**Sayan Banerjee**

**UWM**

EE:890 Mini-project\_6 report

# 

# **Independent Component Analysis (Fast ICA)**

clear all;

clc;

x1=imread('Recording-1.png');

x2=imread('Recording-2.png');

x1=double(reshape(x1,1,453600));

x2=double(reshape(x2,1,453600));

N = 453600;

x=[x1;x2];

avg = mean(x,2);

x(1,:) = x(1,:) - avg(1);

x(2,:) = x(2,:) - avg(2);

ce = cov(x');

[E D E\_T]=svd(ce);

white = E \* D^(-1/2) \* E';

x\_white = white \* x;

cm = cov(x\_white');

W = fastica(x\_white);

s1 = W(:,1)'\*x\_white;

s2 = W(:,2)'\*x\_white;

pic1=reshape(s1,567,800);

pic2=reshape(s2,567,800);

subplot(1,2,1);

imshow(pic1);

subplot(1,2,2);

imshow(pic2);

function W = fastica(x)

W = rand(size(x,1), size(x,1));

lw = zeros(size(W,1),1);

p = 0;

while 1

W(:,p+1) = negentropy(x,W(:,p+1));

wsum = zeros(size(W,1),1);

if p >= 1

for j = 1:p

wsum = wsum + W(:,p+1)'\*W(:,j)\*W(:,j);

end

W(:,p+1) = W(:,p+1) - wsum;

end

W(:,p+1) = W(:,p+1) / norm(W(:,p+1));

if (1 - abs(dot(W(:,p+1), lw))) < 0.000000000001

p = p+1;

if p+1 > size(W,2)

break;

end

else

lw = W(:,p+1);

end

end

end

function w = negentropy(x, w)

tmp1 = zeros(size(x));

tmp2 = zeros(1, size(x, 2));

for j = 1:size(x, 2)

tmp1(:,j) = x(:,j)\*g1((w'\*x(:,j)));

tmp2(:,j) = g1d(w'\*x(:,j));

end

w = mean(tmp1,2) - mean(tmp2,2)\*w;

end

function y = g1(u)

a1 = 1;

y = tanh(a1\*u);

end

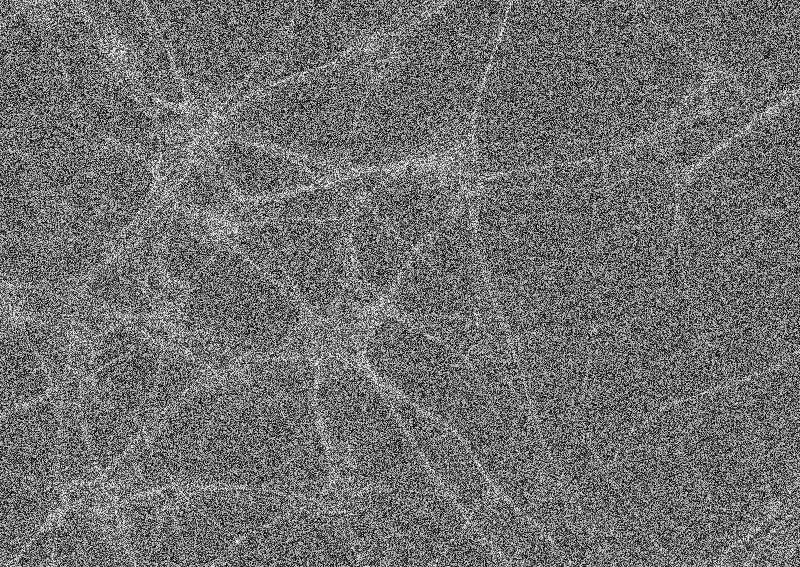
function y = g1d(u)

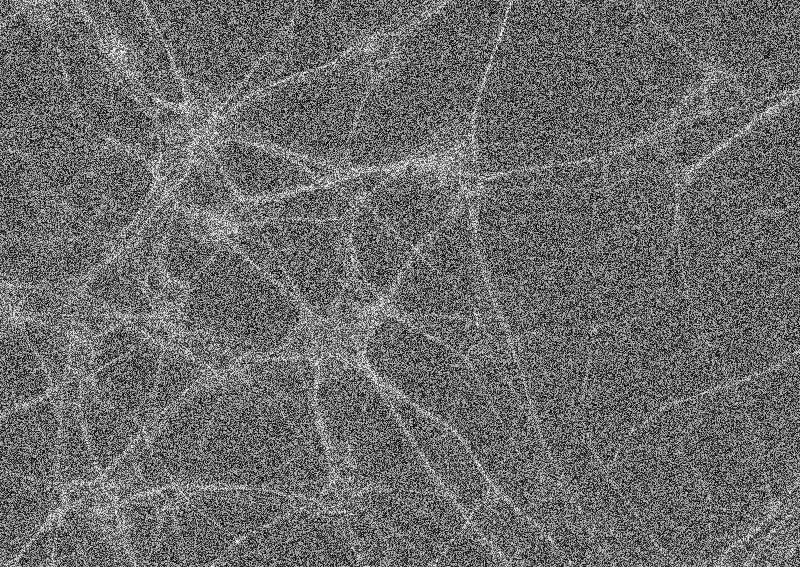
a1 = 1;

y = a1\*(1-tanh(a1\*u)^2);

end

**Inputs:**





**Outputs:**

