sdcrlrmlb;
78 char ccimg[CAYMAX][CAXMAX];
79 struct ccddef cccimg[256];
80 };
81 struct histo {
82     unsigned int rhist0[256];
83     unsigned int ghist0[256];
84     unsigned int bhist0[256];
85     unsigned int rihist0[256];
86     unsigned int gihist0[256];
87     unsigned int bihist0[256];
88     unsigned int rehist0[256];
89     unsigned int gehist0[256];
90     unsigned int behist0[256];
91 };
92 struct chisto {
    /* cell trace maximum y-position */
    /* cell points in edge */
    /* cell chain code */
    /* cell red image */
    /* cell green image */
    /* cell blue image */
    /* cell normalized red image */
    /* cell normalized green image */
    /* cell normalized blue image */
    /* log green minus log red image */
    /* log blue minus log green image */
    /* log red minus log blue image */
    /* log green minus log red cell quantiles */
    /* log blue minus log green cell quantiles */
    /* log red minus log blue cell quantiles */
    /* log green minus log red cell pixel mean */
    /* log blue minus log green cell pixel mean */
    /* log red minus log blue cell pixel mean */
    /* cell chain code image */
    /* cell chain code structure */
    /* red image histogram */
    /* green image histogram */
    /* blue image histogram */
    /* red image interior histogram */
    /* green image interior histogram */
    /* blue image interior histogram */
    /* red image exterior histogram */
    /* green image exterior histogram */
    /* blue image exterior histogram */
    /* cell image data histogram structure */
};

```

```

73     unsigned int crhist0[256];          /* red image cum histogram */
74     unsigned int cghist0[256];         /* green image cum histogram */
75     unsigned int cbhist0[256];         /* blue image cum histogram */
76     unsigned int crihist0[256];        /* red image interior cum histogram */
77     unsigned int cgihist0[256];        /* green image interior cum histogram */
78     unsigned int cbihist0[256];        /* blue image interior cum histogram */
79     unsigned int crehist0[256];        /* red image exterior cum histogram */
100    unsigned int cgehist0[256];        /* green image exterior cum histogram */
101    unsigned int cbehist0[256];        /* blue image exterior cum histogram */
102    },
103    struct qt10 {
104        unsigned int qr0[50];          /* red image quantiles */
105        unsigned int qg0[50];          /* green image quantiles */
106        unsigned int qb0[50];          /* blue image quantiles */
107        unsigned int qri0[50];         /* red image interior quantiles */
108        unsigned int qgi0[50];         /* green image interior quantiles */
109        unsigned int qbi0[50];         /* blue image interior quantiles */
110        unsigned int qre0[50];         /* red image exterior quantiles */
111        unsigned int qge0[50];         /* green image exterior quantiles */
112        unsigned int qbe0[50];         /* blue image exterior quantiles */

```

```

113 unsigned int mr0;
114 unsigned int mg0;
115 unsigned int mb0;
116 unsigned int mri0;
117 unsigned int mgi0;
118 unsigned int mbi0;
119 unsigned int mre0;
120 unsigned int mge0;
121 unsigned int mbe0;
122 unsigned int sdr0;
123 unsigned int sdg0;
124 unsigned int sdb0;
125 unsigned int sdri0;
126 unsigned int sdgi0;
127 unsigned int sdbi0;
128 unsigned int sdre0;
129 unsigned int sdge0;
130 unsigned int sdbe0;
131 };
132 /*
133 *****
134 ** SUBROUTINE DECLARATIONS
135 *****
136 */
137 char #1btb1( );
/* red image mean */
/* green image mean */
/* blue image mean */
/* red image interior mean */
/* green image interior mean */
/* blue image interior mean */
/* red image exterior mean */
/* green image exterior mean */
/* blue image exterior mean */
/* red image std deviation */
/* green image std deviation */
/* blue image std deviation */
/* red image interior std deviation */
/* green image interior std deviation */
/* blue image interior std deviation */
/* red image exterior std deviation */
/* green image exterior std deviation */
/* blue image exterior std deviation */
/* qt10 - cell image data quantile structure */
*****
SUBROUTINE DECLARATIONS
*****

```

```

1 /*      ndaphO()      Analysis Software      ndaphO()
2 **
3 **      Programmer:   H. L. Kasdan
4 **
5 **      Copyright :   1988 by
6 **                    International Remote Imaging Systems, Inc.
7 **
8 *****
9 **
10 **      NAME      ndaphO - image histogram routine
11 **
12 **
13 **      SYNOPSIS
14 **      #include <stdio.h>
15 **      #include <string.h>
16 **      #include "cca.h"
17 **      #include "ndap.h"
18 **      int ndaphO( int mode, struct imageO *p_imageO,
19 **                struct histo *p_histo, struct chisto *p_chisto )
20 **      int mode;
21 **      struct imageO *p_imageO;    histogram formation mode
22 **      struct histo *p_histo;     pointer to imageO data structure
23 **      struct chisto *p_chisto;  pointer to histo data structure
24 **
25 **      DESCRIPTION
26 **      Routine generates histogram to histo data structure
27 **      using data from imageO data structure according to
28 **      specified mode.
29 **      mode      operation
30 **      -----
31 **      0         use un-normalized image data from imageO
32 **      1         use normalized image data from imageO
33 **
34 **      SEE ALSO

```

```

35 **
36 **      DIAGNOSTICS
37 **
38 **      BU05
39 **
40 **      REVISION HISTORY          h1k      modified for *.URN file precursor to use window size
41 **      900507
42 **
43 **
44 **/
45 #include <stdio.h>
46 #include <string.h>
47 #include "cca.h"
48 #include "ndap.h"
49 #define FALSE 0
50 #define TRUE 1
51 #define PMAXX 50
52 #define PMAXY 50
53 #define PMINX 0
54 #define PMINY 0
55
56
57 int ndaph0( mode, p_image0, p_histo, p_chisto )
58     int mode; /* histogram formation mode */
59     struct image0 *p_image0; /* pointer to image0 data structure */
60     struct histo *p_histo; /* pointer to histo data structure */
61     struct chisto *p_chisto; /* pointer to chisto data structure */
62     {
63     /*
64     **
65     **      LOCAL STORAGE
66     **
67     **

```

```

67 */
68 int err = 0;
69 int i;
70 int x;
71 int y;
72 /*
73 *****
74 ** Do Histogram According to Mode
75 *****
76 */
77 switch (mode)
78 {
79 case 1: /* normalized image is data source */
80
81     for ( i = 0; i < 256; i++)
82     {
83         p_hist0->rhist0[i] = 0;
84         p_hist0->ghist0[i] = 0;
85         p_hist0->bhist0[i] = 0;
86         p_hist0->rhist0[i] = 0;
87         p_hist0->ghist0[i] = 0;
88         p_hist0->bhist0[i] = 0;
89         p_hist0->rehist0[i] = 0;
90         p_hist0->gehist0[i] = 0;
91         p_hist0->behist0[i] = 0;
92     } /* end - for ( i = 0; i < 256; i++) */
93
94     for (y = PMINY; y < p_image0->window; y++)
95     {
96         for (x = PMINX; x < p_image0->window; x++)
97         {
98             p_hist0->rhist0[(int)p_image0->cnrimg[y][x]] += 1;
99             p_hist0->ghist0[(int)p_image0->cnimg[y][x]] += 1;
100             p_hist0->bhist0[(int)p_image0->cnbimg[y][x]] += 1;
101             if ((p_image0->ccimg[y][x] == 'x') ||
102                 ((p_image0->ccimg[y][x] >= '0') && (p_image0->ccimg[y][x] < '8'))))
103             {
104                 /*

```

```

105 **          Do Interior
106 */
107 p_histo->rihisto[(int)p_image0->cnrimg[y][x]] += 1;
108 p_histo->gihisto[(int)p_image0->cnngimg[y][x]] += 1;
109 p_histo->bihisto[(int)p_image0->cnbimg[y][x]] += 1;
110 } /* end - if (... || ...) */
111 else
112 {
113 /*
114 **
115 */          Do Exterior
116 p_histo->rehisto[(int)p_image0->cnrimg[y][x]] += 1;
117 p_histo->gehisto[(int)p_image0->cnngimg[y][x]] += 1;
118 p_histo->behisto[(int)p_image0->cnbimg[y][x]] += 1;
119 } /* end - else (... || ...) */
120 } /* end - for (x = PMINX; x < p_image0->window; x++) */
121 } /* end - for (y = PMINY; y < p_image0->window; y++) */
122
123 /*
124 **
125 */          Compute Cumulative Histogram
126 p_chisto->crhisto[0] = p_histo->rhisto[0];
127 p_chisto->cghisto[0] = p_histo->ghisto[0];
128 p_chisto->cbhisto[0] = p_histo->bhisto[0];
129 p_chisto->crihisto[0] = p_histo->rihisto[0];
130 p_chisto->cgihisto[0] = p_histo->gihisto[0];
131 p_chisto->cbehisto[0] = p_histo->behisto[0];
132 p_chisto->crehisto[0] = p_histo->rehisto[0];
133 p_chisto->cgehisto[0] = p_histo->gehisto[0];
134 p_chisto->cbehisto[0] = p_histo->behisto[0];
135
136 for ( i = 1; i < 256; i++)
137 {

```



```

138 p_chisto->crhisto[i] = p_histo->rhisto[i] + p_chisto->crhisto[i - 1];
139 p_chisto->cgihisto[i] = p_histo->ghisto[i] + p_chisto->cgihisto[i - 1];
140 p_chisto->cbhisto[i] = p_histo->bhisto[i] + p_chisto->cbhisto[i - 1];
141 p_chisto->crihisto[i] = p_histo->rhisto[i] + p_chisto->crihisto[i - 1];
142 p_chisto->cgihisto[i] = p_histo->ghisto[i] + p_chisto->cgihisto[i - 1];
143 p_chisto->cbihisto[i] = p_histo->bihisto[i] + p_chisto->cbihisto[i - 1];
144 p_chisto->crehisto[i] = p_histo->rehisto[i] + p_chisto->crehisto[i - 1];
145 p_chisto->cgehisto[i] = p_histo->gehisto[i] + p_chisto->cgehisto[i - 1];
146 p_chisto->cbehisto[i] = p_histo->behisto[i] + p_chisto->cbehisto[i - 1];
147 } /* end - for ( i = 0; i < 256; i++) */
148 break;
149
150 case 0: default: /* use un-normalized image data */
151
152 for ( i = 0; i < 256; i++)
153 {
154     p_histo->rhisto[i] = 0;
155     p_histo->ghisto[i] = 0;
156     p_histo->bhisto[i] = 0;
157     p_histo->rhisto[i] = 0;
158     p_histo->ghisto[i] = 0;
159     p_histo->bihisto[i] = 0;
160     p_histo->rehisto[i] = 0;
161     p_histo->gehisto[i] = 0;
162     p_histo->behisto[i] = 0;
163 } /* end - for ( i = 0; i < 256; i++) */
164
165 for (y = PMINY; y < p_image0->window; y++)
166 {
167     for (x = PMINX; x < p_image0->window; x++)
168     {

```

```

169 p_hist0->rhistol(int)p_image0->crimg[y][x]] += 1;
170 p_hist0->ghistol(int)p_image0->cgimg[y][x]] += 1;
171 p_hist0->bhistol(int)p_image0->cbimg[y][x]] += 1;
172 if ((p_image0->ccimg[y][x] == 'x') ||
173     ((p_image0->ccimg[y][x] >= '0') && (p_image0->ccimg[y][x] < 'B'))))
174 {
175     /*
176     **
177     */
178     Do Interior
179     p_hist0->rihistol(int)p_image0->crimg[y][x]] += 1;
180     p_hist0->gihistol(int)p_image0->cgimg[y][x]] += 1;
181     p_hist0->bihistol(int)p_image0->cbimg[y][x]] += 1;
182 } /* end - if (... || ...) */
183 else
184 {
185     /*
186     **
187     */
188     Do Exterior
189     p_hist0->rehistol(int)p_image0->crimg[y][x]] += 1;
190     p_hist0->gehistol(int)p_image0->cgimg[y][x]] += 1;
191     p_hist0->behistol(int)p_image0->cbimg[y][x]] += 1;
192 } /* end - else (... || ...) */
193 } /* end - for (y = PMINY; y < p_image0->window; y++) */
194
195 /*
196 **
197 */
198 Compute Cumulative Histogram
199 p_chisto->crhisto[] = p_hist0->rhistol[];
200 p_chisto->cghisto[] = p_hist0->ghistol[];
201 p_chisto->cbhisto[] = p_hist0->bhistol[];
202 p_chisto->crihisto[] = p_hist0->rihistol[];
203 p_chisto->cghisto[] = p_hist0->gihistol[];
204 p_chisto->cbhisto[] = p_hist0->bihistol[];
205 p_chisto->crehisto[] = p_hist0->rehistol[];
206 p_chisto->cehisto[] = p_hist0->gehisto[];
207 p_chisto->cbhisto[] = p_hist0->behistol[];

```

```

206
207     for ( i = 1; i < 256; i++)
208     {
209         p_hist0->crhist0[i] = p_hist0->rhist0[i] + p_hist0->crhist0[i] - 1];
210         p_hist0->cghist0[i] = p_hist0->ghist0[i] + p_hist0->cghist0[i] - 1];
211         p_hist0->cbhist0[i] = p_hist0->bhist0[i] + p_hist0->cbhist0[i] - 1];
212         p_hist0->crihist0[i] = p_hist0->rihist0[i] + p_hist0->crihist0[i] - 1];
213         p_hist0->cgihist0[i] = p_hist0->gihist0[i] + p_hist0->cgihist0[i] - 1];
214         p_hist0->cbihist0[i] = p_hist0->bihist0[i] + p_hist0->cbihist0[i] - 1];
215         p_hist0->crehist0[i] = p_hist0->rehist0[i] + p_hist0->crehist0[i] - 1];
216         p_hist0->cgehist0[i] = p_hist0->gehist0[i] + p_hist0->cgehist0[i] - 1];
217         p_hist0->cbehist0[i] = p_hist0->behist0[i] + p_hist0->cbehist0[i] - 1];
218     } /* end - for ( i = 0; i < 256; i++) */
219     break;
220 } /* end - switch (mode) */
221
222 return(err);
223
224 } /* end - int ndaph0(char *cs) */

```

```

1 /*      ndapctq0()      Analysis Software      ndapctq0()
2 **
3 **      Programmer:      H. L. Kasdan
4 **
5 **      Copyright :      1988 by
6 **      International Remote Imaging Systems, Inc.
7 **
8 *****
9 **
10 **      NAME      ndapctq0 -- produce cum hist quantiles and statistics routine
11 **
12 **
13 **      SYNOPSIS
14 **      #include <stdio.h>
15 **      #include <string.h>
16 **      #include "cca.h"
17 **      #include "ndap.h"
18 **      int ndapctq0( struct chist0 *p_chist0, struct qt10 *p_qt10 )
19 **      struct chist0 *p_chist0;      pointer to chist0 data structure
20 **      struct qt10 *p_qt10;      pointer to qt10 data structure
21 **
22 **      DESCRIPTION
23 **      Routine generates quantiles of cell
24 **      image data using data from chist0 data structure.
25 **
26 **      SEE ALSO
27 **
28 **      DIAGNOSTICS
29 **
30 **      BUGS
31 **
32 **      REVISION HISTORY
33 **
34 *****

```

```

35 */
36 #include <math.h>
37 #include <stdio.h>
38 #include <string.h>
39 #include "cca.h"
40 #include "ndap.h"
41 #define BNPCT 95
42 #define BOFF 28
43 #define GNPCT 95
44 #define GOFF 21
45 #define FALSE 0
46 #define NVALUE 200
47 #define PMAXX 46
48 #define PMAXY 46
49 #define PMINX 2
50 #define PMINY 2
51 #define RNPCT 95
52 #define ROFF 25
53 #define TRUE 1
54
55 void ndapqnt1();
56 void ndapmsd1();
57
58 int ndapctq0( p_chisto, p_qt10 )
59 struct chisto *p_chisto, /* pointer to chisto data structure */
60 struct qt10 *p_qt10, /* pointer to qt10 data structure */
61 {
62
63 /*
64 ***** LOCAL STORAGE *****
65 *****
66 */
67 int err = 0; /* error return */
68 int i; /* generic index */
69 int n; /* number of interior pixels - length of index
70 array */
71
72 /*
73 ***** Compute Statistics for 9 images *****
74 *****
75 *****

```

/\* normalized maximum image value \*/

```

76 */
77 /*
78 **
79 */
80      r Cell Image
81      ndepqnt1(&p_chisto->crhisto[0], &p_qt10->qr0[0]),
82      ndapmsd1(&p_chisto->crhisto[0],
83              &p_qt10->mr0, &p_qt10->sdr0),
84
85      g Cell Image
86      ndepqnt1(&p_chisto->cghisto[0], &p_qt10->qg0[0]),
87      ndapmsd1(&p_chisto->cghisto[0],
88              &p_qt10->mg0, &p_qt10->sdg0),
89
90      b Cell Image
91      ndepqnt1(&p_chisto->cbhisto[0], &p_qt10->qb0[0]),
92      ndapmsd1(&p_chisto->cbhisto[0],
93              &p_qt10->mb0, &p_qt10->bdb0),
94
95      ri Cell Image
96      ndepqnt1(&p_chisto->crihisto[0], &p_qt10->qri0[0]),
97      ndapmsd1(&p_chisto->crihisto[0],
98              &p_qt10->mr10, &p_qt10->sdr10),
99
100      gi Cell Image
101      ndepqnt1(&p_chisto->cgihisto[0], &p_qt10->qgi0[0]),
102      ndapmsd1(&p_chisto->cgihisto[0],
103              &p_qt10->mg10, &p_qt10->sdg10),
104
105
106
107
108
109      ndepqnt1(&p_chisto->cgihisto[0], &p_qt10->qgi0[0]),
110      ndapmsd1(&p_chisto->cgihisto[0],
111              &p_qt10->mg10, &p_qt10->sdg10),
112
113 */

```

```

113 **                               bi Cell Image
114 */
115 ndapqnt1(&p_chisto->cbihisto[0], &p_qt10->qbio[0]);
116 ndapmsd1(&p_chisto->cbihisto[0],
117          &p_qt10->mbio, &p_qt10->sdbio);
118
119 /*
120 **                               re Cell Image
121 */
122 ndapqnt1(&p_chisto->crehisto[0], &p_qt10->qreo[0]);
123 ndapmsd1(&p_chisto->crehisto[0],
124          &p_qt10->mreo, &p_qt10->sdreo);
125
126 /*
127 **                               ge Cell Image
128 */
129 ndapqnt1(&p_chisto->cgehisto[0], &p_qt10->qgeo[0]);
130 ndapmsd1(&p_chisto->cgehisto[0],
131          &p_qt10->mgeo, &p_qt10->sdgeo);
132
133 /*
134 **                               be Cell Image
135 */
136 ndapqnt1(&p_chisto->cbehisto[0], &p_qt10->qbeo[0]);
137 ndapmsd1(&p_chisto->cbehisto[0],
138          &p_qt10->mbeo, &p_qt10->sdbeo);
139
140
141 return(err);
142
143 } /* end - int ndapctq(char *cs) */
144
145 /* ndapqnt1()      Analysis Software      ndapqnt1()
146 **
147 ** Programmer:    H. L. Kasdan
148 **

```

```

149 ** Copyright : 1988 by
150 ** International Remote Imaging Systems, Inc.
151 **
152 ** *****
153 **
154 ** NAME ndapqnt1 - compute quantiles from cumulative histogram
155 **
156 **
157 ** SYNOPSIS
158 ** #include <stdio.h>
159 ** #include <string.h>
160 ** #include "cca.h"
161 ** #include "ndap.h"
162 ** void ndapqnt1( unsigned int chist[256] int quant[50])
163 **                cumulative histogram array
164 **                array in which 50 quantiles
165 **                will be returned
166 **
167 ** int quant[50];
168 **
169 ** DESCRIPTION
170 ** Routine orders determines quantiles from cumulative histogram
171 ** array.
172 **
173 ** SEE ALSO
174 **
175 ** DIAGNOSTICS
176 **
177 ** BUGS
178 **
179 ** REVISION HISTORY
180 ** *****
181 void ndapqnt1( chist, quant)
182 unsigned int chist[256], /* cumulative histogram array */
183 int quant[50], /* array in which 50 quantiles
184 will be returned */
185

```



```

186 {
187 /*
188 ***** LOCAL STORAGE *****
189 **
190 *****
191 **
192 */
193 long int cJ; /* target cum hist value for next quantile */
194 int i;
195 int j0 = 0; /* quantile index */
196 long int l50 = 50; /* number of quantiles */
197 long int maxnum = chist[255]; /* total number of values in histgram */
198 int n = 256; /* size of histogram array */
199 /*
200 ** Generate Quantiles
201 **
202 cJ = (((long int)(j0 + 1)) * maxnum) / 150;
203 for (i = 1; (i <= n) && (j0 < 50); i++)
204 {
205 while (((unsigned int)cJ <= chist[i]) && (j0 < 50))
206 {
207 quant[j0++] = i;
208 cJ = (((long int)(j0 + 1)) * maxnum) / 150;
209 } /* end - while ((jnext - j0++) > 1) */
210
211 } /* end - for (i = 1; i <= n; i++) */
212 } /* end - void ndappqnt1( int n, int errinf[], int indx[]) */
213
214
215 /* ndapmsd1() Analysis Software ndapmsd1()
216 **
217 ** Programmer: H. L. Kasdan
218 **
219 ** Copyright : 1988 by
220 ** International Remote Imaging Systems, Inc.
221 **
222 *****
223 ** NAME
224 **

```



```

262 {
263 /*
264 *****
265 *****
266 ***** LOCAL STORAGE *****
267 *****
268 */
269 int chvalue; /* current histogram value */
270 double di; /* double index */
271 double dmean; /* running first moment */
272 int dopixel; /* previous temp pixel value */
273 double dpixel; /* temp pixel value */
274 double dnt = (double)chist[255]; /* total number of points in histogram */
275 double dsdev; /* double std dev */
276 double dsm; /* running second moment */
277 int i;
278 int n = 256; /* histogram array size */
279 /*
280 Compute Moments

```

```

281 */
282 dmean = 0.;
283 dsm = 0.;
284 dopixel = 0.;
285 for (i = 0; i < n; i++)
286 {
287     chvalue = (int)chist[i];
288     dpixel = (double)(chvalue - dopixel);
289     if (dpixel != 0.)
290     {
291         di = (double)i;
292         dmean += dpixel * di;
293         dsm += dpixel * di * di;
294     } /* end - if (dpixel != 0.) */
295     dopixel = chvalue;
296 } /* end - for (i = 0; i <= n) i++) */
297 dmean /= dnt;
298 #mean = (int)dmean;
299 dsm /= dnt;
300 dsdev = dsm - (dmean * dmean);
301 dsdev = (dsdev >= (double)0.) ? dsdev : (double)0.;
302 dsdev = sqrt(dsdev);
303 #sdev = (int)dsdev;
304 } /* end - void ndapmsd1( int n, int errin[], int indx[]) */
305

```

```

1 /*      ndapmlq0()      Analysis Software      ndapmlq0()
2 **
3 **      Programmer:      H. L. Kasdan
4 **
5 **      Copyright :      1988 by
6 **                      International Remote Imaging Systems, Inc.
7 **
8 *****
9 **
10 **      NAME      ndapmlq0 - produce log image quantiles and statistics routine
11 **
12 **
13 **      SYNOPSIS
14 **      #include <stdio.h>
15 **      #include <string.h>
16 **      #include "cca.h"
17 **      #include "ndap.h"
18 **      int ndapmlq0( struct image0 *p_image0 )
19 **      struct image0 *p_image0;
20 **
21 **      DESCRIPTION
22 **      Routine generates quantiles of log difference cell
23 **      image data using data from image0 data structure.
24 **      Modified heapsort algorithm is used to order pixel data.
25 **
26 **      SEE ALSO
27 **
28 **      DIAGNOSTICS
29 **
30 **      BUGS
31 **
32 **      REVISION HISTORY      modified for *.URN precursor to use window value
33 **      900507 hlk
34 **
35 *****

```

```

36 */
37 #include <math.h>
38 #include <stdio.h>
39 #include <string.h>
40 #include "cca.h"
41 #include "ndap.h"
42 #define BNPCT 95
43 #define BOFF 28
44 #define ONPCT 95
45 #define GOFF 21
46 #define FALSE 0
47 #define NVALUE 200
48 #define PMAXX 46
49 #define PMAXY 46
50 #define PMINX 2
51 #define PMINY 2
52 #define RNPCT 95
53 #define ROFF 25
54 #define TRUE 1
55
56 void indexx();
57 void ndapqnt0();
58 void ndapmsd0();
59
60 int ndapmlq0( p_image0 )
61 struct image0 *p_image0; /* pointer to image0 data structure */
62 {
63 /*
64 **
65 **
66 **
67 */
68 extern int ltab0[256]; /* precomputed log table values */
69
70 /*
71 **
72 **
73 **
74 */
75 int bnvalue; /* blue pixel normalizing value */

```

/\* normalized maximum image value \*/

GLOBAL STORAGE

LOCAL STORAGE

```

76 int err = 0;
77 int i;
78 int indx[CAXMAX * CAYMAX + 1];
79 int gnvalue;
80 long int lptmp;
81 int n;
82
83 int rnvalue;
84 int x;
85 int y;
86 /*
87 *****
88 **          Generate Index Array
89 *****
90 */
91 n = 0;
92 /* index count */
93 for ( y = 0; y < p_image0->window; y++ )
94 {
95   for ( x = 0; x < p_image0->window; x++ )
96   {
97     if ((p_image0->ccimg[y][x] == 'x') ||
98         ((p_image0->ccimg[y][x] >= '0') && (p_image0->ccimg[y][x] < '8'))))
99     {
100      indx[+n] = (y * CAXMAX) + x;
101      /* end - if (...) && (...) */
102    }
103   } /* end - for ( x = 0; x < p_image0->window; x++ ) */
104 } /* end - for ( y = 0; y < p_image0->window; y++ ) */
105 /*
106 **
107 */
108 indx[n, &p_image0->lgmlr[0][0], indx]; /* order index array for lgmlr */
109 ndapqnt0(n, &p_image0->lgmlr[0][0], indx, &p_image0->qclgmlr[0]);
110 ndapmsd0(n, &p_image0->lgmlr[0][0], indx,
111          &p_image0->mc1gmlr, &p_image0->sdclgmlr);
112

```

```

113 /*          lbmlg Cell Image
114 **
115 */
116 indexx(n, &p_image0->lbmlg[0][0], indx); /* order index array for lbmlg */
117 ndapqnt0(n, &p_image0->lbmlg[0][0], indx, &p_image0->qclbmlg[0]);
118 ndapmsd0(n, &p_image0->lbmlg[0][0], indx,
119          &p_image0->mc1bmlg, &p_image0->sdclbmlg);
120
121 /*          lrmlb Cell Image
122 **
123 */
124 indexx(n, &p_image0->lrmlb[0][0], indx); /* order index array for lrmlb */
125 ndapqnt0(n, &p_image0->lrmlb[0][0], indx, &p_image0->qclrmlb[0]);
126 ndapmsd0(n, &p_image0->lrmlb[0][0], indx,
127          &p_image0->mc1rmlb, &p_image0->sdclrmlb);
128
129
130 return(err);
131
132 } /* end - int ndapmlq(char *cs) */
133
134 /*          Analysis Software          indexx()
135 **
136 **          Programmer:      H. L. Kasdan
137 **
138 **          Copyright :      1988 by
139 **          International Remote Imaging Systems, Inc.
140 **
141 *****
142 **
143 **          NAME          indexx - order index array for log image data
144 **
145 **
146 **          SYNOPSIS
147 **          #include <stdio.h>
148 **          #include <string.h>
149 **          #include "cca.h"
150 **          #include "ndap.h"

```



```

151 ** void indexx( int n, int arrin[], int indx[])
152 ** int n;          number of elements to be ordered
153 ** int arrin[];   array containing elements to
154 **              be ordered
155 ** int indx[];    array containing indices of
156 **              elements in arrin that will
157 **              be ordered. This index array
158 **              will be ordered in place.
159 **

```

```

160 ** DESCRIPTION
161 ** Routine orders elements in array arrin specified in
162 ** index array, indx.
163 **

```

```

164 ** SEE ALSO

```

```

165 ** DIAGNOSTICS

```

```

166 ** BUGS

```

```

167 ** REVISION HISTORY

```

```

170 ** *****
171 ** *****
172 ** *****
173 ** /

```

```

174 void indexx( n, arrin, indx) /* number of elements to be ordered */
175 int n;
176 int arrin[]; /* array containing elements to
177              be ordered */
178 int indx[]; /* array containing indices of
179             elements in arrin that will
180             be ordered. This index array
181             will be ordered in place. */
182

```

```

183 {
184 /* *****
185 ** ***** LOCAL STORAGE *****
186 ** *****
187 *****

```

-48-

```

188 */
189 int i;
190 int indxt;
191 int ir;
192 int j;
193 int l;
194 int q;
195 /*
196 **
197 */
198 l = (n >> 1) + 1;
199 ir = n;
200
201 for (i)
202 {
203
204     if (l > 1)
205         q = arrin[(indxt = indxt[--l])];
206     else
207     {
208         q = arrin[(indxt = indxt[ir])];
209         indxt[ir] = indxt[l];
210         if (--ir == 1)
211         {
212             indxt[l] = indxt;
213             return;
214         } /* end - if (--ir == 1) */
215         } /* end - else (l > 1) */
216
217     i = 1;
218     j = l << 1;
219     while (j <= ir)
220     {
221         if (j < ir && arrin[indxt[j]] < arrin[indxt[j + 1]]) j++;
222         if (q < arrin[indxt[j]])
223         {
224

```

```

225     indx[i] = indx[j];
226     j += (i=j);
227   } /* end - if (q < arrin[indx[j]]) */
228   else j = ir + 1;
229 } /* end - while (j <= ir) */
230
231     indx[i] = indxt;
232
233   } /* end - for (i) */
234
235   } /* end - void indexx( int n, int arrin[], int indx[]) */
236
237 /* ndapqntO() Analysis Software ndapqntO()
238 **
239 ** Programmer: H. L. Kasden
240 **
241 ** Copyright : 1988 by
242 ** International Remote Imaging Systems, Inc.
243 **
244 *****
245 **
246 ** NAME ndapqntO - order index array for log image data
247 **
248 **
249 ** SYNOPSIS
250 ** #include <stdio.h>
251 ** #include <string.h>
252 ** #include "cca.h"
253 ** #include "ndap.h"
254 ** void ndapqntO( int n, int arrin[], int indx[], int quant[50])
255 ** int n;
256 ** int arrin[];
257 ** int indx[];
258 **
259 ** int quant[50];
260 **
261 **
262 **
263 **
264 **

```

number of elements to be ordered  
array containing elements to  
be ordered  
array containing indecies of  
elements in arrin that will  
be ordered. This index array  
will be ordered in place.  
array in which 50 quantiles  
will be returned

```

265 ** DESCRIPTION
266 ** Routine orders determines quantiles from elements in array
267 ** arrin according to index array, indx.
268 **
269 ** SEE ALSO
270 **
271 ** DIAGNOSTICS
272 **
273 ** BUGS
274 **
275 ** REVISION HISTORY
276 **
277 *****
278 #/
279 void ndepqnt0( n, arrin, indx, quant)
280 int n, /* number of elements to be ordered */
281 int arrin[], /* array containing elements to
282 be ordered */
283 int indx[], /* array containing indecies of
284 elements in arrin that will
285 be ordered. This index array
286 will be ordered in place. */
287 int quant[50], /* array in which 50 quantiles
288 will be returned */
289
290
291
292 {
293 /* ***** LOCAL STORAGE *****
294 ** *****
295 ** *****
296 #/
297 int i, /* quantile index */
298 int j0 = 0, /* target quantile index */
299 int jnext,
300 /*
301 **
302 #/

```

Generate Quantiles

```

303 quant[0] = errin[indx[i]];
304 j0 = i;
305 for (i = 1; (i <= n) && (j0 < 50); i++)
306 {
307     jnext = (i * 50) / n;
308     while (((jnext - j0) > 0) && (j0 < 50))
309     {
310         quant[j0++] = errin[indx[i]];
311     } /* end - while ((jnext - j0++) > 1) */
312 } /* end - for (i = 1; i <= n; i++) */
313 } /* end - void ndapqnt0( int n, int errin[], int indx[]) */
314
315
316 /* ndapmsd() Analysis Software ndapmsd()
317 **
318 ** Programmer: H. L. Kasdan
319 **
320 ** Copyright : 1988 by
321 ** International Remote Imaging Systems, Inc.
322 **
323 ** *****
324 ** *****
325 ** *****
326 ** *****
327 ** *****
328 ** *****
329 ** *****
330 ** *****
331 ** *****
332 ** *****
333 ** *****
334 ** *****
335 ** *****
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

```

337 **      int n;
338 **      int arrin[];
339 **
340 **      int indx[];
341 **
342 **
343 **      int #mean;
344 **
345 **      int #sdev;
346 **
347 **
348 **      DESCRIPTION
349 **      Routine orders computes mean and standard deviation of
350 **      elements in array arrin specified in index array, indx.
351 **
352 **
353 **      BEE ALSO
354 **
355 **      DIAGNOSTICS
356 **
357 **      BUQB
358 **
359 **      REVISION HISTORY
360 **      *****
361 **/
362 void ndepmsd( n, arrin, indx, mean, sdev)
363 int n;
364 int arrin[];
365 int indx[];
366
367
368
369
370 int #mean;
371
372 int #sdev;
373
374
375
376 {
377 /*
number of elements to be ordered
array containing elements to
be ordered
array containing indecies of
elements in arrin that will
be used.
mean value of specified
elements
standard deviation of
specified elements

Routine orders computes mean and standard deviation of
elements in array arrin specified in index array, indx.

BEE ALSO

DIAGNOSTICS

BUQB

REVISION HISTORY
*****
*/
void ndepmsd( n, arrin, indx, mean, sdev)
/* number of elements to be ordered */
/* array containing elements to
be ordered */
/* array containing indecies of
elements in arrin that will
be ordered. This index array
will be ordered in place. */
/* mean value of specified
elements */
/* standard deviation of
specified elements */

```

```

378 *****
379 ** LOCAL STORAGE *****
380 *****
381 **/
382 double dmean; /* running first moment */
383 double dpixel; /* temp pixel value */
384 double dsdev; /* double std dev */
385 double dsm; /* running second moment */
386 int i;
387 /*
388 ** Compute Moments
389 **/
390 dmean = 0.;
391 dsm = 0.;
392 for (i = 1; i <= n); i++)
393 {
394 dpixel = (double)arrin[indx[i]];
395 dmean += dpixel;
396 dsm += dpixel * dpixel;
397 } /* end - for (i = 1; i <= n); i++) */
398 dmean /= (double)n;
399 #mean = (int)dmean;
400 dsm /= (double)n;
401 dsdev = dsm - (dmean * dmean);
402 dsdev = (dsdev >= (double)0.) ? dsdev : (double)0.;
403 dsdev = sqrt(dsdev);
404 #sdev = (int)dsdev;
405 } /* end - void ndapmsd0( int n, int arrin[], int indx[]) */
406

```

**WHAT IS CLAIMED IS:**

1. A method of identifying an object under examination, said object has a discernible boundary; said method comprising the steps of:

5

- a) forming an image of said object;
- b) segmenting said image to form a plurality of pixels;

10

- c) detecting the boundary of said object under examination in said image;
- d) measuring a parameter of each pixel within the boundary detected;
- e) forming an ordered sequence of the parameter of the pixel measured, within the boundary detected;

15

$P_1 \dots P_k \dots P_N$

$P_1$  - first value of the parameter of the pixel within the boundary detected;

$P_N$  - last value of the parameter of the pixel within the boundary detected;

20

$N$  - total number of pixels within the boundary detected;

$k$  - index to the  $N$  pixels;

- f) determining the  $P_k$  data point(s) such that  $k$  satisfies the following relationship; and

25

$$\frac{i-1}{M} < \frac{k}{N} \leq \frac{i}{M}$$

$M$  - total number of quantile partitions;

$i$  -  $i^{\text{th}}$  fraction of partition or  $i^{\text{th}}$  quantile partitions;

30

- g) identifying said object under examination, based upon the relationship developed from step (f) associated with said object.

35

2. The method of Claim 1 wherein  $P_1$  is the smallest value and  $P_N$  is the largest value.



-55-

3. The method of Claim 1, wherein said identifying step further comprising:

5 comparing said relationship of step (f) associated with said object under examination, to a table of pre-determined relationships of other identified objects to identify the object under examination.

4. The method of Claim 3, wherein said table of pre-determined relationships of other identified objects comprises a range of values.

10 5. The method of Claim 3, wherein said table of predetermined relationships of other identified object comprises a single value.

15 6. The method of Claim 4, wherein said table of pre-determined relationships are based upon experimental results.

7. The method of Claim 6, wherein said table of predetermined relationships of other identified object comprises a single value.

20 8. The method of Claim 1, wherein said parameter is intensity of visible light.

9. The method of Claim 1, wherein said parameter is a function of color representations at the same position.

25 10. The method of Claim 9, wherein said color representation is based upon three primary colors.

-56-

11. The method of Claim 9, wherein said color representation is based upon hue, intensity, and saturation of colors.

5 12. The method of Claim 10, wherein said parameter is

$$\log (a) - \log (b)$$

where a and b are the intensities of two different colors.

10 13. The method of Claim 12 wherein a is red color and b is blue color.

14. The method of Claim 12, wherein a is blue color and b is green color.

15 15. The method of Claim 12, wherein a is green color and b is red color.

16. A method of differentiating a plurality of different objects under examination, wherein each object has a discernible boundary; said method comprising the steps of:

- 20 a) forming a plurality of images, with each image of each of said plurality of objects;
- b) segmenting each of said plurality of images to form a plurality of pixels;
- c) detecting the boundary of each object under examination in each of said images;
- 25 d) measuring a parameter of each pixel within the boundary of each object detected;
- e) forming an ordered sequence of the parameter of the pixel measured, within the boundary detected for each object;
- 30  $P_1 \dots P_{k1} \dots P_{N1}$  - object 1 having  $N_1$  pixels

$P_1 \dots P_{k_2} \dots P_{N_2}$  - object 2 having  $N_2$  pixels  
 f) determining, for each ordered sequence formed,  
 the  $P_k$  data point(s) such that  $k$  satisfies the  
 following; and

5 
$$\frac{i-1}{M} < \frac{k}{\text{Total No. of pixels for the object}} \leq \frac{i}{M}$$

10  $M$  - total number of quantile partitions;  
 $i$  -  $i^{\text{th}}$  fraction of partition or  $i^{\text{th}}$  quantile  
 partition;

g) comparing the  $P_k$  data point(s) of each ordered  
 sequence for each object to one another, for  
 identical variables of  $i$  and  $M$ , to  
 15 differentiate the objects.

17. The method of Claim 16, wherein said parameter  
 is intensity of visible light.

18. The method of Claim 16, wherein said parameter  
 is a function of color representations at the same  
 20 position.

19. The method of Claim 18, wherein said color  
 representation is based upon three primary colors.

20. The method of Claim 18, wherein said color  
 representation is based upon hue, intensity, and  
 25 saturation of colors.

21. The method of Claim 19, wherein said parameter  
 is

$$\log (a) - \log (b)$$

where  $a$  and  $b$  are the intensities of two different  
 30 colors.

-58-

22. The method of Claim 21 wherein a is red color and b is blue color.

23. The method of Claim 21, wherein a is blue color and b is green color.

5 24. The method of Claim 21, wherein a is green color and b is red color.

25. An apparatus for identifying an object under examination, said object has a discernible boundary; said apparatus comprising:

- 10 a) means for forming an image of said object;  
 b) means for segmenting said image to form a plurality of pixels;  
 c) means for detecting the boundary of said image of said object under examination;  
 15 d) means for measuring a parameter of each pixel within the boundary detected;  
 e) means for forming an ordered sequence of the parameter of the pixel measured, within the boundary detected;
- 20  $P_1 \dots P_k \dots P_N$   
 $P_1$  - first value of the parameter of the pixel within the boundary detected;  
 $P_N$  - last value of the parameter of the pixel within the boundary deducted;
- 25  $N$  - total number of pixels within the boundary detected;  
 $k$  - index to the  $N$  pixels;
- f) means for determining the  $P_k$  data point(s) such that  $k$  satisfies the following relationship;  
 30 and

$$\frac{i-1}{M} < \frac{k}{N} \leq \frac{i}{M}$$

-59-

M - total number of quantile partitions;  
i -  $i^{\text{th}}$  fraction of partition or  $i^{\text{th}}$  quantile  
partition;

5 g) means for identifying said object under  
examination, based upon the relationship of  $P_k$   
to  $i^{\text{th}}$  quantile in M partitions developed from  
step (f) associated with said object.

26. The apparatus of Claim 25 wherein  $P_1$  is the  
smallest value and  $P_N$  is the largest value.

10 27. The apparatus of Claim 25, wherein said means  
for identifying further comprising:

means for comparing said relationship of step  
(f) associated with said object under examination, to a  
table of pre-determined relationships of other identified  
15 objects to identify the object under examination.

28. The apparatus of Claim 27, wherein said table  
of pre-determined relationships of other identified  
objects comprises a range of values.

20 29. The apparatus of Claim 26, wherein said  
parameter is intensity of visible light.

30. The apparatus of Claim 29, wherein said  
parameter is a difference of color representations at the  
same position.

25 31. The apparatus of Claim 30, wherein said color  
representation is based upon three primary colors.

-60-

32. The apparatus of Claim 30, wherein said color representation is based upon hue, intensity, and saturation of colors.

5 33. The apparatus of Claim 31, wherein said parameter is

$$\log (a) - \log (b)$$

where a and b are the intensities of two different primary colors.

10 34. The apparatus of Claim 33, wherein a is red color and b is blue color.

35. The apparatus of Claim 33, wherein a is blue color and b is green color.

36. The apparatus of Claim 33, wherein a is green color and b is red color.

15 37. An apparatus for differentiating a plurality of objects under examination, each of said objects has a discernible boundary; said apparatus comprising:

- 20 a) means for forming an image of each of said plurality of objects to form a plurality of images;
- b) means for segmenting each of said plurality of images to form a plurality of pixels;
- 25 c) means for detecting the boundary of each of said plurality of objects under examination in each of said plurality of images;
- d) means for measuring a parameter of each pixel within the boundary of each object detected;
- 30 e) means for forming an ordered sequence of the parameter of the pixel measured, within the boundary detected for each object;

$P_1 \dots P_{k_1} \dots P_{N_1}$  - object 1 having  $N_1$  pixels

$P_1 \dots P_{k_2} \dots P_{N_2}$  - object 2 having  $N_2$  pixels

f) means for determining, for each ordered sequence formed, the  $P_k$  data point(s) such that k satisfies the following; and

5

$$\frac{i-1}{M} < \frac{k}{\text{Total No. of pixels for the object}} \leq \frac{i}{M}$$

10

M - total number of quantile partitions;  
 i -  $i^{\text{th}}$  fraction of partition or  $i^{\text{th}}$  quantile partition;

g) means for comparing the  $P_k$  data point(s) of each ordered sequence for each object to one another, for identical variables of i and M, to differentiate the objects.

15

38. The apparatus of Claim 37 wherein  $P_1$  is the smallest value and  $P_N$  is the largest value.

20

39. The apparatus of Claim 37, wherein said means for identifying further comprising:

means for comparing said relationship of step (f) associated with said object under examination, to a table of pre-determined relationships of other identified objects to identify the object under examination.

25

40. The apparatus of Claim 39, wherein said table of pre-determined relationships of other identified objects comprises a range of values.

41. The apparatus of Claim 38, wherein said parameter is intensity of visible light.

-62-

42. The apparatus of Claim 41, wherein said parameter is a difference of color representations at the same position.

5 43. The apparatus of Claim 42, wherein said color representation is based upon three primary colors.

44. The apparatus of Claim 42, wherein said color representation is based upon hue, intensity, and saturation of colors.

10 45. The apparatus of Claim 43, wherein said parameter is

$$\log (a) - \log (b)$$

where a and b are the intensities of two different primary colors.

15 46. The apparatus of Claim 45, wherein a is red color and b is blue color.

47. The apparatus of Claim 45, wherein a is blue color and b is green color.

48. The apparatus of Claim 45, wherein a is green color and b is red color.



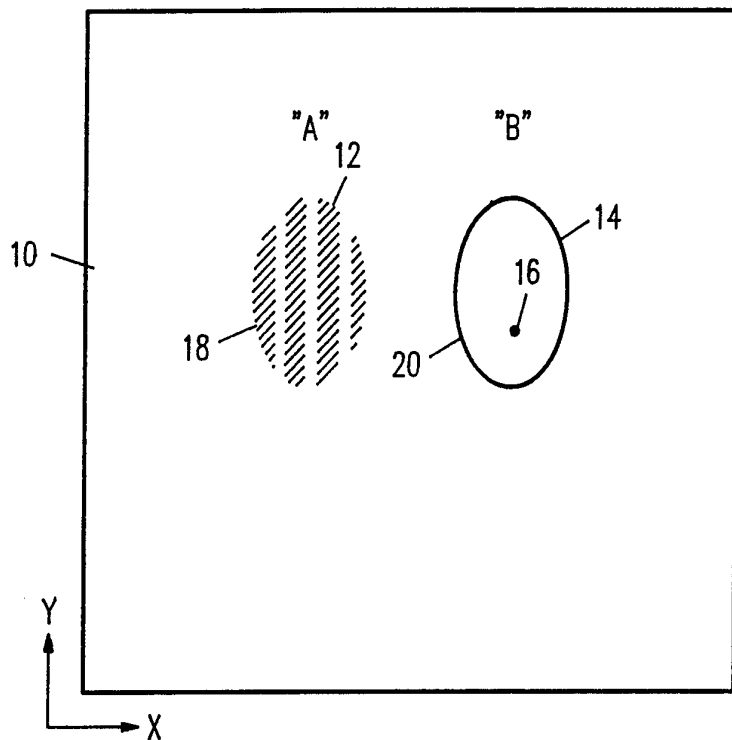


FIG. 1

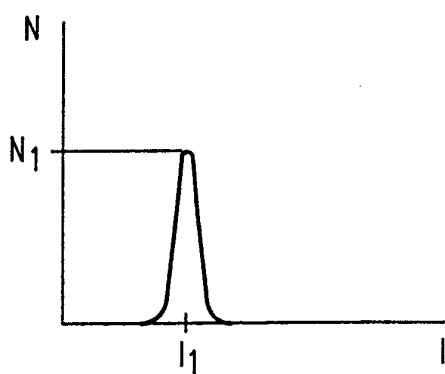


FIG. 2a

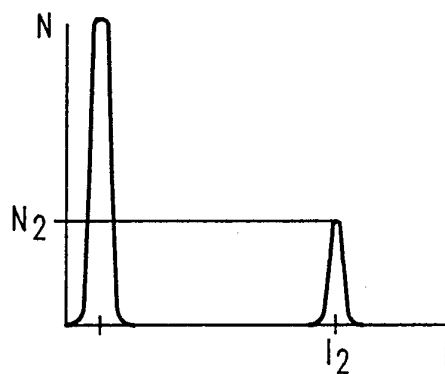


FIG. 2b

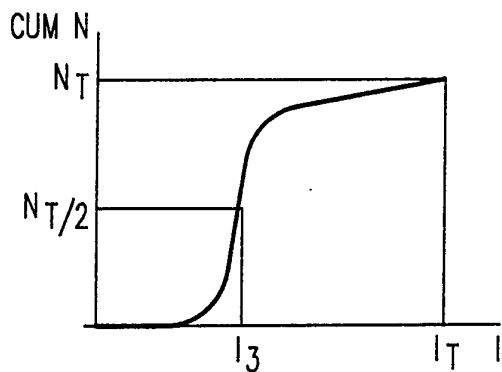


FIG. 3a

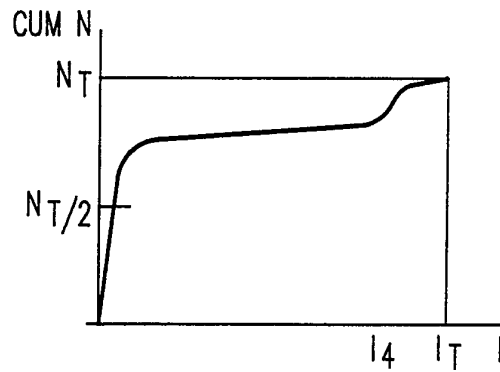


FIG. 3b

50th QUANTILE	INTENSITY	OBJECT
	$I_a$	xxx
	⋮	⋮
	$I_3$	12
	⋮	⋮
	$I_4$	14
	⋮	⋮
	$I_x$	xxx

FIG. 4

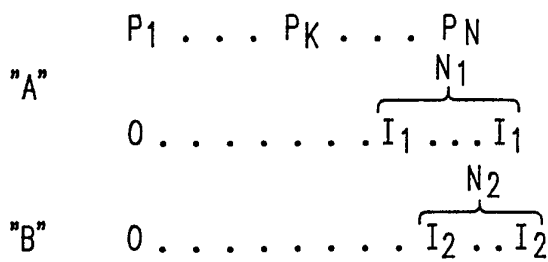


FIG. 5

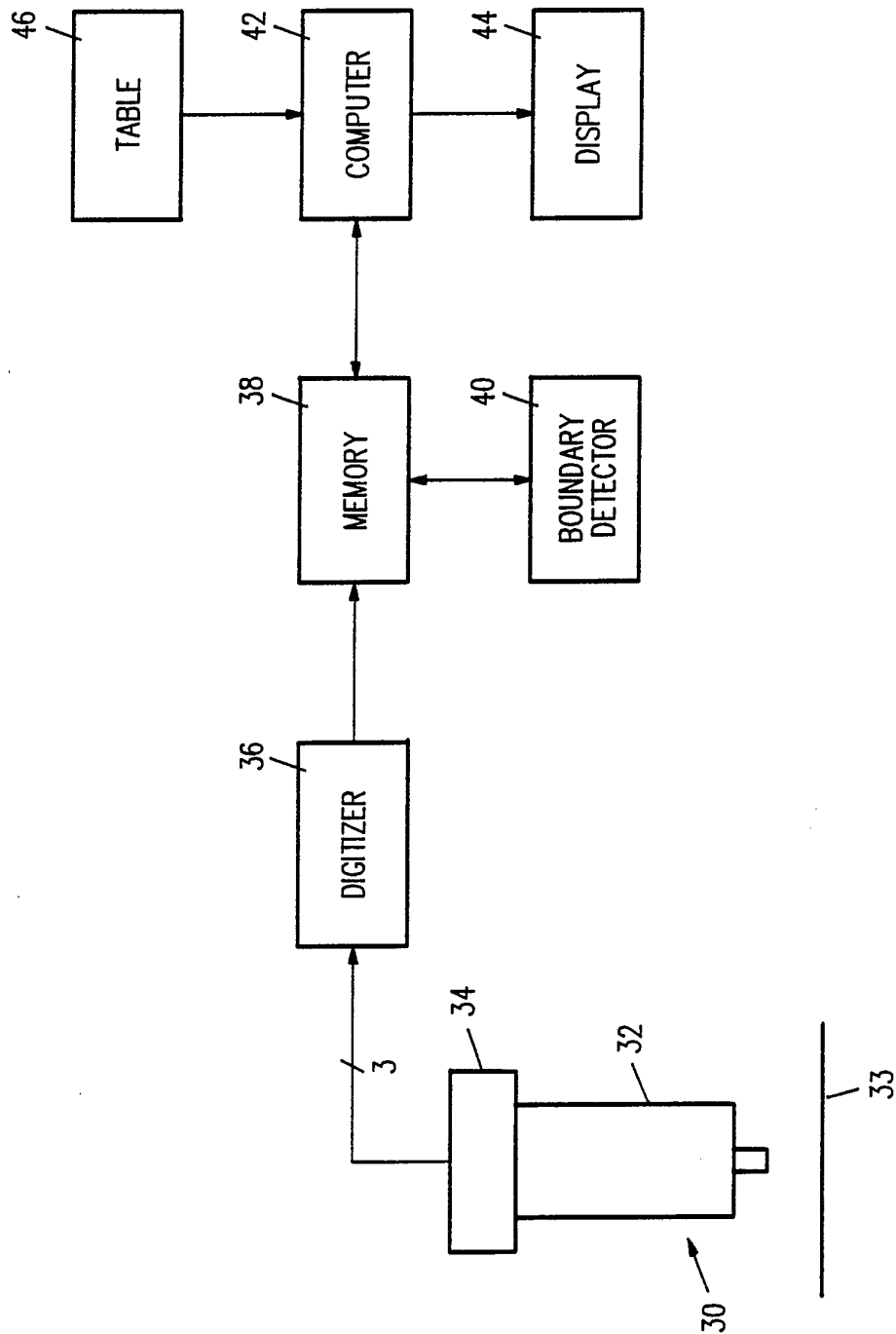


FIG. 6

## INTERNATIONAL SEARCH REPORT

International application No.

PCT/US93/09159

**A. CLASSIFICATION OF SUBJECT MATTER**

IPC(5) :G06K 09/48

US CL :382/22, 18

According to International Patent Classification (IPC) or to both national classification and IPC

**B. FIELDS SEARCHED**

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 382/22, 18, 6, 17, 21, 54

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched  
CHECKED REFERENCES CITED IN US PATENT 5,123,055Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)  
Please See Extra Sheet.**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO, A, WO90/14639 (KASDAN ET AL.) 29 NOVEMBER 1990, whole document.	1-48
A	US, A, 4,075,604 (DEGASPERI) 21 FEBRUARY 1978, see fig. 3 and col. 1, line 63 - col. 2, line 13; col. 4, lines 12-31.	1, 16, 25, 37
A	US, A, 5,121,338 (LODDER) 09 JUNE 1992, see Fig. 2 and col. 6 - col 8, line 25; col. 5, lines 8-29.	1-9, 16, 18, 25-30, 37-42
A	US, A, 4,573,197 (CRIMMINS) 25 FEBRUARY 1986, see Abstract, Figs. 1-3.	1, 16, 25, 37
A	US, A, 3,705,383 (FRAYER) 05 DECEMBER 1972, see col. 20, lines 41-56.	8-10, 17-19, 29-31, 41-43

 Further documents are listed in the continuation of Box C.
  See patent family annex.

* Special categories of cited documents:	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A" document defining the general state of the art which is not considered to be part of particular relevance	"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E" earlier document published on or after the international filing date	"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&" document member of the same patent family
"O" document referring to an oral disclosure, use, exhibition or other means	
"P" document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search

08 November 1993

Date of mailing of the international search report

FEB 07 1994

Name and mailing address of the ISA/US  
Commissioner of Patents and Trademarks  
Box PCT  
Washington, D.C. 20231Authorized officer *Dave Moor*  
LARRY J. PRIKOCKIS

Facsimile No. NOT APPLICABLE

Telephone No. (703) 305-4791

## INTERNATIONAL SEARCH REPORT

International application No.  
PCT/US93/09159

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US, A, 4,782,389 (MAYWEATHER III) 01 NOVEMBER 1988, see Abstract, Fig. 1.	1, 16, 25, 37