ISE-TR-82-27

# FALSE CONTOUR REMOVAL BY RANDOM BLURRING

by

Seiichi Nishihara

Katsuo Ikeda

April 5,1982

INSTITUTE
OF
INFORMATION SCIENCES AND ELECTRONICS

UNIVERSITY OF TSUKUBA

ISE-TR-82-27

# FALSE CONTOUR REMOVAL BY RANDOM BLURRING

Seiichi Nishihara
Katsuo Ikeda

April 5, 1982

Institute of Information Sciences and Electronics
University of Tsukuba
Sakura-mura, Niihari-gun, Ibaraki 305 Japan

# I. INTRODUCTION

When an area containing a gradual variation in brightness is displayed on a digital image display device, we often notice a spurious discontinuity of intensity across the border between two regions whose gray levels differ by only one unit. In Fig. 1, for example, we observe a set of contour lines resembling shell zones. Those discontinuities, however, are not characteristic of the object portrayed; they are intrinsic characteristics of digital image displays because a display device can plot only a limited number of discrete gray levels on the screen. Unfortunately, the human eye is very sensitive to this difference in brightness, so a person will easily perceive and be annoyed by the false contours produced. Those spurious discontinuities are somewhat amplified as the signal-to-noise ratio or the smoothness of border lines in the displayed image is increased. This means, paradoxically, that improvement in the accuracy of each point datum will make those spurious discontinuities even more noticeable. Thus, although no step in image processing should degrade image quality, the final presentation of a result in visual form is often improved by blurring so that those undesirable effects diminish. A method to break up the regular steps of digital coding has been proposed in the context of PCM transmission techniques[1]; this method uses pseudo-random generators to add noise to the signal before it is quantized and to subtract the same noise after reconversion to an analog signal.

********** Fig. 1 **********

This note presents a method which makes use of local picture patterns, in contrast to the previous method, which adds noise uniformly throughout the picture. The procedure developed here essentially performs random blurring near each border where there are spurious discontinuities in apparent brightness, so that it preserves edge sharpness and can be applied directly to digitized pictures without the need for AD/DA conversion processes.

## II. ELIMINATION OF FALSE CONTOUR LINES

Spurious contour lines, which make a picture unacceptable, tend to be intensified where the following conditions hold concerning a border:

1) The shape of the border is simple and smooth, that is, the border line is not jagged,
2) Little noise or randomness in brightness is present near

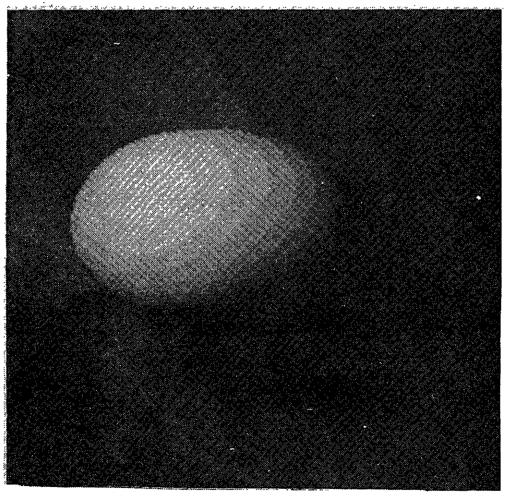


Fig. 1. Spurious contour lines like shell zones are evident in the image.

the border,

3) The border itself is long, that is, the number of border points is large.

A border is a series of points separating two regions whose gray levels differ from each other by just one unit. Hereafter, we use the term 'border' in this sense.

It is hard to get rid of the effect of condition 3), since this condition is related to the extent, or size, of each region. Our method makes the shape of the border jagged and blurred by adding enough random noise or scrambling points near borders so that conditions 1) and 2) do not hold. Thus continuous halftone is achieved without any drift of total brightness because of the eye's ability to average out noise.

## The Method

For each point in a picture, apply the following sequence of operations. First, determine whether that point is an element of a border (A. Border Examination Procedure). If the point is proved to be on a border line, then place there a probing bar mask normal to the border (see Fig. 2). A bar mask is a linear series of a fixed number of points, along which a check is made to determine whether randomness is present (B. Randomness Check Procedure). If the gray levels of all points in each half of a bar mask are found to be identical and equal to that of corresponding region, it is determined that there is no randomness, and a random blurring operation is invoked (C. Random Blurring Procedure).

********** Fig. 2 **********

Before giving a precise description of each procedure, we define some terminology. Let f(i,j) be an integer function with range [0,M-1], representing the gray level of a picture at point (i,j), where M is the number of gray levels. The four points directly adjacent, horizontally and vertically, to a given point (i,j) are called 4-neighbors of (i,j); and 8-neighbors of (i,j) are defined as these 4-neighbors together with the four diagonal neighbors[2]. A point (i,j) is called positive-isolated when the gray level of each 4-neighbor is smaller than that of (i,j) just by one, i.e. f(i,j)-1. Conversely, a negative-isolated point (i,j) is a point whose 4-neighbors all have the identical gray level f(i,j)+1.

A. Border Examination Procedure

For each point (i,j), seek and count the number N ($0 \leq N \leq 4$) of 4-neighbors having the gray level f(i,j)-1. If N=4, then (i,j) is a positive-isolated point which is not a

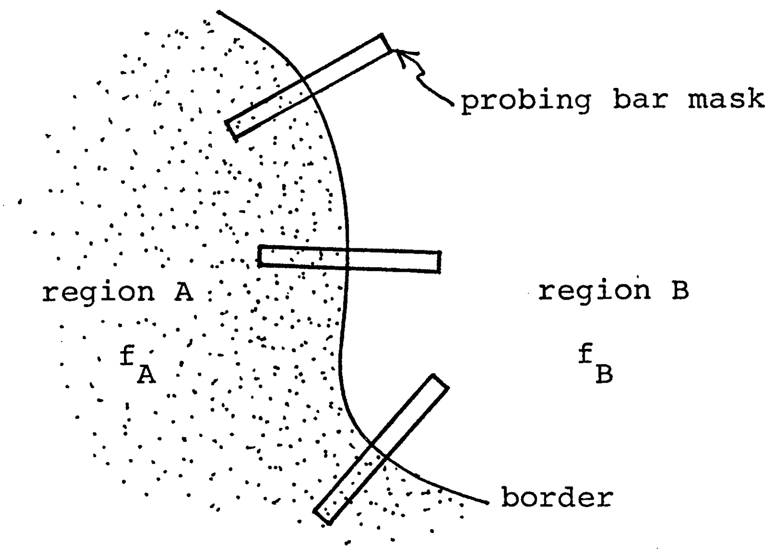



Fig. 2. Probing bar masks are placed so that the border crosses their midpoint at a right angle; $f_A$, $f_B$ are gray levels of region A and B, respectively, where $|f_A - f_B| = 1$.

border point. If no such 4-neighbors exist, (i,j) is an interior point of a region whose density value is f(i,j), or is a negative-isolated point, or is a point which is close to an edge where there is an abrupt change greater than one unit in brightness. In any case, this (i,j) can not be a border point.

If N=1,2 or 3 and at least one of those N points is not negative-isolated and other (4-N) 4-neighbors have the gray level f(i,j), then the point (i,j) is concluded to be lying on a border. All such cases where a point is determined to be a border point are shown in Fig. 3.

********** Fig. 3 **********

A point that passes this examination represents a candidate position where spurious discontinuities of brightness may possibly occur. Such a point is further tested by the following procedure.

B. Randomness Check Procedure

When a border point (i,j) has been identified by the above procedure, each of its 8-neighbors is checked in turn to determine whether or not it is also a border point with the same brightness f(i,j). This 8-neighbor checking procedure would be performed quickly by using the flag bit

```
  L          L          L          L          n
H H H  ,   H H L  ,   H H H  ,   H H n  ,   H H L  ,
  H          H          L          H          H

  L          L          L          n          L
H H H  ,   L H L  ,   L H n  ,   L H L  ,   n H L  ,
  n          H          H          H          H

  L          n          n
n H n  ,   L H n  ,   n H L
  H          H          H
```

Fig. 3. The complete list of cases of those relative disposition patterns (omitting the rotated variations) in which a point is recognized locally to be a border point. The point being examined is the central one expressed by H in each case. Among the 4-neighbors, the gray levels of H and L are f(H) and f(H)-1 respectively, where f(H) is the gray level of the central point. The points expressed by n have a gray level equal to that of L, and, further, they are also negative-isolated points.

which is reserved for each point; this flag is set earlier if the point is determined to be a border point by the Border Examination Procedure. If such an 8-neighbor is found, a bar mask of an arbitrarily specified length 2xw centered at the point (i,j), is placed normal to the direction of the border determined by that 8-neighbor and the point (i,j). All possible bar mask patterns are shown in Fig. 4 in relation to the positions of those two points. When there are two possible candidates, the appropriate bar mask pattern is selected by examining which part of the bar mask divided by the border point concerned belongs to the interior of the corresponding area. Since those bar masks shown in Fig. 4 are checked for every pair of adjacent points, each vertical or horizontal bar mask is checked twice. However, cases 3 to 6 of Fig. 4 show that two or more points lying in a diagonal direction will produce a series of bar masks that are adjacent but do not overlap, and this is preferable for a more natural picture.

********** Fig. 4 **********

The bar mask is then divided into two parts, each of length w. A randomness check is performed by investigating if all the consecutive w points of one part have the same gray level, f(i,j), while those of the other part have level

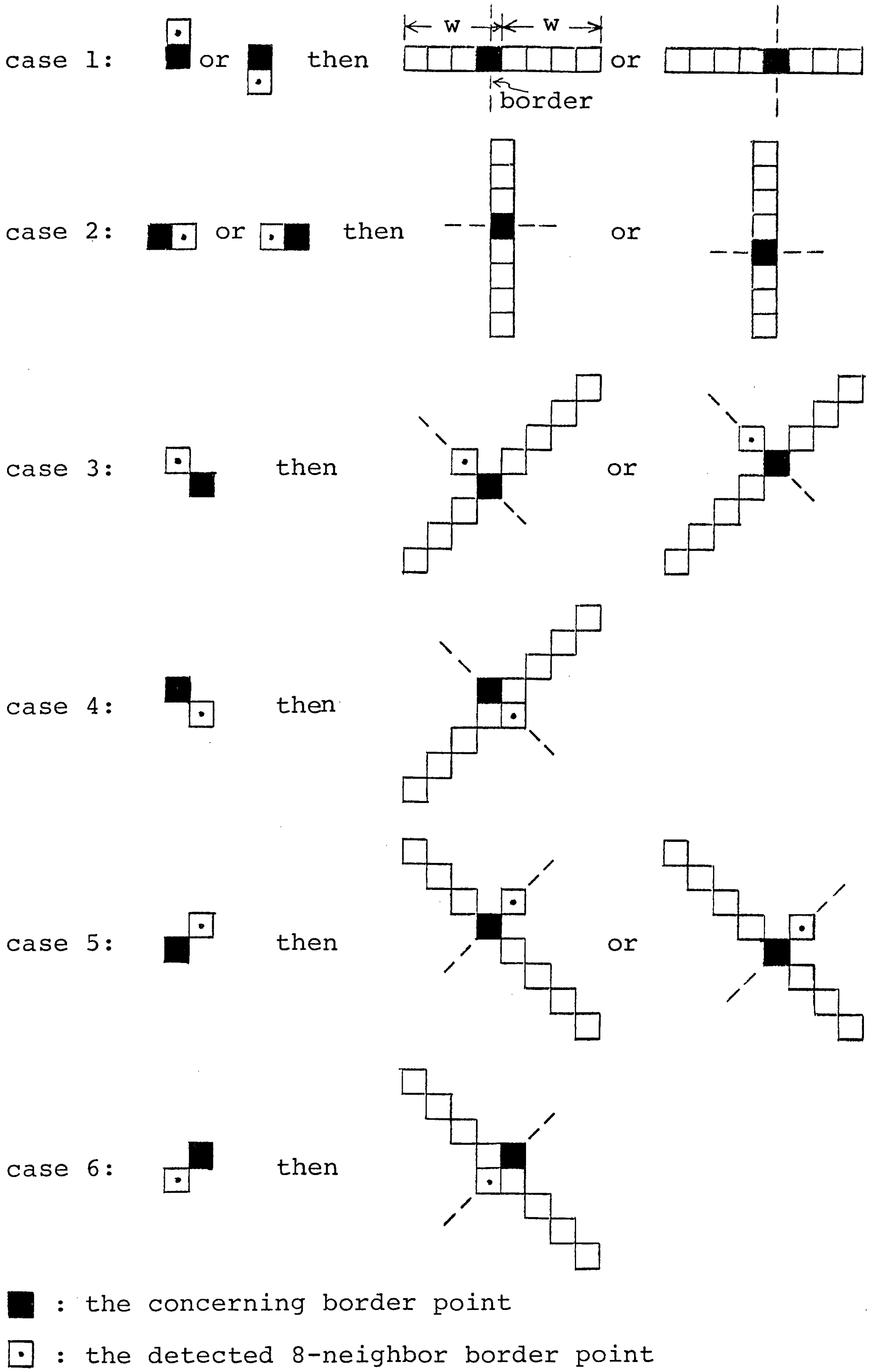



Fig. 4. Bar mask patterns defined according to the relative positions of a border point and its 8-neighbor border point, for w=4. A dashed line shows where the border is passing through.

f(i,j)-1. This checking criterion not only checks uniformness, but also establishes whether there is an edge in the neighborhood. If each of the bar mask is proved uniform, the next procedure, random blurring, is applied.

C. Random Blurring Procedure

This procedure ultimately achieves scrambling an adequate number of points on a given bar mask. Essentially, it produces a probability of changing the gray level of each point to that of the opposite side of the bar, which is zero at the end of the bar and rises linearly to 0.5 at the center. As shown in Fig. 5 of the w=4 case, the actual probability is a step function, computed by using uniformly distributed random numbers. This linearly varying probability causes a gradual average gray-value shift along the bar, similar to halftone reproduction, although in halftone the variation is in dot size, not dot quantity.

********** Fig. 5 **********

## III. EXPERIMENTAL RESULTS

Fig. 1 shows a picture of an egg on a flat table displayed on a digital image display whose resolution is

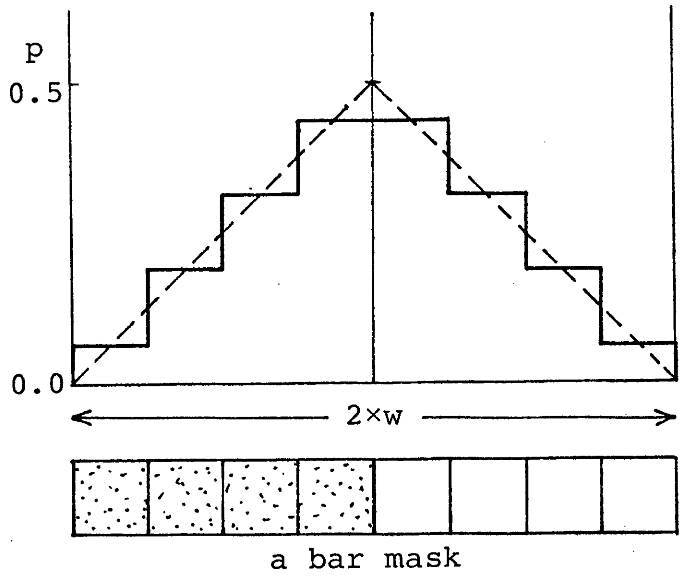
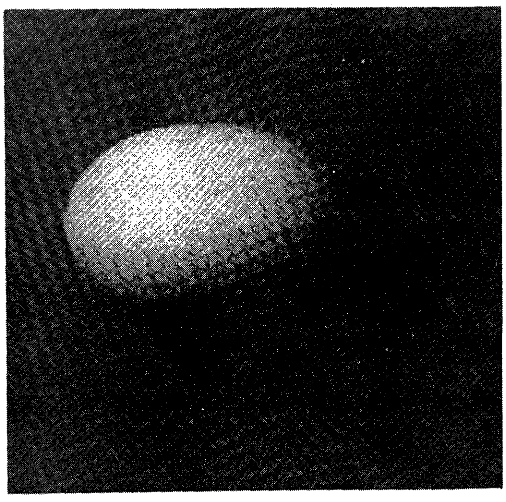



Fig. 5. The probability of changing the gray level of each point of a bar mask in the case w=4. The step function shows the actual probability produced, while the ideal probability is expresed by the broken line.

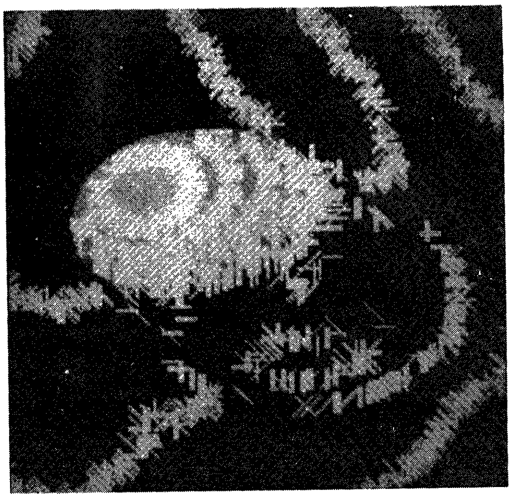
180X180 points. Each point has a 4-bit density value; that is, the number M of gray levels is 16. The spurious discontinuities of brightness, false contours, described before are clearly perceived. This picture was fed to the series of procedures described above. Fig. 6(a) shows the result for the case w=4. In Fig. 6(b), the bar masks in which random blurring has been performed are shown. Note that random blurring operations were performed only on borders where spurious discontinuities occur, and not on true edges.

********** Fig. 6(a)-(b) **********

Fig. 7 shows another example. The resolution of grid points is 230×230, and again M=16. The result of random blurring with w=3 performed on the original picture Fig. 7(a) is shown in Fig. 7(b). Random blurring is a kind of noise addition; in a sense, it is an inverse operation of smoothing. To show this characteristic, edge preserving smoothing by Nagao and Matsuyama[3] was applied to the original picture Fig. 7(a), yielding Fig. 7(d), in which false contours are of course more evident. Although getting smoother pictures by noise removal is one of the important preprocessing operations for image analysis, such as edge detection, it sometimes gives images with an unnatural



(a)



(b)

Fig. 6. (a) Result of random blurring of Fig. 1. (b) Bar masks in which random blurring operations have been performed.

appearance. Applying our method to Fig. 7(d), we get Fig. 7(e), which is very similar in appearance to Fig. 7(b). The bars in which random blurring was actually carried out are shown explicitly in Fig. 7(c) and (f) for the cases of Fig. 7(b) and (d) respectively.

********** Fig. 7(a)-(f) **********

## IV. CONCLUSION

We have proposed a method to eliminate spurious brightness discontinuities appearing in areas in which the brightness gradually varies. It uses the technique of local random blurring to get a gradual average gray-scale shift, and this technique is applied to every digitized picture. The method also preserves edge sharpness, which is not maintained by the simple dithering method that adds a random number before quantization or simply adds random noise. The present method, in a sense, is an inverse of smoothing such as edge preserving smoothing[3].

Even when bars overlap each other, random blurring should be performed additively by examining the original picture every time. Since the procedure is repeated locally by scanning points sequentially, as in the sequence of TV

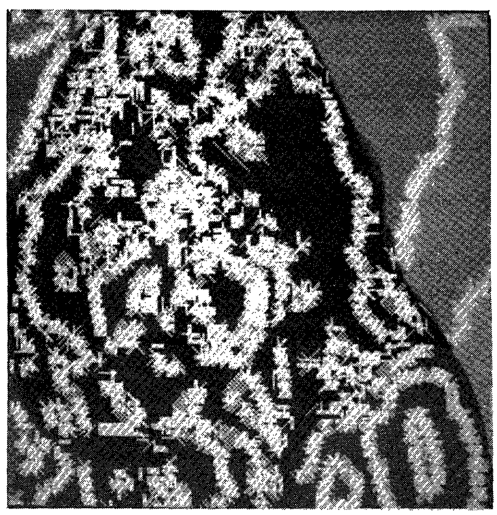
(a)

(d)

(b)

(e)

(c)



(f)

Fig. 7. (a) Original picture. (b),(c) Results of random blurring of (a) with w=3. Bar masks are shown explicitly in (c). (d) Result of edge preserving smoothing[3] of (a). (e),(f) Results of random blurring of (d) with w=3. Bar masks are shown explicitly in (f).

scanning, the original picture should be kept in a buffer memory. When bars whose corresponding borders have different gray levels overlap each other, the point value after several changes becomes the final gray level. This is the only case in which the picture produced is affected by the order in which points are inspected, and this is a very rare case in practice. The method is basically a one-pass procedure, so that it is performed in a length of time proportional to the number of points in the picture.

## ACKNOWLEDGEMENT

The authors would like to thank Professor J. W. Higgins of the Institute of Information Sciences and Electronics of the University of Tsukuba for his valuable suggestions.

## REFERENCES

[1] L. G. Roberts, Picture coding using pseudo-random noise, IRE Trans. Inform. Theory 8, 1962, 145-154.

[2] A. Rosenfeld and A. C. Kak, Digital Picture Processing, Academic Press, New York, 1976.

[3] M. Nagao and T. Matsuyama, Edge preserving smoothing, Computer Graphics and Image Processing, 9,1979, 394-407.

```
$BATCH
C   ELIMINATING MACH-EFFECT.
C   S.NISHIHARA,   JUNE 24,1981.
C   REVISED  JULY, 17, 1981,   ORTHOGONAL MIGRATION IS MODIFIED.
C   REVISED JULY, 14, 1981, TWO KINDS OF SONDE WIDTH IS AVAILABLE.
C   REVISED ON  JUNE 30,1981,   INCLUDED EGPR-ROUTINE (OF SPIDER).
C   SYS:MACHW.FTN
      INTEGER RD,WT,IPBLK(10),HIST(64),SU,SD,SR,SL
      INTEGER*2 IP,JP,CH,KP
      COMMON IP(250,236),JP(250,236),CH(20),KP(250,236)
      DATA RD,WT/X'58',X'38'/
      JJ=2*250*236
      CALL CDMODE(11,0,0,IST)
      CALL CDERS(11,6,0,IST)
      WRITE(5,1000)
 1000 FORMAT(' *SET DVP.....')
C   ISW=0:RDAREA,   1:DISP-IP,   2:DISP-KP,   3:EXCHANGE IP,KP,
C         4:MACH-PREPROCESS,   5:RD-MT1(OR REWIND),  6:WT-MT1(NOT OVER-WT),
C         7:EGPR(IP)-TO-KP,    8:REFINE-IP.
  122 CALL RDKEY(11,23,0,CH,1,1,ICOL,Y'OD',IST)
      DECODE(CH,1003) ISW
 1003 FORMAT(I1)
      IF(ISW .EQ. 0) GOTO 901
      IF(ISW .EQ. 9) GOTO 2000
      IF(ISW.LT.0 .OR. ISW.GT.8) GOTO 122
      GOTO (902,903,904,120,905,906,907,908),ISW
CCC   READ FROM DVP...
  901 CALL RDAREA(10,IP,10,8,509,479,1,2,IST)
      DO 10 M=1,64
      HIST(M)=0
   10 CONTINUE
      DO 20 M=1,236
      DO 30 N=1,250
      IW=63-IP(N,M)/256+1
      IP(N,M)=IW
C     IF(IW.LT.1 .OR. IW.GT.64) PAUSE 'RANGE ERR....'
      HIST(IW)=HIST(IW)+1
   30 CONTINUE
   20 CONTINUE
C
      ISW=0
      DO 40 IW=1,64
      IF(HIST(IW) .EQ.0) GOTO 40
      IE=IW
      IF(ISW .NE. 0) GOTO 40
      IB=IW
      ISW=1
   40 CONTINUE
C     WRITE(5,1002) IB,IE
C1002 FORMAT(' **',2I5)
      IW=IE-IB
C
      DO 50 M=1,236
      DO 60 N=1,250
      JW=(IP(N,M)-IB)*15/IW
      IP(N,M)=(JW*16+JW)*16
C     BLUE AND GREEN
   60 CONTINUE
   50 CONTINUE
      CALL CDERS(11,6,0,IST)
      CALL WAIT(300,1,IST)
  121 CONTINUE
      CALL WRTIMG(11,33,0,282,235,IP,16,Y'1FFF',0,IST)
      GOTO 122
CCC
```

```
CCC
CCC MACH-BAND-EFFECT-ELIMINATION BEGINS---------*********-----*****
  120 CONTINUE
C EDGE DETECTION...
      DO 100 M=3,234
      DO 110 N=3,248
      IV=IP(N,M)
      IU=IP(N,M-1)
      ID=IP(N,M+1)
      IR=IP(N+1,M)
      IL=IP(N-1,M)
      IW=IV-Y'110'
      NU=0
      NL=0
      NR=0
      ND=0
CCC
      SU=0
      IF(IU .NE. IW) GOTO 130
      SU=1
      CALL NEGA(IU,N,M-1,NU)
      SU=SU-NU
  130 SL=0
      IF(IL .NE. IW) GOTO 131
      SL=1
      CALL NEGA(IL,N-1,M,NL)
      SL=SL-NL
  131 SR=0
      IF(IR .NE. IW) GOTO 132
      SR=1
      CALL NEGA(IR,N+1,M,NR)
      SR=SR-NR
  132 SD=0
      IF(ID .NE. IW) GOTO 133
      SD=1
      CALL NEGA(ID,N,M+1,ND)
      SD=SD-ND
CCC
  133 ISUM=SU+SL+SR+SD
      NSUM=NU+NL+NR+ND
      IF(ISUM+NSUM .EQ. 4) GOTO 134
      IF(ISUM .EQ. 0) GOTO 135
C     IP(N,M) IS A BORDER(RED)...
      JP(N,M)=IV/256
      GOTO 136
  134 CONTINUE
C     IP(N,M) IS ISOLATED(GREEN)...
      JP(N,M)=(IV/256)*16
  136 CALL WRTIMG(11,N+32,M-1,N+32,M-1,JP(N,M),16,Y'1FFF',0,IST)
      GOTO 110
  135 CONTINUE
C     IP(N,M) IS NEITHER A BORDER NOR ISOLATED...
      JP(N,M)=0
  110 CONTINUE
  100 CONTINUE
C REVISED...   TWO KINDS OF WIDTH(KURI1,KURI2) CAN BE TESTED.
C              IF ONE WIDTH IS ZERO, THEN IS SKIPPED.
      CALL RDKEY(11,23,35,CH,5,1,ICOL,Y'0D',IST)
      DECODE(CH,1004) ISW,KURI1,KURI2
 1004 FORMAT(I1,2I2)
      IF(KURI1 .GE. KURI2) GOTO 109
      IWW=KURI1
      KURI1=KURI2
      KURI2=IWW
  109 CONTINUE
      IF(ISW .NE. 0) GOTO 300
```

```
      GOTO 122
CCCCCCLCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCC   MACH ELIMINATION PROCEDURE.............................
CC
  300 DO 301 M=1,236
      DO 302 N=1,250
      IPW=IP(N,M)
      IF(IPW .LT. Y'1000') GOTO 305
      IPW=IPW-Y'1000'
      IP(N,M)=IPW
  305 KP(N,M)=IPW
  302 CONTINUE
  301 CONTINUE
C     CALL WRTIMG(11,33,0,282,235,KP,16,Y'1FFF',0,IST)
      IRAN=30031
C
      DO 303 M=3,234
      DO 304 N=3,248
      IW=IP(N,M)
      JW=JP(N,M)
      IF(JW.GT.0 .AND. JW.LT.Y'0010') GOTO 310
      GOTO 304
C     BORDER FOUND...  THEN CHECK NEIGHBORHOODS...
CCC   VERTICAL-BORDER CHECK**********
C...  *  ...
C...(N,M)...
C...  *  ...
  310 CALL BDRCHK(JW,N,M-1,ICODE)
      IF(ICODE .EQ. 1) GOTO 311
      CALL BDRCHK(JW,N,M+1,ICODE)
      IF(ICODE .EQ. 1) GOTO 311
      GOTO 320
  311 CONTINUE
      ISW=0
      KURI=KURI1
  314 CALL HOMCHK(IW,N-1,M,1,KURI,KH)
      CALL HOMCHK(IW,N+1,M,2,KURI,LH)
      IF(KH.EQ.0 .AND. LH.EQ.-1) GOTO 312
      IF(KH.EQ.-1 .AND. LH.EQ.0) GOTO 313
      IF(KURI2.EQ.0 .OR. ISW.EQ.1) GOTO 320
      ISW=1
      KURI=KURI2
      GOTO 314
  312 CALL SCRMBL(N,M,1,KURI,IRAN)
      GOTO 320
  313 CALL SCRMBL(N-1,M,1,KURI,IRAN)
CCC   HORIZONTAL-BORDER CHECK**********
C.............
C...*(N,M)*...
C.............
  320 CALL BDRCHK(JW,N-1,M,ICODE)
      IF(ICODE .EQ. 1) GOTO 321
      CALL BDRCHK(JW,N+1,M,ICODE)
      IF(ICODE .EQ. 1) GOTO 321
      GOTO 330
  321 CONTINUE
      ISW=0
      KURI=KURI1
  324 CALL HOMCHK(IW,N,M-1,3,KURI,KH)
      CALL HOMCHK(IW,N,M+1,4,KURI,LH)
      IF(KH.EQ.0 .AND. LH.EQ.-1) GOTO 322
      IF(KH.EQ.-1 .AND. LH.EQ.0) GOTO 323
      IF(KURI2.EQ.0 .OR. ISW.EQ.1) GOTO 330
      KURI=KURI2
      ISW=1
      GOTO 324
```

```
  322 CALL SCRMBL(N,M,2,KURI,IRAN)
      GOTO 330
  323 CALL SCRMBL(N,M-1,2,KURI,IRAN)
CCC   LEFT-UP-BORDER CHECK**********
C...*.........
C....(N,M)....
C.........*...
  330 CALL BDRCHK(JW,N-1,M-1,ICODE)
      IF(ICODE .EQ. 1) GOTO 331
      CALL BDRCHK(JW,N+1,M+1,ICODE)
      IF(ICODE .EQ. 1) GOTO 2331
      GOTO 340
  331 CONTINUE
      ISW=0
      KURI=KURI1
  334 CALL HOMCHK(IW,N-1,M+1,5,KURI,KH)
      CALL HOMCHK(IW,N+1,M-1,6,KURI,LH)
      IF(KH.EQ.0 .AND. LH.EQ.-1) GOTO 332
      IF(KH.EQ.-1 .AND. LH.EQ.0) GOTO 333
      IF(KURI2.EQ.0 .OR. ISW.EQ.1) GOTO 340
      KURI=KURI2
      ISW=1
      GOTO 334
  332 CALL SCRMBL(N,M,3,KURI,IRAN)
      GOTO 340
  333 CALL SCRMBL(N-1,M+1,3,KURI,IRAN)
      GOTO 340
C
 2331 CONTINUE
      ISW=0
      KURI=KURI1
 2334 CALL ORTHOM(IW,N,M+1,5,KURI,KH)
      CALL ORTHOM(IW,N+1,M,6,KURI,LH)
      IF(KH.EQ.0 .AND. LH.EQ.-1) GOTO 2332
      IF(KH.EQ.-1 .AND. LH.EQ.0) GOTO 2332
      IF(KURI2.EQ.0 .OR. ISW.EQ.1) GOTO 340
      KURI=KURI2
      ISW=1
      GOTO 2334
 2332 CALL SCRMBL(N,M+1,3,KURI,IRAN)
      GOTO 340
CCC   LEFT-DOWN-BORDER CHECK**********
C.........*...
C....(N,M)....
C...*.........
  340 CALL BDRCHK(JW,N-1,M+1,ICODE)
      IF(ICODE .EQ. 1) GOTO 2341
      CALL BDRCHK(JW,N+1,M-1,ICODE)
      IF(ICODE .EQ. 1) GOTO 341
      GOTO 304
  341 CONTINUE
      ISW=0
      KURI=KURI1
  344 CALL HOMCHK(IW,N-1,M-1,7,KURI,KH)
      CALL HOMCHK(IW,N+1,M+1,8,KURI,LH)
      IF(KH.EQ.0 .AND. LH.EQ.-1) GOTO 342
      IF(KH.EQ.-1 .AND. LH.EQ.0) GOTO 343
      IF(KURI2.EQ.0 .OR. ISW.EQ.1) GOTO 304
      KURI=KURI2
      ISW=1
      GOTO 344
  342 CALL SCRMBL(N,M,4,KURI,IRAN)
      GOTO 304
  343 CALL SCRMBL(N-1,M-1,4,KURI,IRAN)
      GOTO 304
C
```

```
 2341 CONTINUE
      ISW=0
      KURI=KURI1
 2344 CALL ORTHOM(IW,N-1,M,7,KURI,KH)
      CALL ORTHOM(IW,N,M+1,8,KURI,LH)
      IF(KH.EQ.0 .AND. LH.EQ.-1) GOTO 2342
      IF(KH.EQ.-1 .AND. LH.EQ.0) GOTO 2342
      IF(KURI2.EQ.0 .OR. ISW.EQ.1) GOTO 304
      KURI=KURI2
      ISW=1
      GOTO 2344
 2342 CALL SCRMBL(N-1,M,4,KURI,IRAN)
      GOTO 304
  304 CONTINUE
  303 CONTINUE
      GOTO 122
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCC   DISPLAY-IP...
  902 GOTO 121
CCC   DISPLAY-KP
  903 CALL WRTIMG(11,33,0,282,235,KP,16,Y'1FFF',0,IST)
      GOTO 122
CCC   EXCHANGE IP AND KP...
  904 DO 201 M=1,236
      DO 202 N=1,250
      IW=IP(N,M)
      IP(N,M)=KP(N,M)
      KP(N,M)=IW
  202 CONTINUE
  201 CONTINUE
      GOTO 122
CCC   RD-MT1,  IF EOF IS FOUND THEN REWIND.....
  905 CALL SYSIO(IPBLK,RD,1,IP,JJ,0)
      CALL ILBYTE(IST,IPBLK,2)
      IF(IST .EQ. 0) GOTO 121
      IF(IST .NE. X'88') PAUSE 'MT1-RD-ERROR...'
       REWIND 1
      GOTO 122
CCC   WT-MT1 AND EOF (,OVER-WRITE IS PROTECTED.)...
  906 CALL SYSIO(IPBLK,RD,1,JP,JJ,0)
      CALL ILBYTE(IST,IPBLK,2)
      IF(IST.EQ. 0) GOTO 906
      IF(IST .NE. X'88') PAUSE 'MT-ERR(JOB QUIES)'
      BACKSPACE 1
      CALL SYSIO(IPBLK,WT,1,IP,JJ,0)
      ENDFILE 1
      BACKSPACE 1
      GOTO 122
CCC   EDGE-PRESERVING-SMOOTH(EGPR) IP TO KP------------------------
  907 DO 500 M=1,236
      DO 501 N=1,250
C     EXTRACT GREEN-FIELD(4-BITS)...
      IPW=IP(N,M)
      IP(N,M)=IPW/16-(IPW/256)*16
  501 CONTINUE
  500 CONTINUE
C
      CALL WRTIMG(11,33,0,282,235,IP,16,Y'1FFF',0,IST)
C
      CALL EGPR
C
      DO 502 M=1,236
      DO 503 N=1,250
      IPW=IP(N,M)
      IP(N,M)=(IPW*16+IPW)*16
      KPW=KP(N,M)
```

```
      KP(N,M)=(KPW*16+KPW)*16
  503 CONTINUE
  502 CONTINUE
      GOTO 122
CCC   REFINE IP (I.E. BLUE AND GREEN ARE SET TO EQUIVALENT,
CCC                  AND RED IS 0)...
  908 DO 203 M=1,236
      DO 204 N=1,250
      IGREEN=IP(N,M)/16
      IGREEN=IGREEN-(IGREEN/16)*16
      IP(N,M)=(IGREEN*16+IGREEN)*16
  204 CONTINUE
  203 CONTINUE
      GOTO 121
CCC
 2000 CONTINUE
      WRITE(5,2001)
 2001 FORMAT(' KEYIN 0, OR IX,IY,JX,JY...')
      READ(5,2005) IX,IY,JX,JY
 2005 FORMAT(4I3)
      DO 2007 IW=1,3
      DO 2008 IW1=1,250
 2008 JP(IW1,IW)=0
 2007 CONTINUE
      IF(IX.EQ.0 .AND. JX.EQ.0) STOP
      IWY=IY-1
      IF(IWY .EQ. 0) GOTO 2006
      DO 2002 IW1=1,IWY
      IW=IW1-1
      CALL WRTIMG(11,33,IW,282,IW,JP,16,Y'1FFF',0,IST) .
 2002 CONTINUE
 2006 1X1=33+IX-1
      IX2=33+JX
      DO 2003 IW1=IY,JY
      IY1=IW1-1
      CALL WRTIMG(11,33,IY1,IX1,IY1,JP,16,Y'1FFF',0,IST)
      CALL WRTIMG(11,IX2,IY1,282,IY1,JP,16,Y'1FFF',0,IST)
 2003 CONTINUE
      DO 2004 IW1=JY,235
      CALL WRTIMG(11,33,IW1,282,IW1,JP,16,Y'1FFF',0,IST)
 2004 CONTINUE
      GOTO 122
CCC
      CALL WRTTXT(11,10,10,CH,1,1,ICOL,Y'0A',IST)
      END
      SUBROUTINE NEGA(IV,IX,IY,ICODE)
      INTEGER U,D,L,R
      INTEGER*2 IP,JP,PIXEL(1),KP,CH
      COMMON IP(250,236),JP(250,236),CH(20),KP(250,236)
C CHECK P(IX,IY) (=IV) IS NEGATIVE-ISOLATED-PIXEL(NIP) OR NOT,
C IF SO, JP(IX,IY) IS SET AND DISPLAYED IN BLUE+1.
      I=Y'110'
      U=IP(IX,IY-1)-IV-I
      D=IP(IX,IY+1)-IV-I
      L=IP(IX-1,IY)-IV-I
      R=IP(IX+1,IY)-IV-I
      IF(U.EQ.0 .AND. D.EQ.0 .AND. L.EQ.0 .AND. R.EQ.0) GOTO 10
      ICODE=0
      RETURN
CCC
   10 ICODE=1
      PIXEL(1)=(IV/256+1)*256
C     NIP IS FOUND(BLUE)...
      JP(IX,IY)=PIXEL(1)
      CALL WRTIMG(11,IX+32,IY-1,IX+32,IY-1,PIXEL,16,Y'1FFF',0,IST)
      RETURN
```

```
      END
      SUBROUTINE BDRCHK(JW,JX,JY,ICODE)
      INTEGER*2 IP,JP,CH,KP
      COMMON IP(250,236),JP(250,236),CH(20),KP(250,236)
C ICODE=1:SAME BORDER FOUND,  0:FAIL,  -1:IMAGE FRAME-OUT.
      IF(JX.LT.3 .OR. JX.GT.248) GOTO 10
      IF(JY.LT.3 .OR. JY.GT.234) GOTO 10
      JJ=JP(JX,JY)
      IF(JJ.GT.0 .AND. JJ.LT.Y'0010') GOTO 20
   30 ICODE=0
      RETURN
   20 IF(JJ .NE. JW) GOTO 30
      ICODE=1
      RETURN
   10 ICODE=-1
      RETURN
      END
      SUBROUTINE HOMCHK(IW,IX,IY,ID,KURI,IHOM)
      INTEGER*2 IP,JP,CH,KP
      COMMON IP(250,236),JP(250,236),CH(20),KP(250,236)
C  FROM (IX,IY), DIR=ID, NO. OF PIXELS=KURI.
C  IHOM=0:ALL THE SAME TO IW,
C        -1:ALL THE SAME TO IW-1,
C         1:OTHERS.
      IWW=IW-Y'110'
      KARI=KURI
      IF(IP(IX,IY) .EQ. IW) GOTO 10
      IF(IP(IX,IY) .EQ. IWW) GOTO 20
   60 IHOM=1
      RETURN
CCCCC
   10 ICHK=IW
      KURI=KURI-1
      IHOM=0
      GOTO 30
   20 ICHK=IWW
      IHOM=-1
      GOTO 30
CCC
   30 IF(ID.LT.1 .OR. ID.GT.8) PAUSE 'HOMCHK ERR...'
      GOTO (31,32,33,34,35,36,37,38),ID
   31 IF(IX-KURI .LT. 1) GOTO 60
      IDX=-1
      IDY=0
      GOTO 40
   32 IF(IX+KURI .GT. 250) GOTO 60
      IDX=1
      IDY=0
      GOTO 40
   33 IF(IY-KURI .LT. 1) GOTO 60
      IDX=0
      IDY=-1
      GOTO 40
   34 IF(IY+KURI .GT. 236) GOTO 60
      IDX=0
      IDY=1
      GOTO 40
   35 IF(IX-KURI.LT.1 .OR. IY+KURI.GT.236) GOTO 60
      IDX=-1
      IDY=1
      GOTO 40
   36 IF(IX+KURI.GT.250 .OR. IY-KURI.LT.1) GOTO 60
      IDX=1
      IDY=-1
      GOTO 40
   37 IF(IX-KURI.LT.1 .OR. IY-KURI.LT.1) GOTO 60
```

```
      IDX=-1
      IDY=-1
      GOTO 40
   38 IF(IX+KURI.GT.250 .OR. IY+KURI.GT.236) GOTO 60
      IDX=1
      IDY=1
      GOTO 40
CCC
   40 CONTINUE
      IXW=IX
      IYW=IY
      DO 50 IK=1,KURI-1
      IXW=IXW+IDX
      IYW=IYW+IDY
      IF(ICHK .EQ. IP(IXW,IYW)) GOTO 50
      IHOM=1
   50 CONTINUE
      KURI=KARI
      RETURN
      END
      SUBROUTINE ORTHOM(IW,IX,IY,ID,KURI,IHOM)
      INTEGER*2 IP,JP,CH,KP
      COMMON IP(250,236),JP(250,236),CH(20),KP(250,236)
C  FROM (IX,IY), DIR=ID, NO. OF PIXELS=KURI.
C  IHOM=0:ALL THE SAME TO IW,
C        -1:ALL THE SAME TO IW-1,
C         1:OTHERS.
C     WRITE(5,1000) IW,IX,IY,ID,KURI,IHOM
C1000 FORMAT(' *ORTHOM ',6I5)
      IWW=IW-Y'110'
      KARI=KURI
      IF(IP(IX,IY) .EQ. IW) GOTO 10
      IF(IP(IX,IY) .EQ. IWW) GOTO 20
   60 IHOM=1
C     WRITE(5,1002) IHOM
C1002 FORMAT(' RETURN-ORTHOM.*.*.*',I5)
      RETURN
CCCCC
   10 ICHK=IW
      IHOM=0
      GOTO 30
   20 ICHK=IWW
      IHOM=-1
      GOTO 30
CCC
   30 IF(ID.LT.1 .OR. ID.GT.8) PAUSE 'HOMCHK ERR...'
      GOTO (31,32,33,34,35,36,37,38),ID
   31 IF(IX-KURI .LT. 1)`GOTO 60
      IDX=-1
      IDY=0
      GOTO 40
   32 IF(IX+KURI .GT. 250) GOTO 60
      IDX=1
      IDY=0
      GOTO 40
   33 IF(IY-KURI .LT. 1) GOTO 60
      IDX=0
      IDY=-1
      GOTO 40
   34 IF(IY+KURI .GT. 236) GOTO 60
      IDX=0
      IDY=1
      GOTO 40
   35 IF(IX-KURI.LT.1 .OR. IY+KURI.GT.236) GOTO 60
      IDX=-1
      IDY=1
```

```
      GOTO 40
   36 IF(IX+KURI.GT.250 .OR. IY-KURI.LT.1) GOTO 60
      IDX=1
      IDY=-1
      GOTO 40
   37 IF(IX-KURI.LT.1 .OR. IY-KURI.LT.1) GOTO 60
      IDX=-1
      IDY=-1
      GOTO 40
   38 IF(IX+KURI.GT.250 .OR. IY+KURI.GT.236) GOTO 60
      IDX=1
      IDY=1
      GOTO 40
CCC
   40 CONTINUE
      IXW=IX
      IYW=IY
      DO 50 IK=1,KURI-1
      IXW=IXW+IDX
      IYW=IYW+IDY
      IF(ICHK .EQ. IP(IXW,IYW)) GOTO 50
      IHOM=1
   50 CONTINUE
      KURI=KARI
C     WRITE(5,1001) IHOM
C1001 FORMAT(' RETURN-ORTHOM.....',I5)
      RETURN
      END
      SUBROUTINE SCRMBL(JX,JY,ID,KURI,IRAN)
      INTEGER*2 IP,JP,CH,KP
      COMMON IP(250,236),JP(250,236),CH(20),KP(250,236)
C     WRITE(5,1000) JX,JY,ID,KURI,IRAN
C1000 FORMAT(' SCRMBL ',4I6,I11)
      IF(KURI .EQ. 0) PAUSE 'SCRMBL ERR(KURI=0)...'
      N2=2*KURI
      N3=N2+1
      IF(ID.LT.1 .OR. ID.GT.4) PAUSE 'SCRMBL ERR...'
      GOTO (11,12,13,14),ID
   11 IDX=1
      IDY=0
      GOTO 20
   12 IDX=0
      IDY=1
      GOTO 20
   13 IDX=1
      IDY=-1
      GOTO 20
   14 IDX=1
      IDY=1
      GOTO 20
C
   20 KX=JX-KURI*IDX
      KY=JY-KURI*IDY
      ICL=IP(JX,JY)+Y'1000'
      JXW=JX+IDX
      JYW=JY+IDY
      ICR=IP(JXW,JYW)+Y'1000'
      DO 30 K=1,N2
      KX=KX+IDX
      KY=KY+IDY
      CALL RANGEN(IRAN,N3,IVAL)
      IF(IVAL .LT. K) GOTO 31
      IF(KP(KX,KY) .GE. Y'1000') GOTO 32
      KP(KX,KY)=ICL
      GOTO 32
   31 CONTINUE
```

```
      IF(KP(KX,KY) .GE. Y'1000') GOTO 32
      KP(KX,KY)=ICR
   32 CALL WRTIMG(11,KX+32,KY-1,KX+32,KY-1,KP(KX,KY),16,Y'1FFF',0,IST)
   30 CONTINUE
      RETURN
      END
C GENERATION OF RANDOM NUMBER IN A RANGE.
      SUBROUTINE RANGEN(IX,IRANGE,IVAL)
      IY=IX*65539
      IF(IY) 5,6,6
    5 IY=(IY+2147483647)+1
    6 IW=IY/13
      IX=IY
      IVAL=IW-(IW/IRANGE)*IRANGE
      RETURN
      END
```

INSTITUTE OF INFORMATION SCIENCE AND ELECTRONICS
UNIVERSITY OF TSUKUBA
SAKURA-MURA, NIIHARI-GUN, IBARAKI, JAPAN

| REPORT DOCUMENTATION PAGE | REPORT NUMBER<br>ISE-TR-82-27 |
|---|---|

**TITLE**

FALSE CONTOUR REMOVAL BY RANDOM BLURRING

**AUTHOR(S)**

Seiichi Nishihara

Katsuo Ikeda

| REPORT DATE | NUMBER OF PAGES |
|---|---|
| April 5, 1982 | 28 |
| **MAIN CATEGORY** | **CR CATEGORIES** |
| Image Processing | I.3.3, I.4.3, I.4.4 |

**KEY WORDS**

image processing, image display algorithm, dither, border, gray level, false contour

**ABSTRACT**

A method is presented to eliminate spurious brightness discontinuities, or false contours, in areas in which brightness varies gradually, while preserving edge sharpness. The basic idea of the method is randomly blurring points only in the neighborhood of each border on which those spurious brightness discontinuities may occur. The method is used for producing natural appearance of a picture for the human eye, not for picture analysis.

**SUPPLEMENTARY NOTES**