Universidad Popular del Cesar

Tratamiento de Señales

Version 2024-1

Transformada de Gabor

[Capítulo 4]

Dr. José Ramón Iglesias
DSP-ASIC BUILDER GROUP
Director Semillero TRIAC
Ingenieria Electronica
Universidad Popular del Cesar

Funciones de Gabor

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 18, NO. 8, AUGUST 1996

Texture Features for Browsing and Retrieval of Image Data

B.S. Manjunath and W.Y. Ma

Abstract—Image content based retrieval is emerging as an important research area with application to digital libraries and multimedia databases. The focus of this paper is on the image processing aspects and in particular using texture information for browsing and retrieval of large image data. We propose the use of Gabor wavelet features for texture analysis and provide a comprehensive experimental evaluation. Comparisons with other multiresolution texture features using the Brodatz texture database indicate that the Gabor features provide the best pattern retrieval accuracy. An application to browsing large air photos is illustrated.

Index Terms—Digital libraries, image database, content-based image retrieval, texture analysis, Gabor wavelets.

Funciones de Gabor

The Gabor functions are a complete (but a nonorthogonal) basis set given by:

$$f(x,y) = \frac{1}{2\pi\sigma_x\sigma_y}\exp\left(-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right)\right)\exp(2\pi j u_0 x)$$

where σ_x and σ_y denote the Gaussian envelope along the x and y-axes, and u_0 defines the radial frequency of the Gabor function.

Funciones de Gabor

self-similar filter bank can be obtained by appropriate dilation and rotation of $f(x, y)$ through the generating function

$$f_{pq}(x, y) = \alpha^{-p} f(x', y')$$

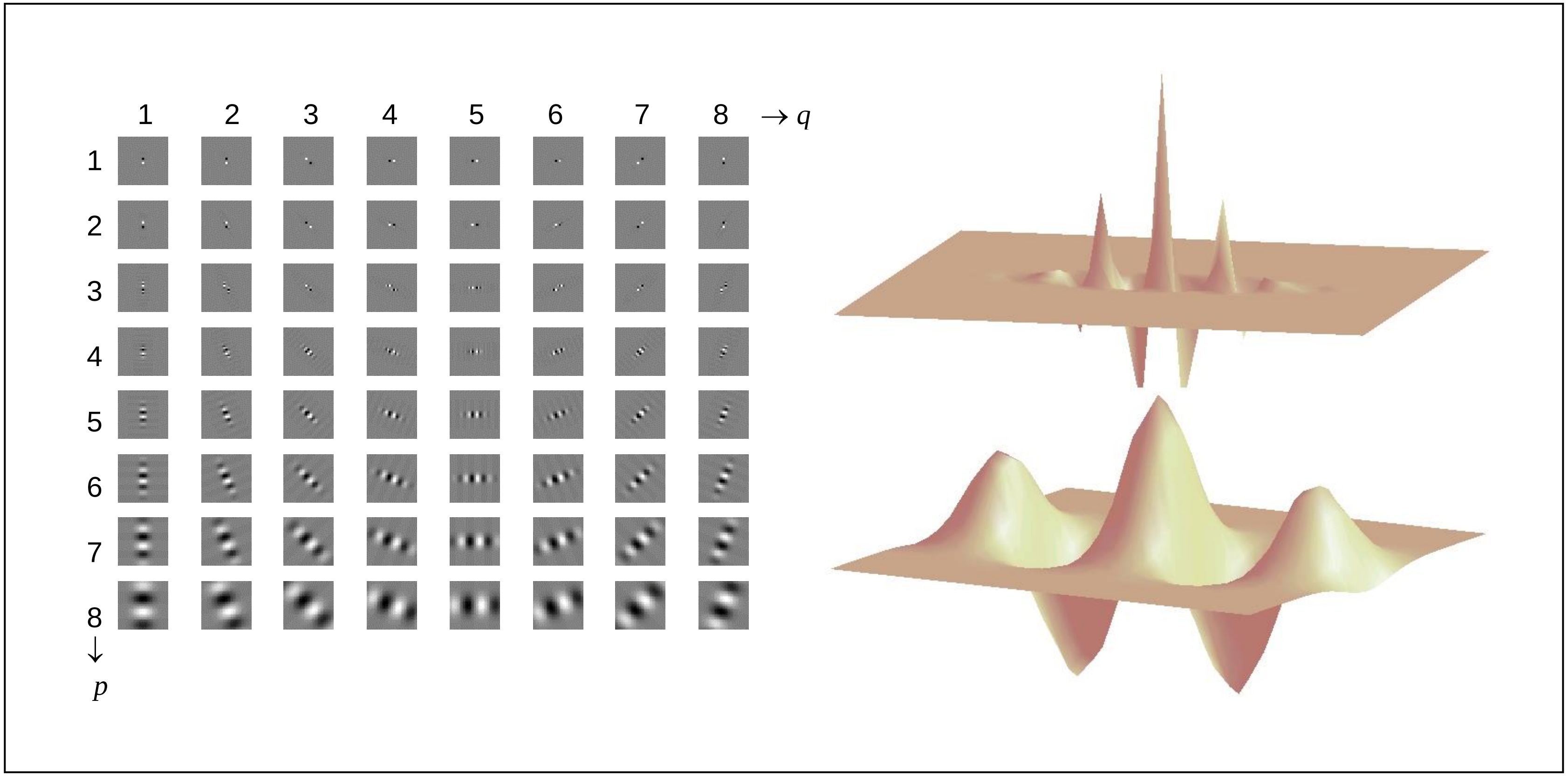
where

$$\begin{aligned} x' &= \alpha^{-p}(x \cos \theta_q + y \sin \theta_q) \\ &= \alpha^{-p}(-x \sin \theta_q + y \cos \theta_q), \\ &\quad \alpha > 1; \quad p = 1, 2, \ldots, S; \quad q = 1, 2, \ldots, L. \end{aligned}$$

The integer subscripts p and q represent the index for scale (dilation) and orientation (rotation), respectively. S is the total number of scales and L is the total number of orientations in the self-similar Gabor filter bank. For each orientation q, the angle θ_q is given by

$$\theta_q = \frac{\pi(q-1)}{L}, \qquad q = 1, 2, \ldots, L.$$

Funciones de Gabor



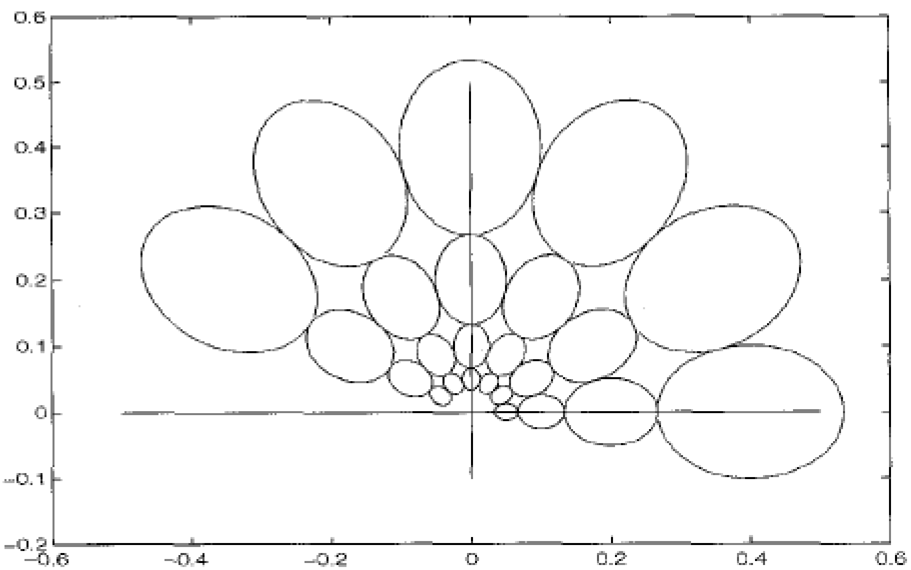
Funciones de Gabor

$$\alpha = \left(\frac{f_h}{f_l}\right)^{-(1/(S-1))}$$

$$\sigma_x = \frac{\sqrt{2\ln 2}(\alpha+1)}{2\pi f_h(\alpha-1)}$$

$$\sigma_y = \left[2\ln 2 - \left(\frac{2\ln 2}{2\pi\sigma_x f_h}\right)^2\right]^{1/2} \cdot \left[2\pi\tan\left(\frac{\pi}{2L}\right)\left(f_h - 2\ln\left(\frac{1}{4\pi^2\sigma_x^2 f_h}\right)\right)\right]^{-1}$$

Se asegura así que la respuesta en frecuencia de los filtros Gabor escogidos no se traslapen.



Funciones de Gabor

$$I_{pq}(x, y) = \Big\{[f_{pq}(x, y)_e * I(x, y)]^2 + [f_{pq}(x, y)_o * I(x, y)]^2\Big\}^{1/2} \quad (6)$$

where "$*$" denotes 2-D convolution operation, and $f_{pq}(x, y)_e$ and $f_{pq}(x, y)_o$ represent the even and odd parts of the Gabor filter separated from (3).

for p=1…S for q=1…L (S scales and L orientations)

g(p,q) = I_{pq}

J = (gmax – gmin) / gmin

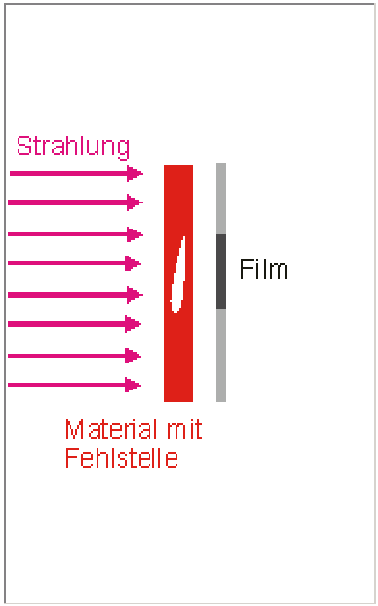
Implementación en MATLAB

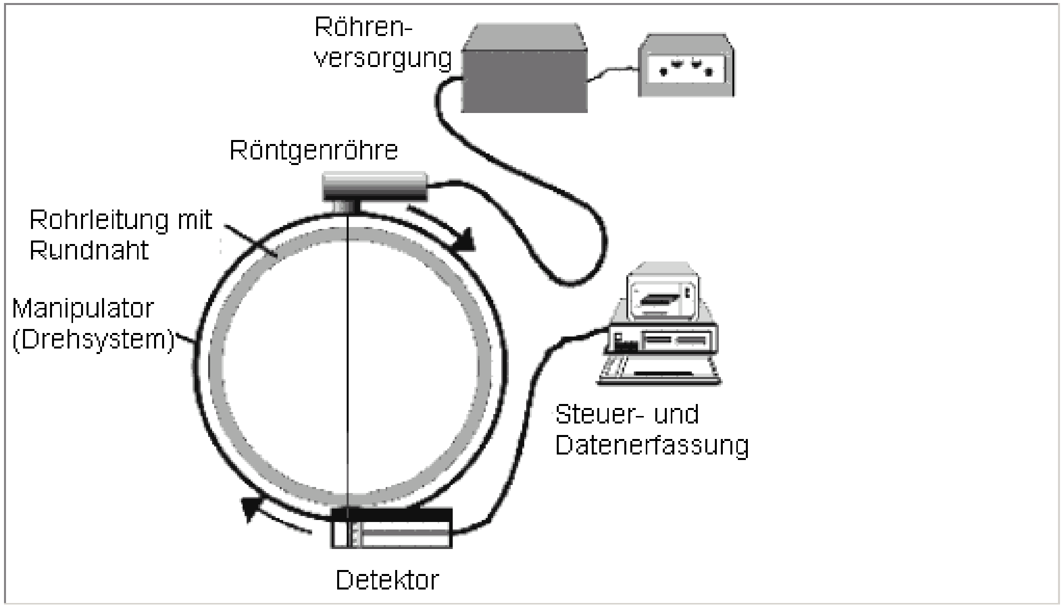
```
function gab = gaborfeatures(I,R,L,S,fh,fl,M)

alpha = (fh/fl)^(1/(S-1));
sx = sqrt(2*log(2))*(alpha+1)/2/pi/fh/(alpha-1);
sy = sqrt(2*log(2)-(2*log(2)/2/pi/sx/fh)^2)/(2*pi*tan(pi/2/L)*(fh-2*log(1/4/pi^2/sx^2/fh)));
u0 = fh;

g = zeros(S,L);
size_out = size(I)+[M M]-1;
Iw = fft2(I,size_out(1),size_out(2));
n1 = (M+1)/2;
[NN,MM] = size(I);

for p=1:S;
    for q=1:L
        f = gabor_pq(p,q,L,S,sx,sy,u0,alpha,M);
        Ir = real(ifft2(Iw.*fft2(real(f),size_out(1),size_out(2))));
        Ii = real(ifft2(Iw.*fft2(imag(f),size_out(1),size_out(2))));
        Ir = Ir(n1:n1+NN-1,n1:n1+MM-1);
        Ii = Ii(n1:n1+NN-1,n1:n1+MM-1);
        Iout = sqrt(Ir.*Ir + Ii.*Ii);
        g(p,q) = mean(Iout(k));
    end;
end
gmax = max(g(:));
gmin = min(g(:));
J = (gmax-gmin)/gmin;
gab = [g(:); gmax; gmin; J];
```

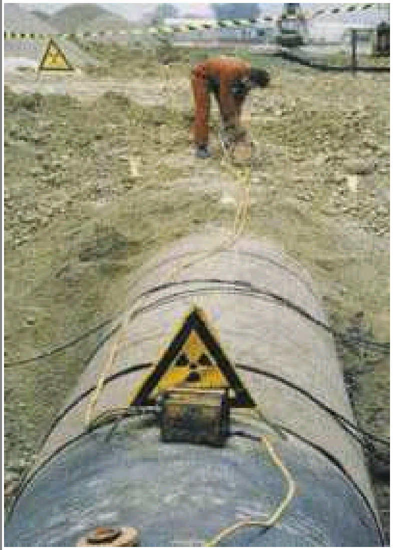
Implementación en MATLAB

```
function f = gabor_pq(p,q,L,S,sx,sy,u0,alpha,M)

f = zeros(M,M);
sx2 = sx*sx;
sy2 = sy*sy;
c = (M+1)/2;
ap = alpha^-p;
tq = pi*(q-1)/L;

f_exp = 2*pi*sqrt(-1)*u0;
for i=1:M
    x = i - c;
    for j=1:M
        y = j - c;
        x1 = ap*(x*cos(tq)+y*sin(tq));
        y1 = ap*(y*cos(tq)-x*sin(tq));
        f(i,j) = exp(-0.5*(x1*x1/sx2+y1*y1/sy2))*exp(f_exp*x1);
    end
end
f = ap*f/2/pi/sx/sy;
```

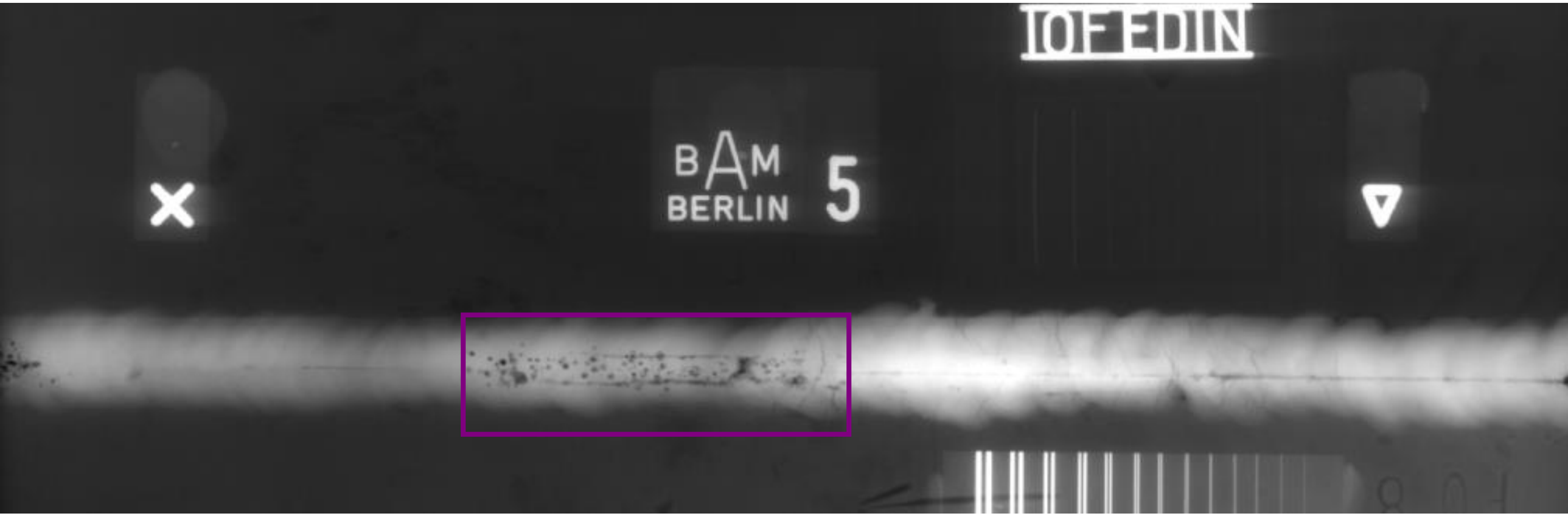
Detection of welding defects



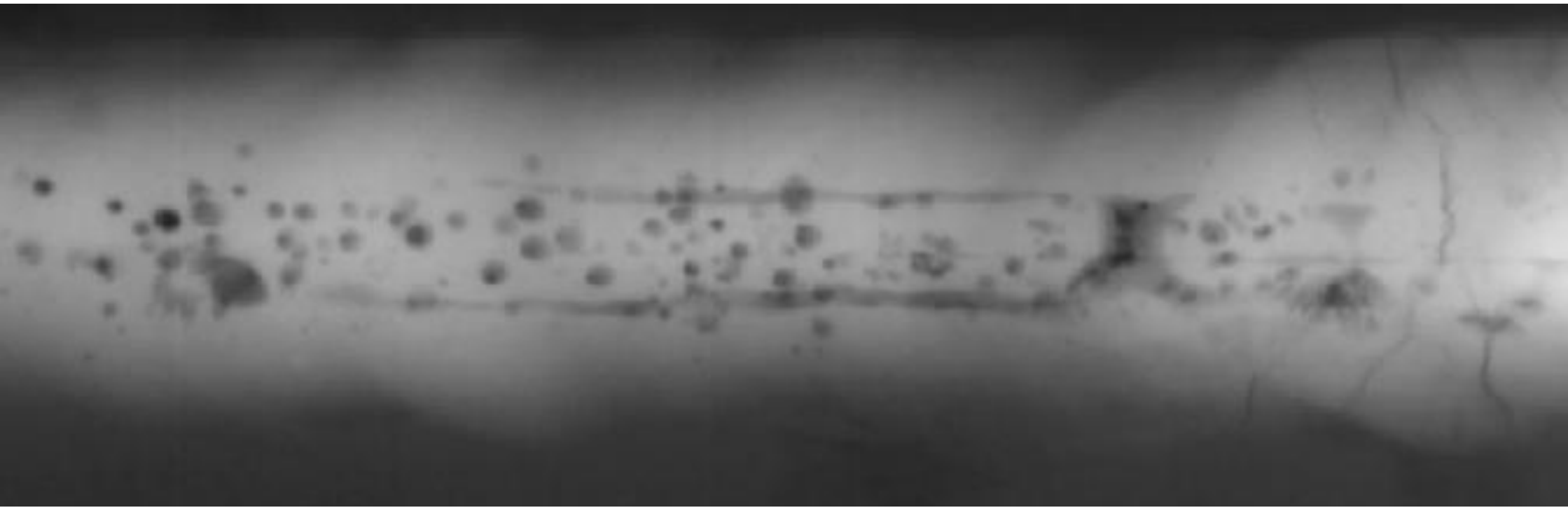




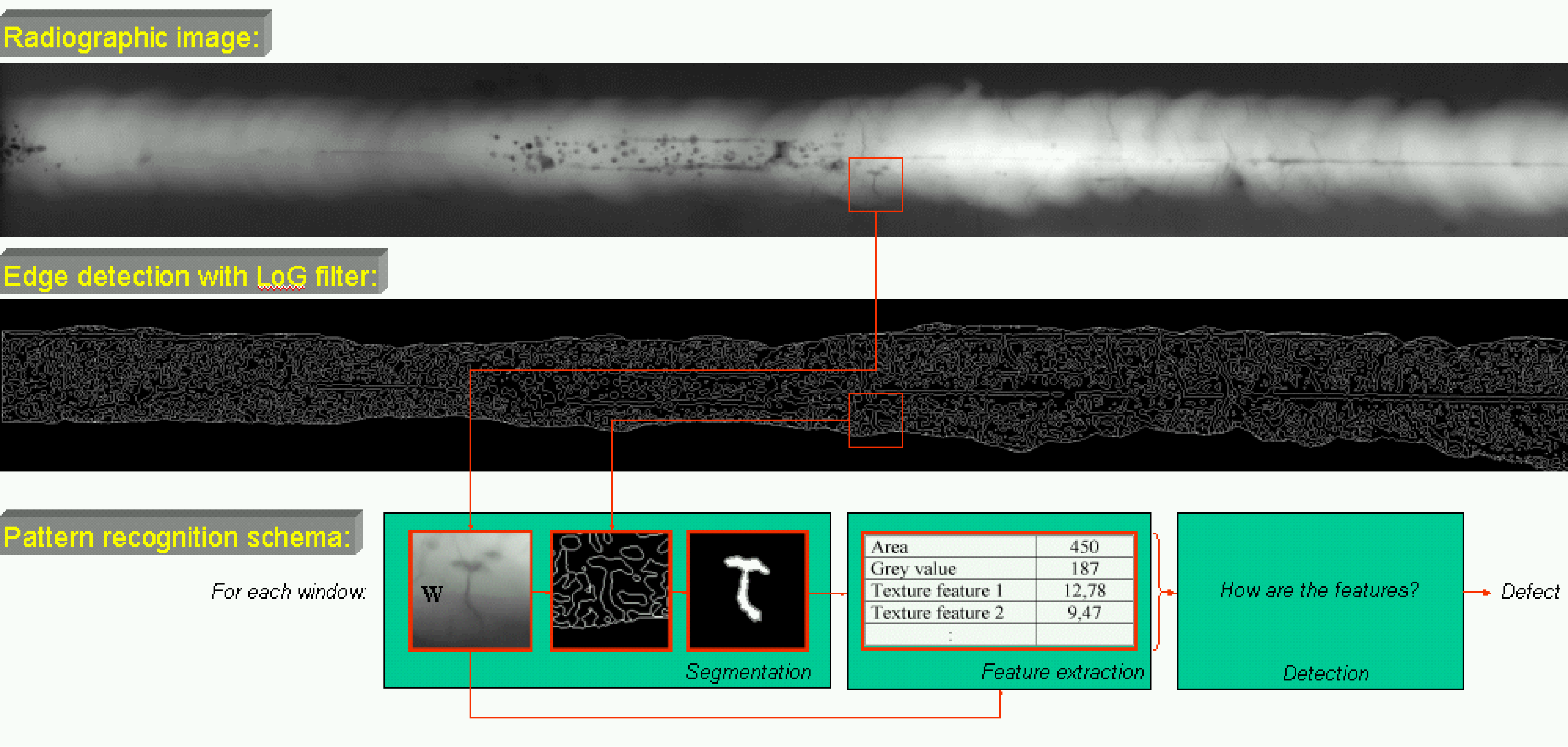
Detection of welding defects



Detection of welding defects



Detection of welding defects



Area	450
Grey value	187
Texture feature 1	12,78
Texture feature 2	9,47
:	

Detection of welding defects

Feature extraction:
-texture features (28 in 3 distances)
-Gabor features (8 scales and 8 directions)
-Crossing line profile features

Total: 158 features

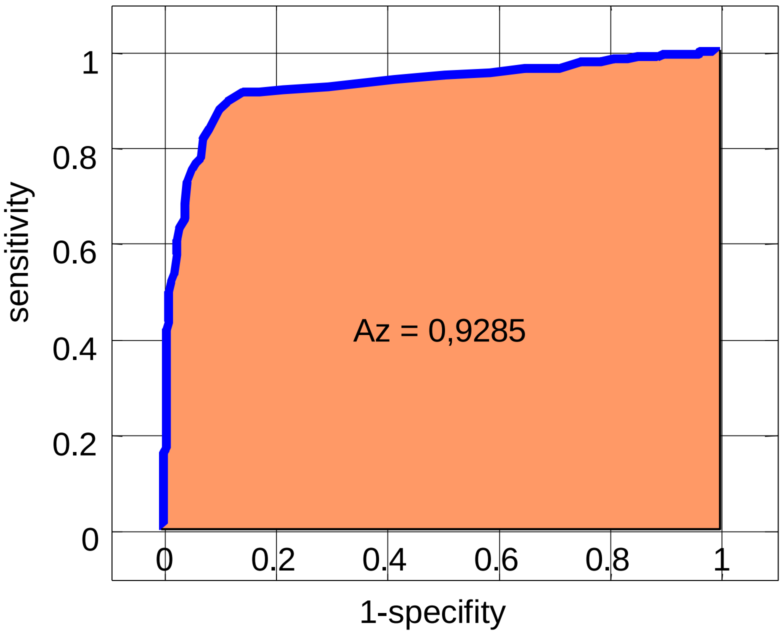
Detection of welding defects

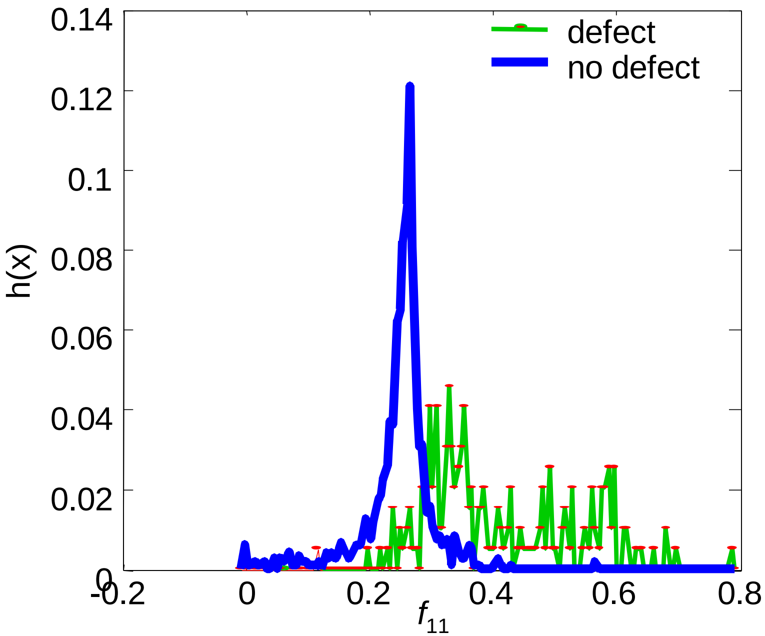
Feature	A_z
g_{63}	0.9287
f'_{11}@ d=3	0.9285
f'_{10}@ d=3	0.9207
g_{65}	0.917[illegible]
g_{67}	0.91[illegible]
f'_{11}@ d=2	0.89[illegible]
f'_{10}@ d=2	0.8620
g_{57}	0.8600
f'_{2}@ d=3	0.8523
f'_{2}@ d=2	0.8474

Feature	J
f'_{11}@ d=3	1.1376
f'_{10}@ d=3	1.0496
g_{63}	0.9132
[illegible]67	0.8997
[illegible] d=2	0.8948
[illegible] d=2	0.7638
f'_{11}@ d=1	0.6936
f'_{10}@ d=1	0.6700
g_{65}	0.6525
f'_{5}@ d=1	0.5998

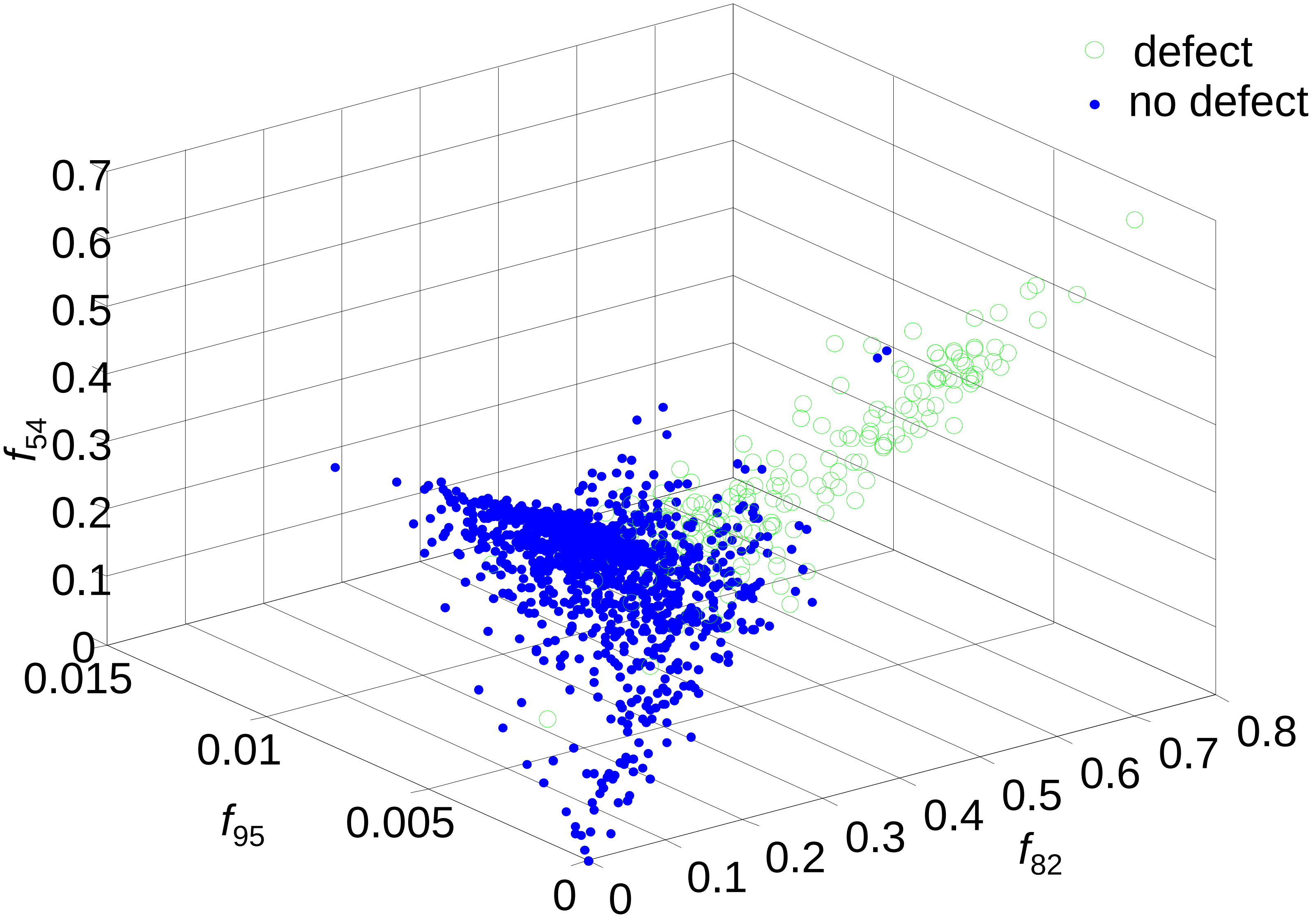
CLP
F1 : Az = 0.9376

Detection of welding defects





Detection of welding defects



Detection of welding defects

Detector	***TP***	***FP***	***FN***	***TN***	S_n	$1\text{-}S_p$
Ideal	198	0	0	1221	100,0%	0,0%
Polynomial	180	99	18	1122	90,91%	8,11%
Mahalanobis	180	155	18	1066	90,91%	12,69%
nearest neighbour	157	168	41	1053	79,29%	13,75%