beeld=input('Which image do you want to be outlined (1-264)?');

beginbeeld=17;

eindbeeld=33;

%hier kun je gewoon de gewenste range opgeven

if beeld < 10

         name=['000' num2str(beeld)];

    elseif beeld >=10 && beeld<100

         name=['00' num2str(beeld)];

    else name=['0' num2str(beeld)];

end

%het vinden van de tafelrand

totedges=zeros(410,410,141);

for b=beginbeeld:eindbeeld

    if mod(b,30)==0

        disp(b)

    end

    if b >=10 && b<100

         naam=['00' num2str(b)];

    else naam=['0' num2str(b)];

    end

    load(['img' naam '.mat'])

    a=double(image);

    a=min(a,mean(a(:))+2.5\*std(a(:)));

    a=uint8(a\*255/max(a(:)));

    totedges(:,:,b-beginbeeld+1)=edge(a,'sobel',[]);

end

%het originel dicom beeld wordt geladen. Er zijn hier al uint16s van

%gemaakt.

load(['img' name '.mat'])

image=double(image);

ORIGimage=image;

realimage=sort(image(:));

%de buitenste cirkel wordt weggehaald voor het maken van de histogram

lb=find(realimage>round(sum(sum(image(1:10,1:10)))/100),1,'first');

image=realimage(lb:length(realimage));

%de outliers naar boven worden eruit gehaald

image=min(image,mean(image)+2.5\*std(image));

[histogram,centers]=hist(image);

figure,subplot(1,2,1)

%het tekenen van de histogram zelf

bar(centers,histogram), hold on

title(['Image ' name])

axis([min(image) max(image) 0 max(histogram)])

histogram=[0 histogram 0];

%het vinden van de twee pieken

topvalue='f';

for t=2:length(histogram)-1

    if histogram(t-1)<histogram(t) && histogram(t)>histogram(t+1)

        if topvalue=='f'

            fpos=t-1;

            topvalue='l';

        else lpos=t-1;

        end

    end

end

%het plotten van de rode lijntjes

plot([centers(fpos);centers(fpos)],[0;max(histogram)],'r:')

plot([centers(lpos);centers(lpos)],[0;max(histogram)],'r:')

%HET BEPALEN VAN DE THRESHOLD

threshold=(centers(fpos)+centers(lpos))/2;

a=ORIGimage;

a=min(a,mean(a(:))+2.5\*std(a(:)));

a=uint8(a\*255/max(a(:)));

edges=edge(a,'sobel',[]);

v=sum(sum(totedges,3),2);

%figure,plot(v,'k.')

tableedge=find(v==max(v),1,'first');

topedge=tableedge;%die '9' hebben we experimenteel gevonden.

edges(topedge:size(ORIGimage,1))=0;

numedges=bwlabel(edges,8);

mask=ORIGimage>threshold;

%mask2=mask;

mask(topedge:size(mask,1),:)=0;

mask=imfill(mask,4,'holes');

%bepalen van de outercontour

body=bwperim(mask);

%opvullen eventuele losse puntjes of eilandjes, zodat je echt een solide rand krijgt

%Ik denk inderdaad dat het voldoende is te kijken of de horizontale lijn

%die er al ligt lang genoeg is, en je dan gewoon de hele lijn moet trekken

opvuller=mask-body;

[regions,n\_o\_regions]=bwlabel(opvuller,4);

regionsizes=zeros(n\_o\_regions,1);

for t=1:n\_o\_regions

    regionsizes(t)=sum(sum(regions==t));

end

for ver=1:size(body,1)

    for hor=1:size(body,2)

        if regions(ver,hor)~=0 && regions(ver,hor)~=find(regionsizes==max(regionsizes),1,'first')

            body(ver,hor)=1;

        end

    end

end

%alleen randpixels overlaten die echt aan het binnengebied grenzen.

body2=imfill(body,4,'holes');

body2=body2-body;

body3=zeros(size(body,1)+2,size(body,2)+2);

body3(2:size(body,1)+1,2:size(body,2)+1)=body2;

for ver=1:size(body,1)

    for hor=1:size(body,2)

        if body(ver,hor)==1

            if sum(body3(ver:ver+2,hor+1))+sum(body3(ver+1,hor:hor+2))==0

                body(ver,hor)=0;

            end

        end

    end

end

%variabelen die de tweede keer anders moeten:

for innerouter=1:1

    %stukken sobelcontour vinden die autercontour echt doorkruisen

    if innerouter==1

        mask2=imfill(body,4,'holes');

    else mask2=imfill(body,[1 1],4);

    end

    numedges=numedges.\*mask2;

    whichcontours=unique(numedges.\*body);

    crossedges=zeros(size(edges,1),size(edges,2));

    for t=2:length(whichcontours)

        crossedges=crossedges|(numedges==whichcontours(t));

    end

    combcontour=body+2\*crossedges;

    %alle uiteinden bepalen

    %eerst alle stukken sobelcontour die precies op outercontour liggen

    %(dwz geen enkele 2 in de buurt hebben)

    %verwijderen,

    combcontour2=zeros(2+size(combcontour,1),2+size(combcontour,2));

    combcontour2(2:1+size(combcontour,1),2:1+size(combcontour,2))=combcontour;

    for ver=1:size(combcontour,1)

        for hor=1:size(combcontour,2)

            if combcontour(ver,hor)==3

                if max(max(combcontour2(ver:ver+2,hor:hor+2)==2))==0

                    combcontour(ver,hor)=1;

                    combcontour2(ver+1,hor+1)=1;

                end

            end

        end

    end

    %nu een uiteinde maken aan alle sobelstukken

    %merk op dat de lijnen gladjes verondersteld worden, dus dat dit volgens

    %mij niet lukt met alleen de horizontale sobeloperator!

    clockwise=[-1 -1;-1 0;-1 1;0 1;1 1;1 0;1 -1;0 -1];

    for ver=1:size(combcontour,1)

        for hor=1:size(combcontour,2)

            if combcontour(ver,hor)==3

                %nu hebben we het beginpunt van een of meer contourstukken te pakken en

                %gaan we het lijnstuk af op zoek naar eindpunten

                opweg=1;

                while opweg==1

                    v=ver;h=hor;

                    t=1;

                    flag=0;

                    opweg=0;

                    while flag==0

                        if combcontour2(v+1+clockwise(t,1),h+1+clockwise(t,2))==2

                            combcontour(v+clockwise(t,1),h+clockwise(t,2))=4;

                            combcontour2(v+1+clockwise(t,1),h+1+clockwise(t,2))=4;

                            if opweg==0

                                combcontour(v,h)=7;

                                combcontour2(v+1,h+1)=7;

                            end

                            v=v+clockwise(t,1);h=h+clockwise(t,2);

                            t=1;

                            opweg=1;

                        elseif combcontour2(v+1+clockwise(t,1),h+1+clockwise(t,2))==3 && opweg==1

                            combcontour(v+clockwise(t,1),h+clockwise(t,2))=6;

                            combcontour2(v+1+clockwise(t,1),h+1+clockwise(t,2))=6;

                            flag=1;

                        else t=t+1;

                        end

                        if t==9

                            if opweg==1

                                combcontour(v,h)=5;

                                combcontour2(v+1,h+1)=5;

                            end

                            flag=1;

                        end

                    end

                end

            end

        end

    end

    %nu voor ieder uiteinde het dichtstsbijzijnde punt bepalen en die verbinden

    [vu,hu]=find(combcontour==5);

    [va,ha]=find(combcontour==1);

    pairs=zeros(length(vu),4);pairs(:,1)=vu;pairs(:,2)=hu;

    for t=1:length(vu)

        vtot=(cat(1,vu,va)-vu(t)).\*(cat(1,vu,va)-vu(t))+(cat(1,hu,ha)-hu(t)).\*(cat(1,hu,ha)-hu(t));

        vtot=vtot+max(vtot)\*(vtot==0);

        index=find(vtot==min(vtot),1,'first');

        if index>length(vu)

            pairs(t,3)=va(index-length(vu));

            pairs(t,4)=ha(index-length(vu));

        else pairs(t,3)=vu(index);

            pairs(t,4)=hu(index);

        end

    end

    %teken rechte lijnen

    for t=1:length(vu)

        for lijn=0:100

            v=floor((pairs(t,3)-pairs(t,1))\*lijn/100+pairs(t,1));

            h=floor((pairs(t,4)-pairs(t,2))\*lijn/100+pairs(t,2));

            combcontour(v,h)=4;

            combcontour2(v+1,h+1)=4;

        end

    end

    %alleen randpixels overlaten die echt aan het binnengebied grenzen.

    if innerouter==1

        tussen=combcontour>0;

        innerregion=xor(combcontour>0,imfill(combcontour>0,[round(size(combcontour,1)/2) round(size(combcontour,2)/2)],4));

    else innerregion=xor(combcontour>0,imfill(combcontour>0,[1 1],4));

        tussen=combcontour>0;

    end

    innerregion2=zeros(size(combcontour,1)+2,size(combcontour,2)+2);

    innerregion2(2:size(body,1)+1,2:size(body,2)+1)=innerregion;

    for ver=1:size(combcontour,1)

        for hor=1:size(combcontour,2)

            if combcontour(ver,hor)>0

                if sum(innerregion2(ver+1,hor:hor+2))+sum(innerregion2(ver:ver+2,hor+1))==0

                    combcontour(ver,hor)=0;

                end

            end

        end

    end

    %nu de hele rimram nog eens maar dan voor sobelcontourstukken die buiten de autercontour vallen

    %body=combcontour>0;

end

%overgebleven contour op originele beeld plakken

a=ORIGimage;

a=min(a,mean(a(:))+2.5\*std(a(:)));

hold off

subplot(1,2,2)

%tekenen van de outercontour op beeld

imshow(uint8(a\*255/max(a(:))+double(combcontour>0)\*255))

%imshow(uint8(a\*255/max(a(:))+double(edges)\*255))

%imshow(edges)

title(['Image ' name])

c=uint8(a\*255/max(a(:)));c(tableedge,:)=255;figure,imshow(c)