dilation

Source: http://sci.tech−archive.net/Archive/sci.image.processing/2006−07/msg00065.html

- *From*: "h4mm3r`" <cbiacca@xxxxxxxxx>
- *Date*: 11 Jul 2006 10:18:51 −0700

Hi there,
i'm new to this group n i hope not to be out of topic.
I'm trying to create a dilate() funcion by which i can dilate my
inputimage (m_inputImage). I check all background points of my input
image, if one of them is a foreground point's neighboor (by chessboard
distance), then i add it (or them) to the object.

This is my code, but it doesn't work:

```
static const unsigned char foreground = 0;
static const unsigned char background = 255;

unsigned char * neighD8(IplImage *img, int r, int c) {
unsigned char *n = (unsigned char *)malloc((sizeof(unsigned char))*8);

int k=0;
for(int i = r−1; i<=r+1; i++)
for(int j = c−1; j<=c+1; j++) {
if ((i!=r)&&(j!=c)) {
n[k]=img−>imageData[(i*img−>widthStep)+j];
k++;
}
}
return n;
}



void dilation(void) {
int k = 1;
int *kernel = new int[(2*k+1)*(2*k+1)];

if(m_inputImage−>nChannels!=1) {
AfxMessageBox("immagine RGB!");
return;
}

CvSize size;
```

```
size.height = m_inputImage−>height−2*k;
size.width = m_inputImage−>width−2*k;
m_outputImage = cvCreateImage(size,IPL_DEPTH_8U,1);

unsigned char* inputdata=((unsigned char*)(m_inputImage−>imageData));

for(int i = 0; i < 2*k+1; i++) {
for(int j = 0; j < 2*k+1; j++) {
kernel[i*(2*k+1)+j] = (int)1;
}
}

IplImage* tempimg = autothresholding(m_inputImage);

cvNamedWindow("test", 0);
cvResizeWindow("test",tempimg−>width,tempimg−>height);
cvShowImage("test", tempimg);

int temp;
for(int r = 0; r < size.height; r++) {
for(int c = 0; c < size.width; c++) {
m_outputImage−>imageData[(r*m_outputImage−>widthStep)+c] =
tempimg−>imageData[(r*tempimg−>widthStep)+c];
}
}

for(int r=k; r < tempimg−>height−k; r++) {
for(int c = k; c < tempimg−>width−k; c++) {

if(tempimg−>imageData[r*tempimg−>widthStep+c] ==
background) {
unsigned char *vicinato = neighD8(tempimg,r,c);
for(int k = 0 ; k < 8; k++) {
if (vicinato[k] == foreground) {

m_outputImage−>imageData[(r*m_outputImage−>widthStep)+c]+=foreground;
}
}
}
}
}
}

.
```