

Software Used for Personal Identification and Recognition Based on Iris Patterns

Valeriu Lupu, Catalin Lupu

The „Stefan cel Mare” University of Suceava
valeriul@seap.usv.ro, lupucata@yahoo.com

Abstract: In computer systems, there is an urgent need for accurate authentication techniques to prevent unauthorized access. Authentication is the process of confirming the correctness of the claimed identity. Many computers that store critical information are vulnerable to unauthorized access because of weak authentication. In some cases, the safety of the public can be at risk, such as in the case of a multi-billion dollar passenger-screening system defenceless against terrorists with forged security badges. Traditional authentication techniques such as the ubiquitous username / password method are inadequate for personal identity since they can only provide proof of possession and/or proof of knowledge. Only biometrics, the authentication of individuals using biological identifiers, can offer true proof of identity. Current research suggests that multimodal biometric systems, those that use more than one biological identifier, can improve the accuracy of biometric systems. This improvement in accuracy depends on critical factors in design, implementation and security. This paper explains each of these critical factors so the increase in accuracy observed in current research can be achieved in real-world applications.

Keywords: biometric, identification, verification, recognition, image acquisition, iris, fingerprint, face detection.

This article presents a software for recognition and identification of people by using iris patterns. The program's main window is presented below (Figure 1).

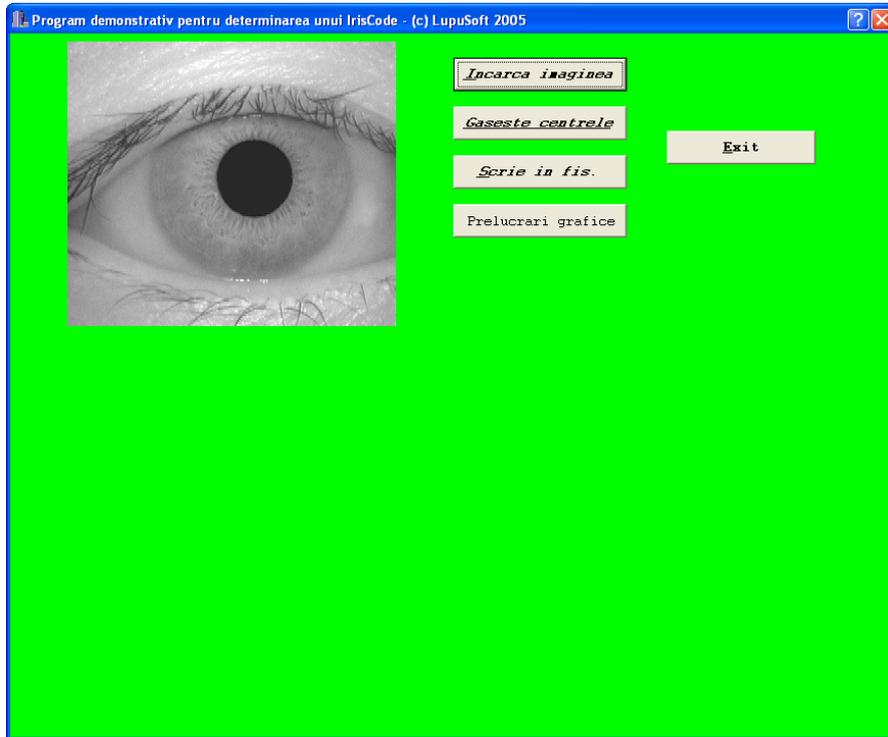


Figure 1
The application's main window

Pressing on the button „Download the image”, a window like the one in the next figure opens (Figure 2):

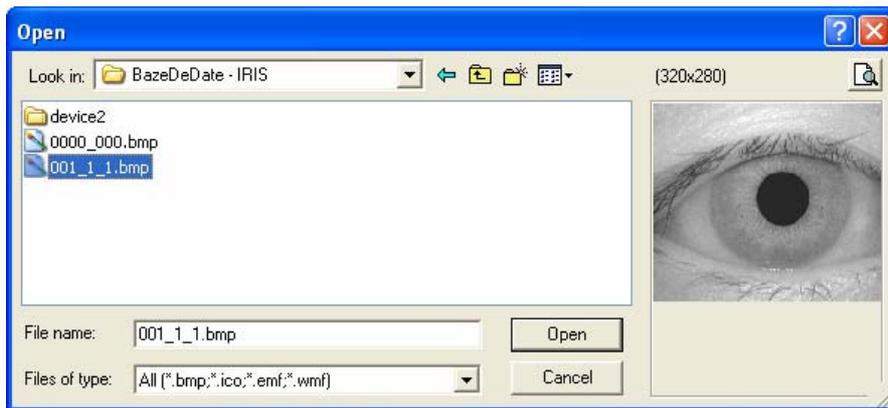


Figure 2
The download of an image

The finding button of the centers will set the centers of the pupil and the iris, at the end showing a dialog window (at such type) like this:

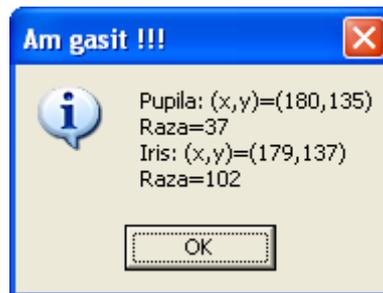


Figure 3

The dialog window with determination of the centers of the pupil and the iris

By pressing the button “Graphical transformations“, the pupil’s image is shown as below (Figure 4):

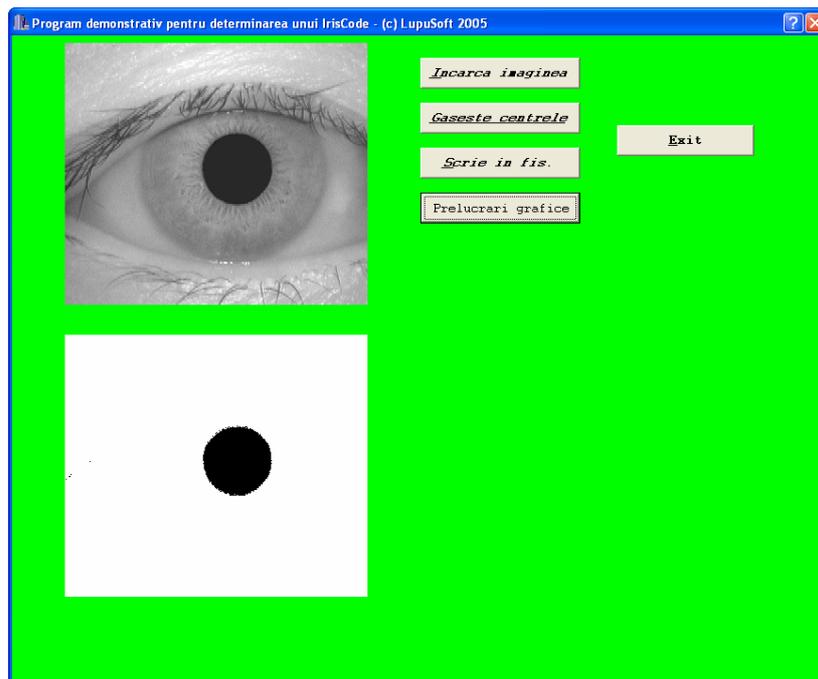


Figure 4

The image is presented in the left down side

In the lower down figure is presenting the second stage of the graphical transformations, in which the iris is segmented in more concentric circles (Figure 5).

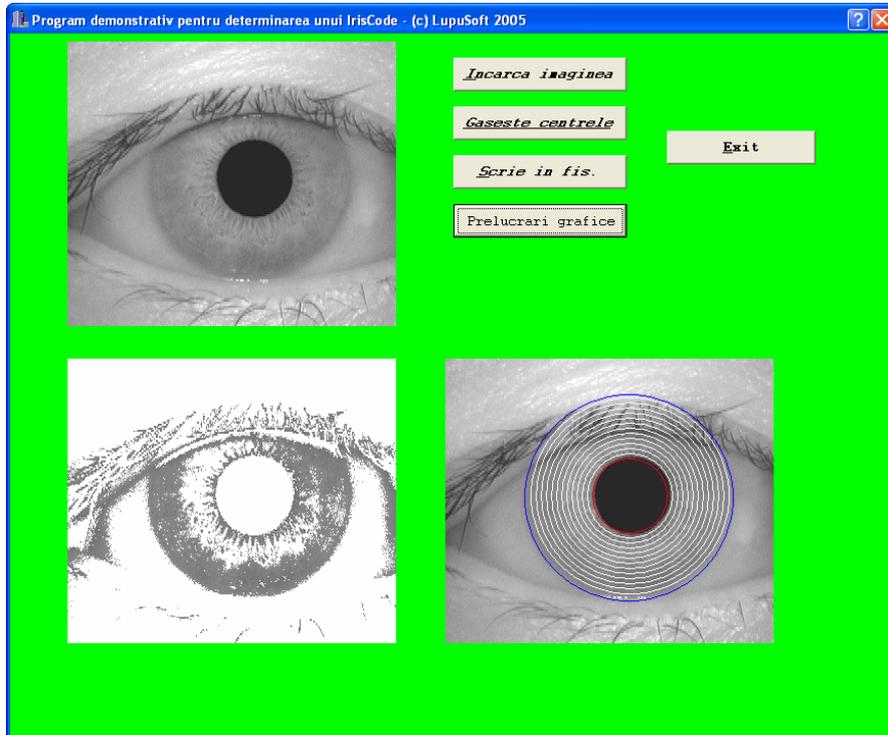


Figure 5

The segmented image of the iris is in the right side

Also, tests have been realized in what concerning the 2D representation of an image read from a bitmap folder (BMP). In this way, was obtained the representation presented in the Figure 6.

A 3D representation is given in the lower down figure (Figure 7).

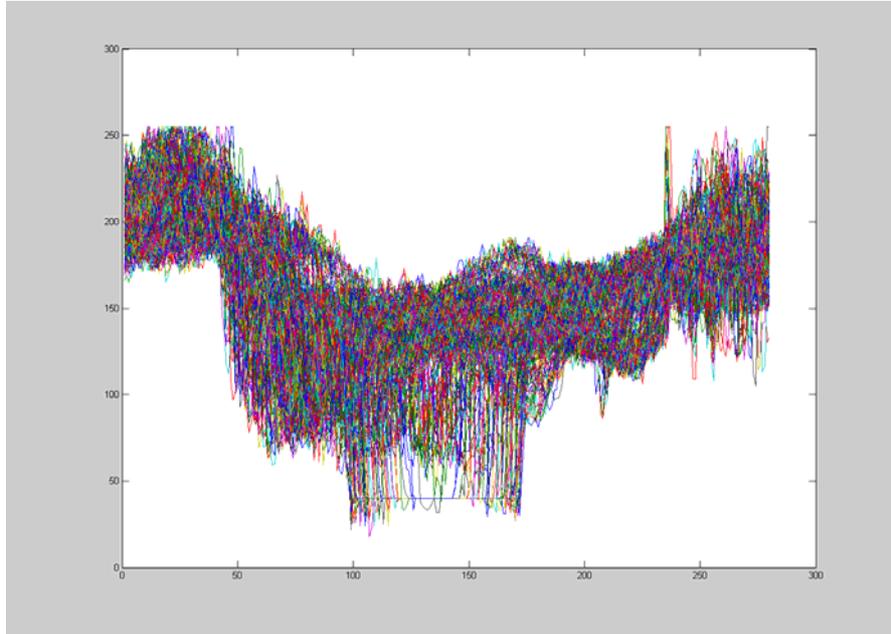


Figure 6
The 2D representation of image

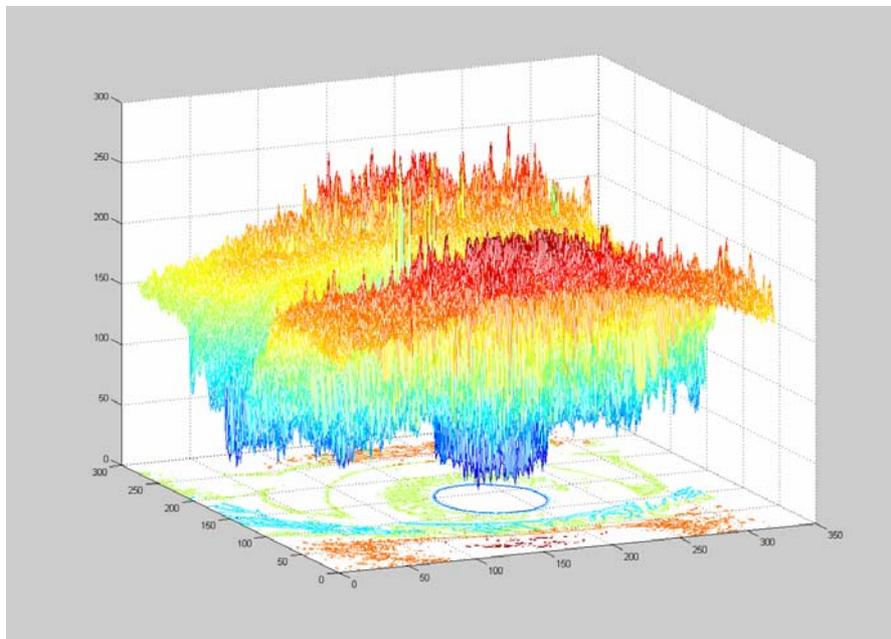


Figure 7
The 3D representation of the image of an iris

From another angle, this representation is presented lower down (Figure 8):

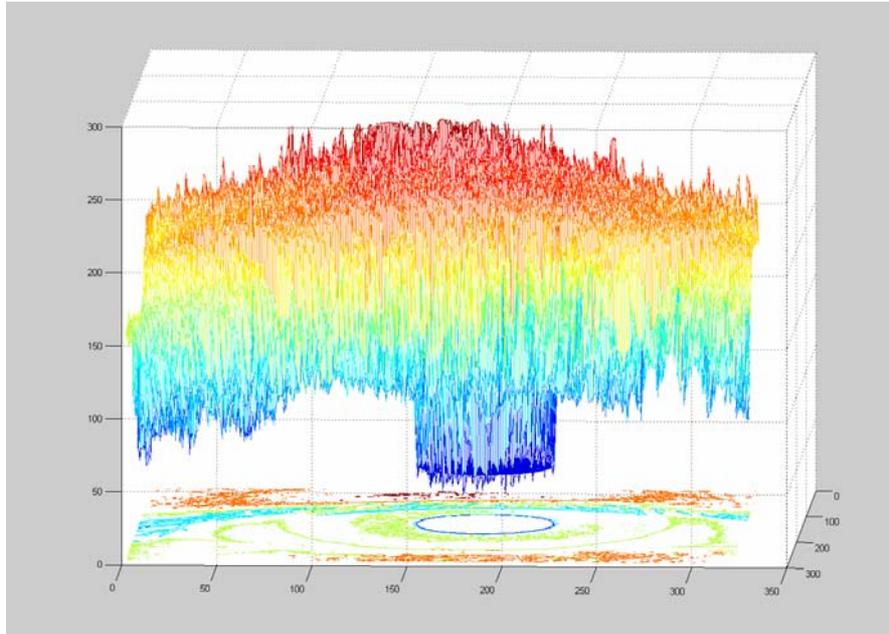


Figure 8

The 3D representation of the iris image with making evident the pupil boundary

Also, was been calculated and inserted in a „workspace” of the Matlab the coefficients of the convolution of the Gabor filter.

The software is presented lower down (Figure 9):

```
I=imread('l.bmp');
```

```
figure,imshow(I);
```

```
for i=0:15
```

```
    eval(['m' num2str(i) '']);
```

```
end
```

```
mask0even = [
```

```
    0.001080, 0.002584, 0.005095, 0.008272, 0.011065, 0.012191, 0.011065,  
    0.008272, 0.005095, 0.002584, 0.001080,
```

```
    0.016185, 0.038738, 0.076365, 0.123999, 0.165854, 0.182737, 0.165854,  
    0.123999, 0.076365, 0.038738, 0.016185,
```

```
    -0.003095, -0.007408, -0.014604, -0.023713, -0.031717, -0.034946, -0.031717, -  
    0.023713, -0.014604, -0.007408, -0.003095,
```

```
-0.051930, -0.124293, -0.245024, -0.397863, -0.532158, -0.586329, -0.532158, -  
0.397863, -0.245024, -0.124293, -0.05130,  
0.002255, 0.005397, 0.010640, 0.017277, 0.023108, 0.025460, 0.023108,  
0.017277, 0.010640, 0.005397, 0.002255,  
0.076591, 0.183318, 0.361381, 0.586800, 0.784869, 0.864765, 0.784869,  
0.586800, 0.361381, 0.183318, 0.076591,  
0.002255, 0.005397, 0.010640, 0.017277, 0.023108, 0.025460, 0.023108,  
0.017277, 0.010640, 0.005397, 0.002255,  
-0.051930, -0.124293, -0.245024, -0.397863, -0.532158, -0.586329, -0.532158, -  
0.397863, -0.245024, -0.124293, -0.051931,  
-0.003095, -0.007408, -0.014604, -0.023713, -0.031717, -0.034946, -0.031717, -  
0.023713, -0.014604, -0.007408, -0.003095,  
0.016185, 0.038738, 0.076365, 0.123999, 0.165854, 0.182737, 0.165854,  
0.123999, 0.076365, 0.038738, 0.016185,  
0.001080, 0.002584, 0.005095, 0.008272, 0.011065, 0.012191, 0.011065,  
0.008272, 0.005095, 0.002584, 0.001080  
];  
maskOdd = [  
-0.006750, -0.016157, -0.031850, -0.051717, -0.069174, -0.076216, -0.069174, -  
0.051717, -0.031850, -0.016157, -0.006750,  
0.002082, 0.004983, 0.009823, 0.015951, 0.021335, 0.023507, 0.021335,  
0.015951, 0.009823, 0.004983, 0.002082,  
0.031950, 0.076470, 0.150748, 0.244781, 0.327404, 0.360732, 0.327404,  
0.244781, 0.150748, 0.076470, 0.031950,  
-0.003364, -0.008051, -0.015872, -0.025772, -0.034471, -0.037980, -0.034471, -  
0.025772, -0.015872, -0.008051, -0.003364,  
-0.069501, -0.166347, -0.327927, -0.532478, -0.712211, -0.784711, -0.712211, -  
0.532478, -0.327927, -0.166347, -0.069501,  
0.000000, 0.000000, -0.000000, 0.000000, -0.000000, 0.000000, 0.000000,  
0.000000, -0.000000, 0.000000, -0.000000,  
0.069501, 0.166347, 0.327927, 0.532478, 0.712211, 0.784711, 0.712211,  
0.532478, 0.327927, 0.166347, 0.069501,  
0.003364, 0.008051, 0.015872, 0.025772, 0.034471, 0.037980, 0.034471,  
0.025772, 0.015872, 0.008051, 0.003364,
```

```

-0.031950, -0.076470, -0.150748, -0.244781, -0.327404, -0.360732, -0.327404, -
0.244781, -0.150748, -0.076470, -0.031950,

-0.002082, -0.004983, -0.009823, -0.015951, -0.021335, -0.023507, -0.021335, -
0.015951, -0.009823, -0.004983, -0.002082,

0.006750, 0.016157, 0.031850, 0.051717, 0.069174, 0.076216, 0.069174,
0.051717, 0.031850, 0.016157, 0.006750
];

```

```

figure('Name','Even Gabor filter of 0
degree','NumberTitle','off'),imshow(conv2(I,mask0even));

```

```

figure('Name','Filtru Gabor impar de ordinul
0','NumberTitle','off'),imshow(conv2(I,mask0odd));

```

In this way there have been calculated 15 coefficients of convolution of the Gabor filter. The results of the convolution with the Gabor filter are presented lower down

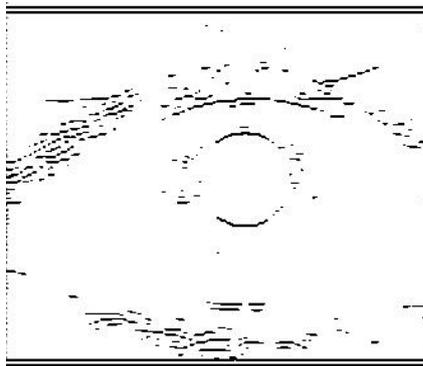


Figure 9
The even Gabor even of 0 degree

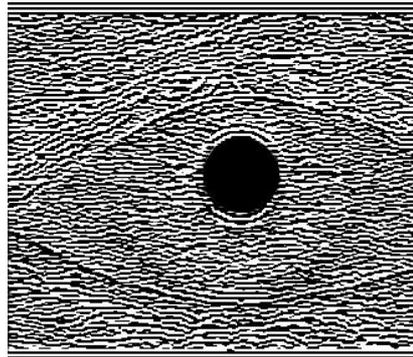


Figure 10
The Odd Gabor Filter of 0 degree

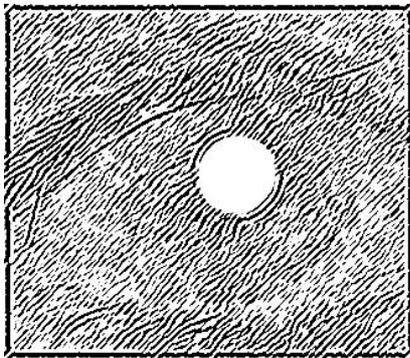


Figure 11
The Even Gabor filter of 1st degree

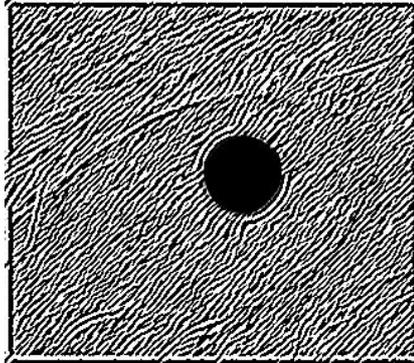


Figure 12
The Odd Gabor Filter of 1 degree

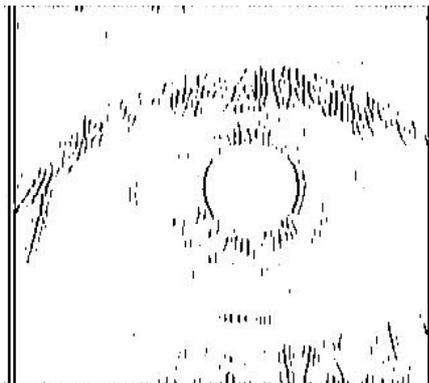


Figure 13
The Even Gabor filter of the 2nd degree

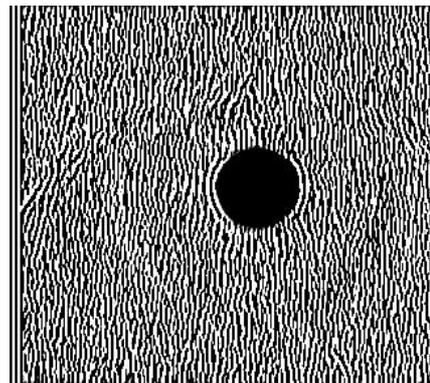


Figure 14
The Odd Gabor Filter of the 2nd degree

It is observing that the angles of the representation differ at each filtration

Also, it had been realized a program for detection („edge-detection”) by using different algorithms. The program is presented below:

```
function test();
imagine=imread('ochi.bmp');
figure('Name','Sobel','NumberTitle','off');
imshow(edge(imagine,'sobel'));
figure('Name','Canny','NumberTitle','off');
imshow(edge(imagine,'canny'));
figure('Name','Prewitt','NumberTitle','off');
```

```
imshow(edge(Imagine,'prewitt'));  
figure('Name','Roberts','NumberTitle','off');  
imshow(edge(Imagine,'roberts'));  
figure('Name','Zero cross','NumberTitle','off');  
imshow(edge(Imagine,'zerocross'));  
figure('Name','Laplacian of Gaussian','NumberTitle','off');  
imshow(edge(Imagine,'log'));  
The edge-detection results are:
```

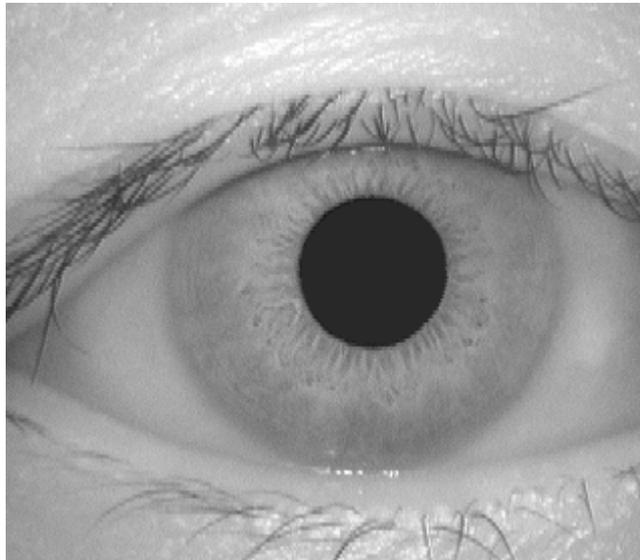


Figure 15
Initial image

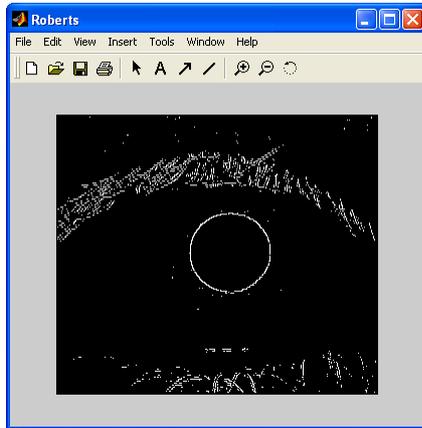


Figure 16. Roberts

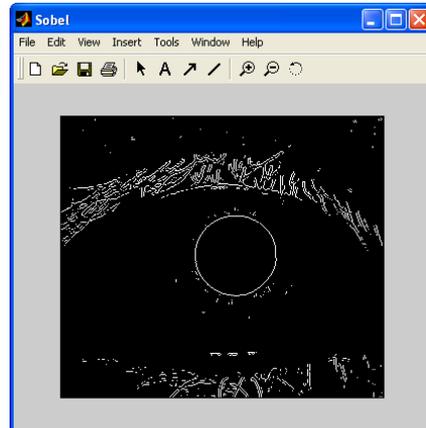


Figure 17. Sobel

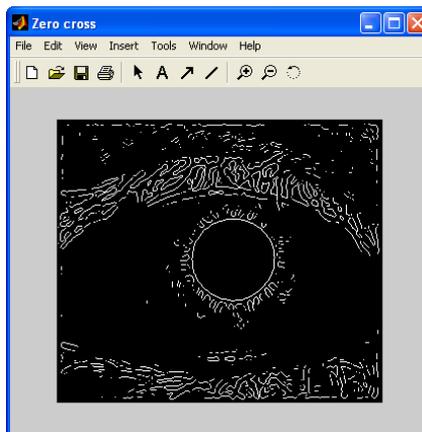


Figure 18. Zero crossin

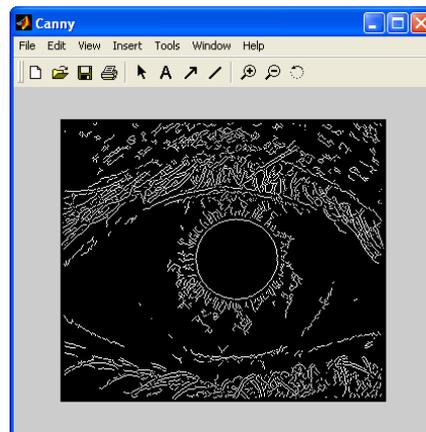


Figure 19. Gaussian's Laplacian

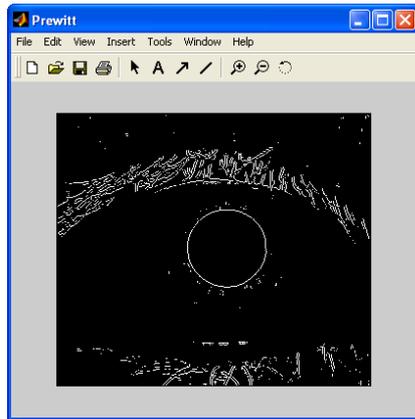


Figure 20. Prewitt

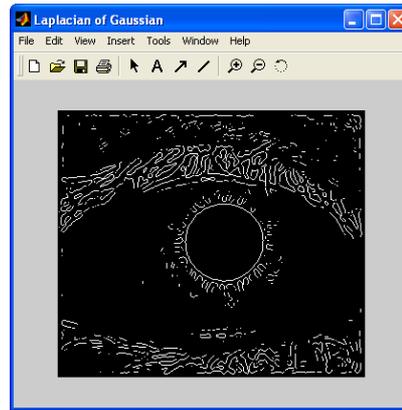


Figure 21. Canny

It was observed the fact that edge detection by using Canny method leads to the finding of more details, but this method can't be used in real-time applications because its computational time is very big (about 2 seconds for the presented image).

Differences between Canny and Sobel's filters (that has 2 filtering capacities – horizontally and vertically) are presented in the next figures.

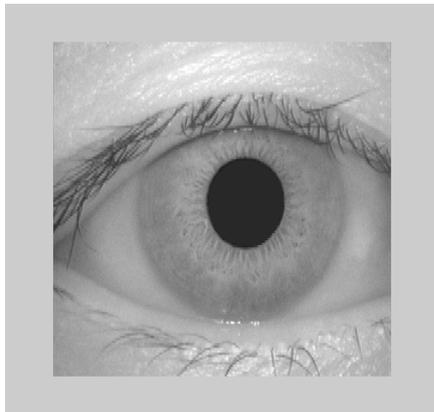


Figure 22
Initial image

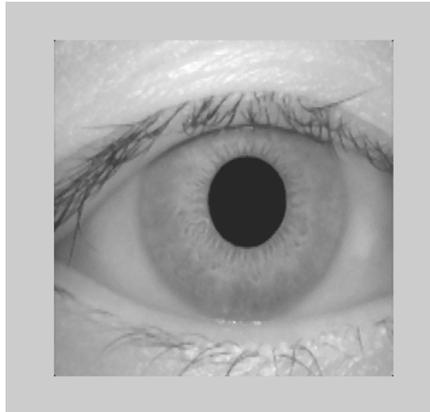


Figure 23
Median filtered image

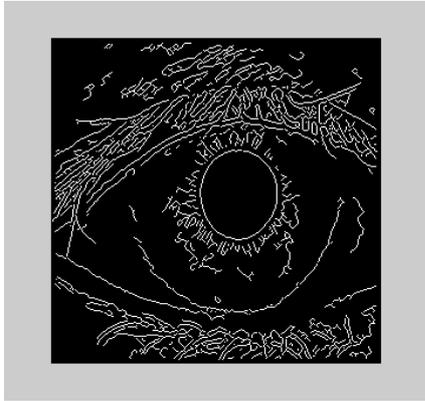


Figure 24
Canny

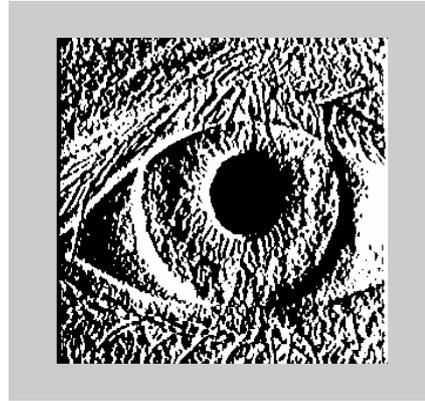


Figure 25
Horizontally Sobel

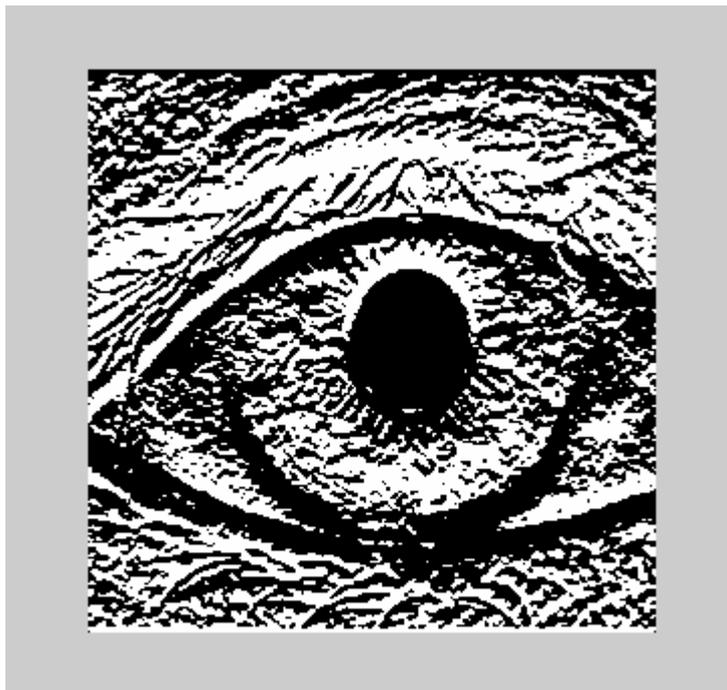


Figure 26
Vertically Sobel

The MatLab source code is:

```
fisier='1.bmp';
```

```
SobelX=[-1 0 1;-2 0 2;-1 0 1];
```

```

SobelY=[-1 -2 -1;0 0 0; 1 2 1];
MexHat=[0 0 -1 0 0; 0 -1 -2 -1 0; -1 -2 16 -2 -1; 0 -1 -2 -1 0; 1 0 -1 0 0];

% programul principal

%fisier=inputdlg('Image:','Fisier');
%fisier=char(fisier);
I=imread(fisier);
I=imresize(I,[min(size(I)) min(size(I))],'bicubic');
figure('NumberTitle','off','Name','1 - Imaginea initiala'),imshow(I);
% etape 1
I=medfilt2(I);
figure('NumberTitle','off','Name','2 - Imaginea filtrata median'),imshow(I);
% etape 2
%I=rgb2gray(I);
% etape 3
ImgCan=edge(I,'canny');
figure('NumberTitle','off','Name','3 - Canny'),imshow(ImgCan);
% etape 4
ImgSobelX=conv2(I,SobelX);
figure('NumberTitle','off','Name','4 - Sobel X'),imshow(ImgSobelX);

ImgSobelY=conv2(I,SobelY);
figure('NumberTitle','off','Name','4 - Sobel Y'),imshow(ImgSobelY);

arct =atan(ImgSobelY/ImgSobelX);
res=1;
%-----Fin-----

```

Below will be presented the Gabor filters and the MatLab programs.

```

%
%
% 1-2
sigma=0.1;
kx=4.2*2*pi;
dx=0.004;
nx=128;
x=(-nx/2+1):1:nx/2)*dx;
for i=1:length(x)
    gse(i)=(1/(sqrt(2*pi)*sigma))*exp((-x(i)*x(i))/(2*sigma*sigma))*cos(kx*x(i));
    gso(i)=(1/(sqrt(2*pi)*sigma))*exp((-x(i)*x(i))/(2*sigma*sigma))*sin(kx*x(i));
end
figure('Name','Filtru Gabor 1-D par','NumberTitle','off');
plot(x,gse),title('Filtru Gabor 1-D par'),xlabel('x'),ylabel('gse(x)');
figure('Name','Filtru Gabor 1-D impar','NumberTitle','off');
plot(x,gso),title('Filtru Gabor 1-D impar'),xlabel('x'),ylabel('gso(x)');

% 3-4
sigmat=0.031;
kt=8*2*pi;
dt=0.002;
nt=128;
t=(-nt/2+1):1:nt/2)*dt;
for i=1:length(t)
    fte(i)=(1/(sqrt(2*pi)*sigmat))*exp((-t(i)*t(i))/(2*sigmat*sigmat))*cos(kt*t(i));
    fto(i)=(1/(sqrt(2*pi)*sigmat))*exp((-t(i)*t(i))/(2*sigmat*sigmat))*sin(kt*t(i));
end
figure('Name','Camp receptiv temporal 1-D pentru functii de transfer Gabor
pare','NumberTitle','off');

```

```
plot(t,fte),title('Camp receptiv temporal 1-D pentru functii de transfer Gabor  
pare'),xlabel('t'),ylabel('fte(t)');
```

```
figure('Name','Camp receptiv temporal 1-D pentru functii de transfer Gabor  
impare','NumberTitle','off');
```

```
plot(t,fto),title('Camp receptiv temporal 1-D pentru functii de transfer Gabor  
impare'),xlabel('t'),ylabel('fto(t)');
```

```
% 5-6
```

```
for i=1:length(x)
```

```
    for j=1:length(t)
```

```
        ge(i,j)=gse(i)*fte(j);
```

```
        geminus(i,j)=gse(i)*fte(j)+gso(i)*fto(j);
```

```
    end
```

```
end
```

```
figure('Name','Profil spatiotemporal 3D al filtrului  
ge(x,t)=gse(x)*fte(t)','NumberTitle','off');
```

```
meshc(x,t,ge),title('Profil spatiotemporal 3D al functiei  
ge(x,t)=gse(x)*fte(t)'),xlabel('x'),ylabel('t'),zlabel('ge(x,t)');
```

```
figure('Name','Profil spatiotemporal 3D al filtrului ge-  
(x,t)=gse(x)*fte(t)+gso(x)*fto(t)','NumberTitle','off');
```

```
meshc(x,t,geminus),title('Profil spatiotemporal 3D al functiei ge-  
(x,t)=gse(x)*fte(t)+gso(x)*fto(t)'),xlabel('x'),ylabel('t'),zlabel('ge(x,t)');
```

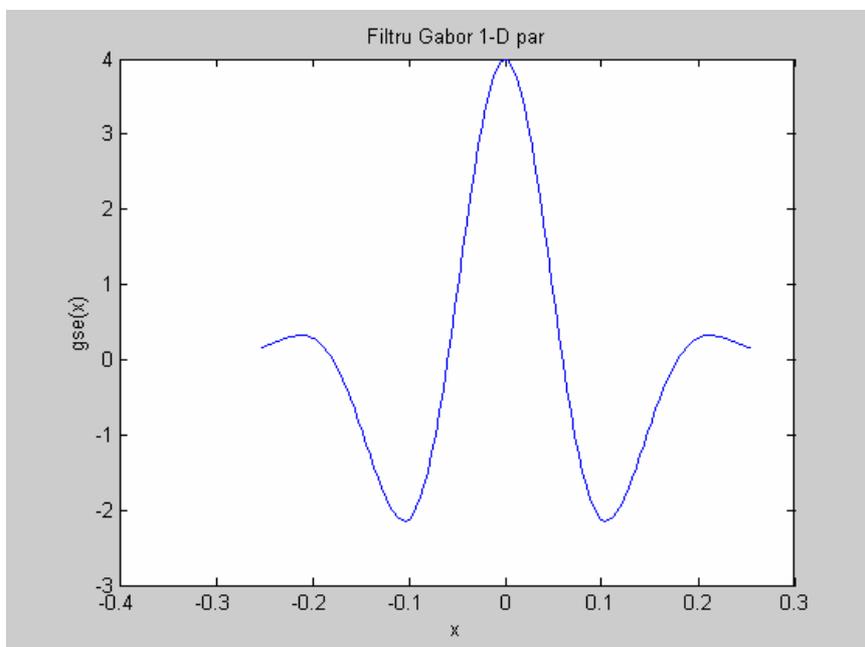


Figure 27
Even 1D Gabor filter

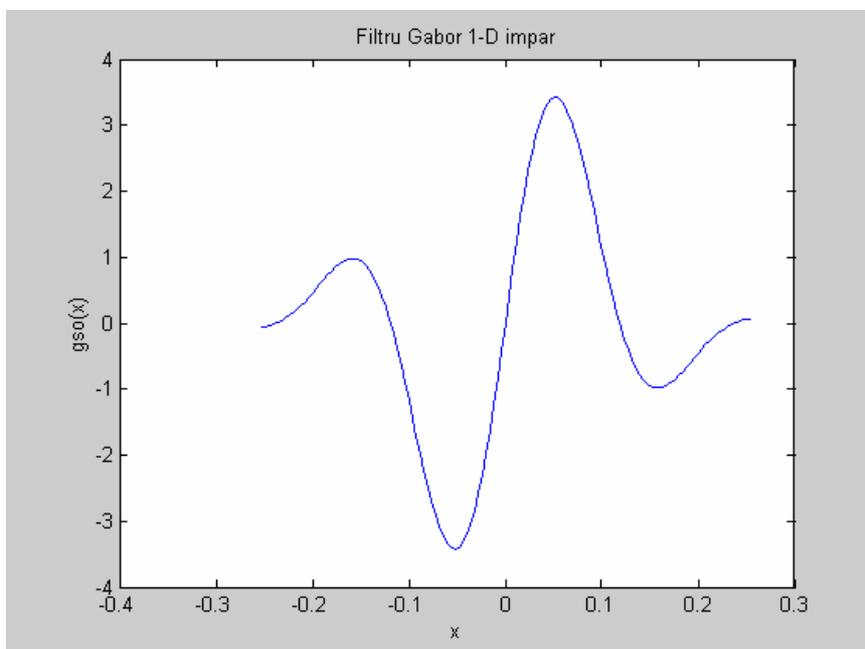


Figure 28
Odd 1D Gabor filter

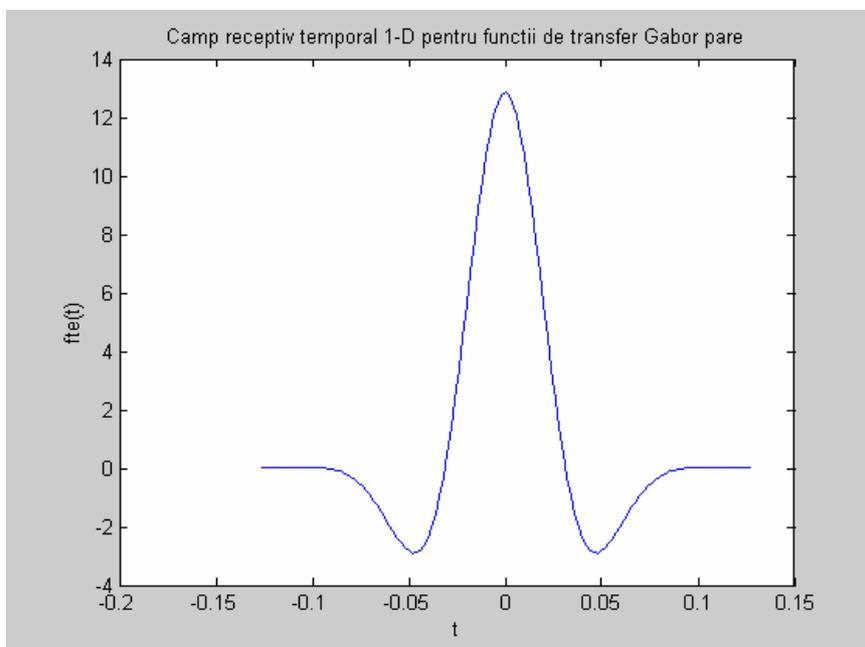


Figure 29a
Receptive field for even 1D Gabor transfer functions

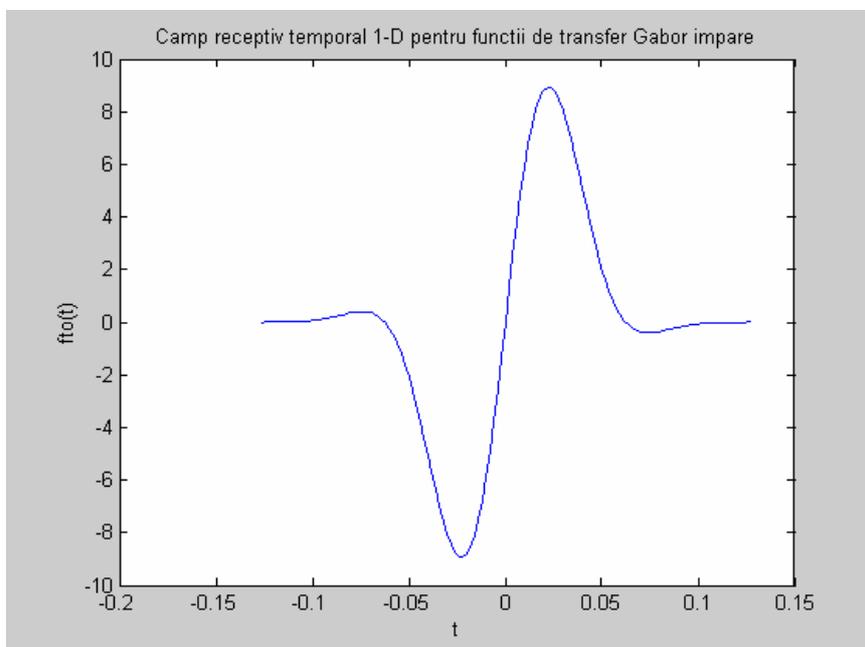


Figure 29b
Receptive field for odd 1D Gabor transfer functions

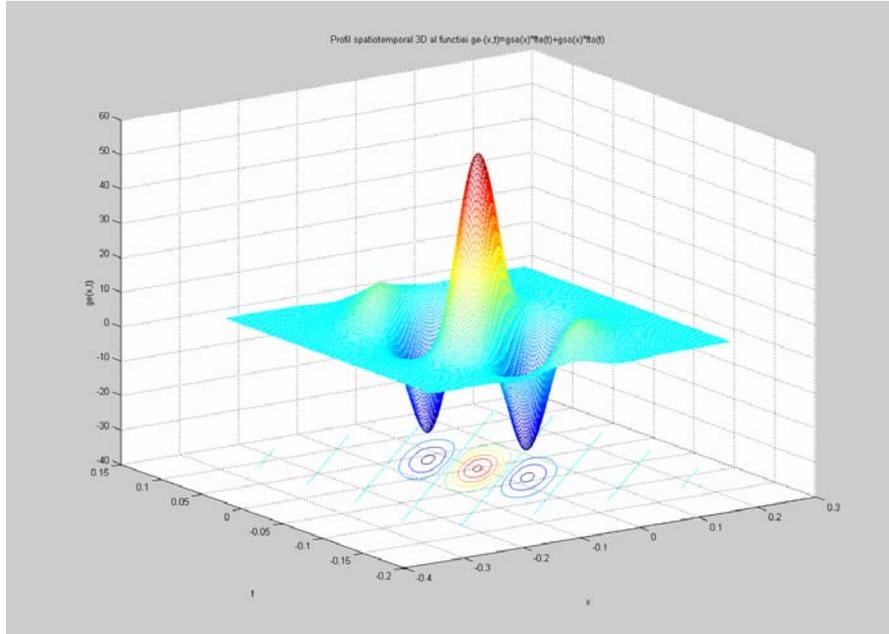


Figure 30
3D spatial-temporal profile for $g_e(x,t)$ function

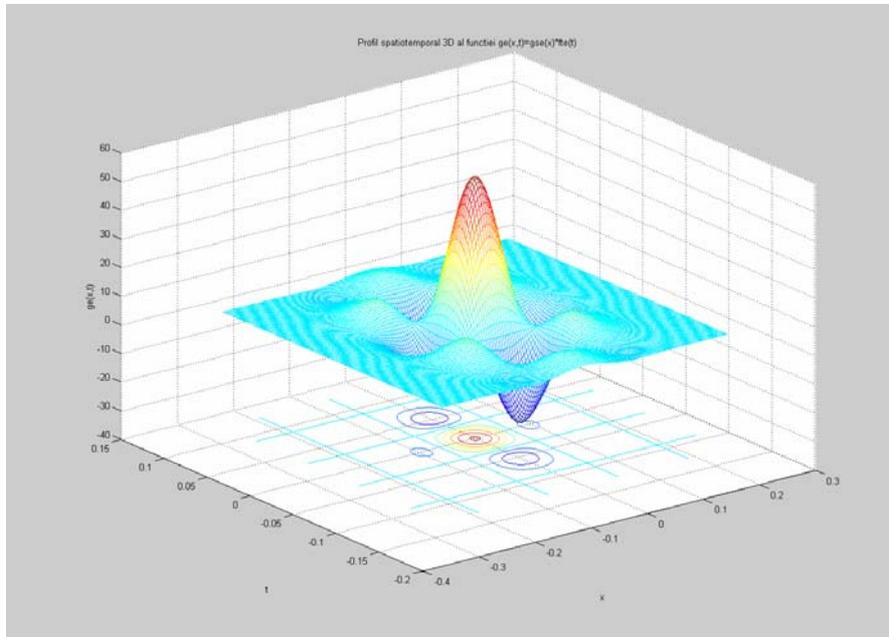


Figure 31
3D spatial-temporal profile for $g(x,t)$ function

Bibliography

- [1] Pellerin, K. (2004) "Increasing Accuracy in Multimodal Biometric Systems", SANS Institute 2004
- [2] O’Gorman. (2003) "Comparing Passwords, Tokens, and Biometrics for User Authentication."
- [3] Jain, Ross,(2004) "Multibiometric systems"
- [4] Ross, and Jain. "Information Fusion in Biometrics." Sept. 2003
- [5] Poh, and Korczak. "Hybrid Biometric Person Authentication using Face and Voice Features." 2001
- [6] J. Daugman. Biometric personal identification system based on iris analysis. United States Patent, Patent Number: 5,291,560, 1994
- [7] J. Daugman. High confidence visual recognition of persons by a test of statistical independence. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 15, No. 11, 1993