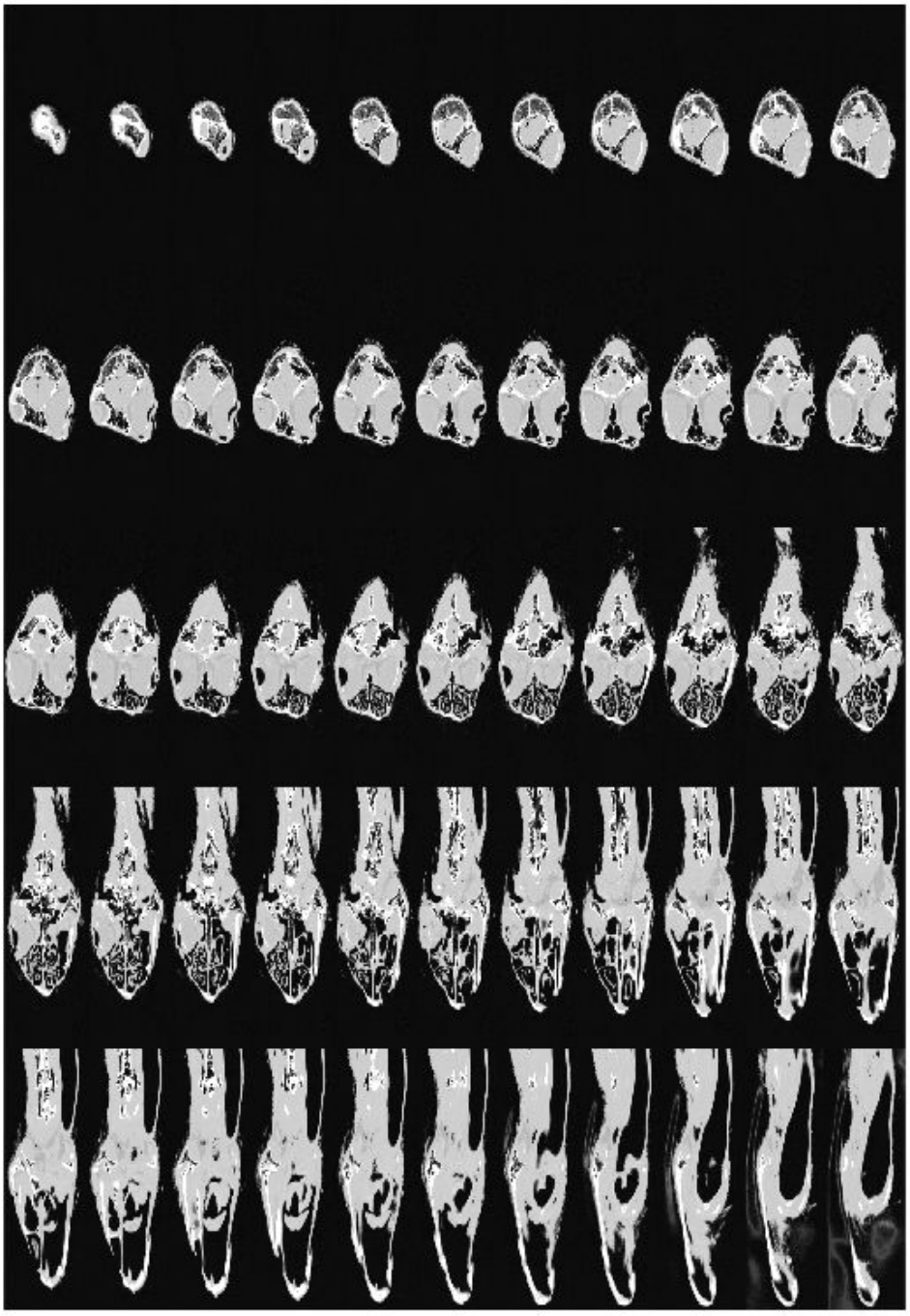
Přímé zobrazení objemu

Načtení dat

```
fid = fopen('Struthio_8.dat');

% Columns=446
% Rows=140
% Start=1
% End=128
% Bits=8
% SpacingX=0.625
% SpacingY=0.78125
% SpacingZ=0.78125
% SkinExtr=18
% BoneExtr=219

data = uint8(zeros([446,140,128]));
for i = 1 : 128
    data(:,:,i) = uint8(fread(fid,[446,140],'uint8'));
end
```

Zobrazení série snímků

```
%imshow(data(:,:,5));
obrazek = uint8(zeros([4*446,10*140]));
index = 10;
for i = 0 : 4
    for j = 0 : 10
        obrazek(i*446 + 1 : ((i+1)*446),j*140 + 1 : ((j+1)*140) ) = data(:,:,index);
        index = index+2;
    end
end

figure, imshow(obrazek);
```



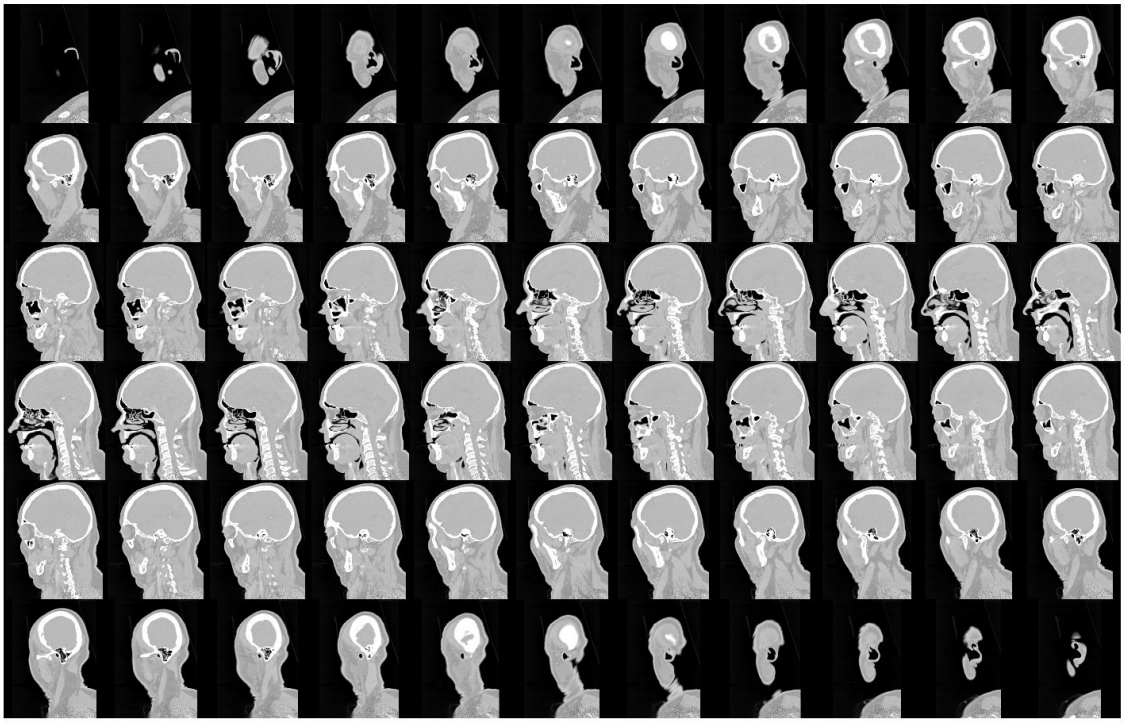
```
%imwrite(obrazek,'ct_pstros.png');
```

```
fid = fopen('hlava.dat');

% Columns=224
% Rows=256
```

```
% Start=1
% End=300
% Bits=8
% SpacingX=0.888
% SpacingY=0.888
% SpacingZ=0.900
% SkinExtr=50
% BoneExtr=220


data = uint8(zeros([224,256,300]));
for i = 1 : 300
    data(:,:,i) = uint8(fread(fid,[224,256],'uint8'));
end


data2 = permute(data,[3,2,1]);
[m,n,o] = size(data2);
```

Zobrazení série snímků

```
%imshow(data(:,:,5));

obrazek = uint8(zeros([5*m,10*n]));
index = 10;
for i = 0 : 5
    for j = 0 : 10
        obrazek(i*m + 1 : ((i+1)*m),j*n + 1 : ((j+1)*n) ) = data2(:,:,index);
        index = index+3;
    end
end

figure, imshow(obrazek);
```



```
%imwrite(obrazek,'ct_hlava.png');
```

Metody nehledající povrch

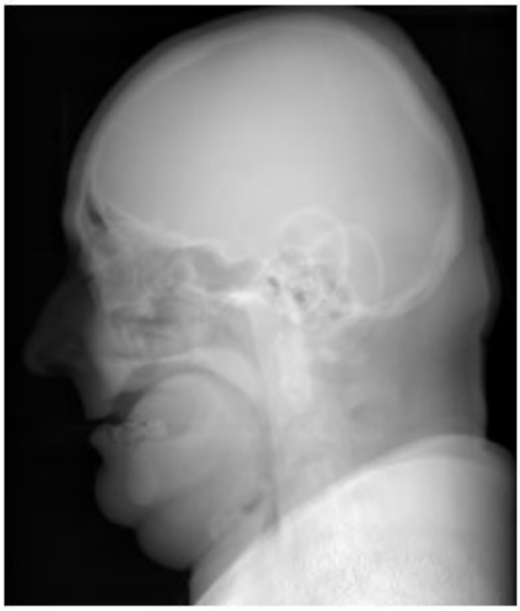
Maximum intensity projection

```
I1 = max(data2,[],3);

figure, imshow(I1);
```

```
%imwrite(I1,'volume_mip.png');
```

Summed intensity projection

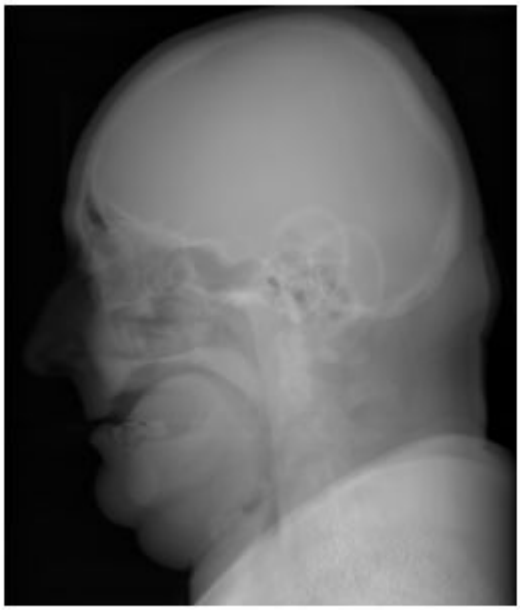
```
I2 = sum(double(data2),3);

figure, imshow(I2,[]);
```



```
%f = getframe;
%imwrite(f.cdata, 'volume_sum.png')
```

Average intensity projection

```
I3 = uint8(mean(data2,3));

figure, imshow(I3);
```

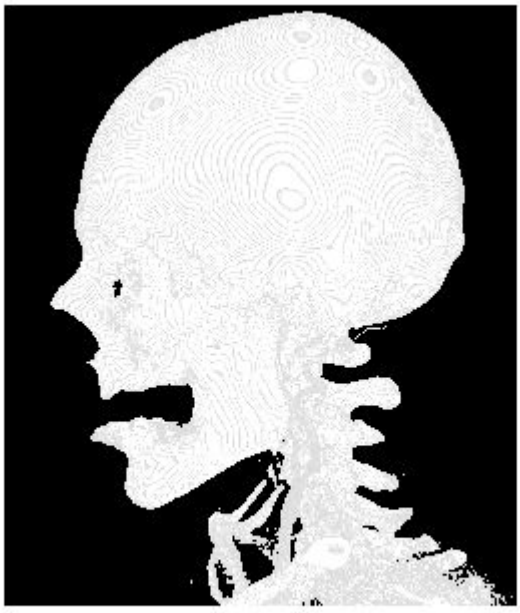


```
imwrite(I3,'volume_avg.png');
```

Metody hledající povrch bez normál

Voxel value projection

```
T = 220;
I4 = uint8(zeros([m,n]));

for i = 1 : m
    for j = 1 : n
        for l = 1 : o
            if(data2(i,j,l) >= T)
                I4(i,j) = data2(i,j,l);
                break;
            end
        end
    end
end

figure, imshow(I4);
```



```
imwrite(I4,'volume_vvp.png');
```

Z-buffer projection

```
T = 220;
I5 = uint8(zeros([m,n]));

for i = 1 : m
    for j = 1 : n
        for l = 1 : o
            if(data2(i,j,l) >= T)
                I5(i,j) = l;
                break;
            end
        end
    end
end

figure, imshow(I5);
```

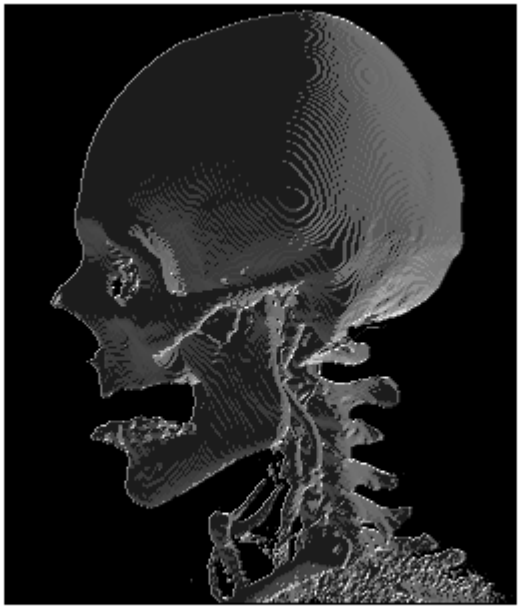
```
imwrite(I5,'volume_zbuf.png');
```

Metody hledající povrch + normály

Z-buffer gradient

```
T = 220;
Z = uint8(zeros([m,n]));

I6 = uint8(zeros([m,n]));
L = [50,100,10];
L = L/norm(L); % vektor ukazujici smer ke svetlu
v = [10,0,1]; %smer pohledu
v = v/norm(v);
ra= 20;
rd = 100;
rs = 7;

h = 3;

for i = 1 : m
    for j = 1 : n
        for l = 1 : o
            if(data2(i,j,l) >= T)
                Z(i,j) = l;
                break;
            end
        end
    end
```

```
end

N = zeros([m,n,3]);

for i = 2 : m-1
    for j = 2 : n-1
        if(Z(i,j)>0)
            n1 = double(Z(i+1,j) - Z(i-1,j));
            n2 = double(Z(i,j+1) - Z(i,j-1));
            n3 = 1;
            n1 = n1/norm([n1, n2, n3]);
            n2 = n2/norm([n1, n2, n3]);
            n3= n3/norm([n1, n2, n3]);

            r = 2*dot(L,[n1,n2,n3])*[n1,n2,n3] - L;
            %difuzni ambientni a spekulani slozka
            I6(i,j) = rd*dot(L,[n1,n2,n3]) + ra + rs*(dot(v,r)^h);
        end
    end
end

figure, imshow(I6);
```

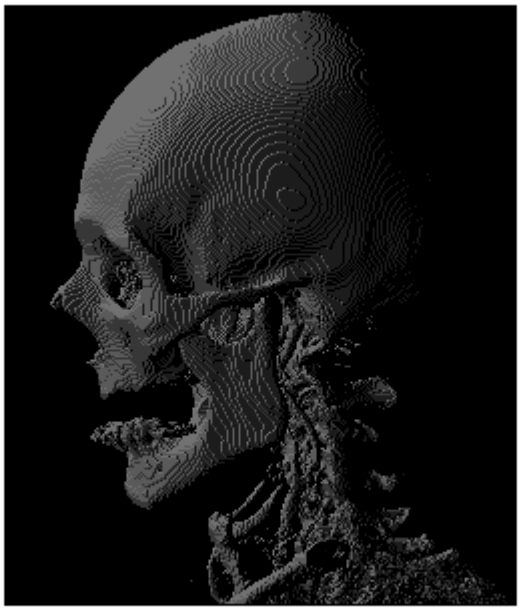


```
imwrite(I6,'volume_zgs.png');
```

Z-buffer gradient

```
T = 220;
Z = uint8(zeros([m,n]));
```

```
I7 = uint8(zeros([m,n]));
L = [50,100,10];
L = L/norm(L); % vektor ukazujici smer ke svetlu
v = [10,0,1]; %smer pohledu
v = v/norm(v);
ra= 20;
rd = 100;
rs = 7;

h = 3;

B = data2 >= T;

for i = 1 : m
    for j = 1 : n
        for l = 2 : o
            if(data2(i,j,l) >= T)
                Z(i,j) = l;
                break;
            end
        end
    end
end

N = zeros([m,n,3]);

for i = 2 : m-1
    for j = 2 : n-1
        if(Z(i,j)>0)
            l = Z(i,j);
            n1 = double(B(i+1,j,l) - B(i-1,j,l));
            n2 = double(B(i,j+1,l) - B(i,j-1,l));
            n3 = double(B(i,j,l+1) - B(i,j,l-1));
            n1 = n1/norm([n1, n2, n3]);
            n2 = n2/norm([n1, n2, n3]);
            n3= n3/norm([n1, n2, n3]);

            r = 2*dot(L,[n1,n2,n3])*[n1,n2,n3] - L;
            %difuzni ambientni a spekulani slozka
            I7(i,j) = rd*dot(L,[n1,n2,n3]) + ra + rs*(dot(v,r)^h);
        end
    end
end

figure, imshow(I7);
```

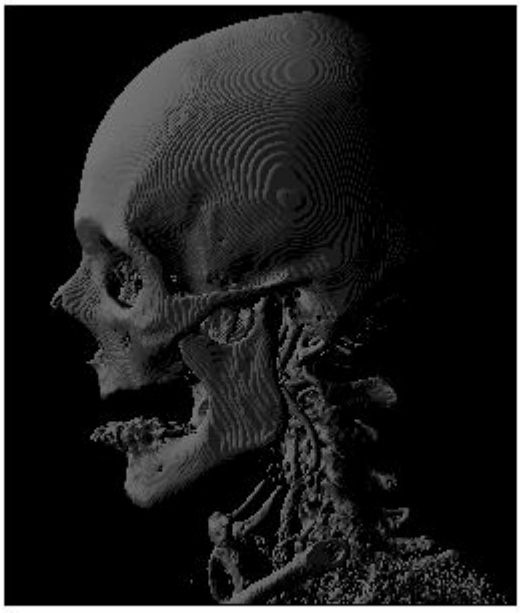


```
imwrite(I7,'volume_vgs.png');
```

gray-level gradient

```
T = 220;
Z = uint8(zeros([m,n]));

I8 = uint8(zeros([m,n]));
L = [50,100,10];
L = L/norm(L); % vektor ukazujici smer ke svetlu
v = [10,0,1]; %smer pohledu
v = v/norm(v);
ra= 20;
rd = 100;
rs = 7;

h = 3;

B = data2 >= T;

for i = 1 : m
    for j = 1 : n
        for l = 2 : o
            if(data2(i,j,l) > T)
                Z(i,j) = l;
                break;
            end
        end
    end
```

```
end

for i = 2 : m-1
    for j = 2 : n-1
        if(Z(i,j)>0)
            l = Z(i,j);
            n1 = double(data2(i+1,j,l)) - double(data2(i-1,j,l));
            n2 = double(data2(i,j+1,l)) - double(data2(i,j-1,l));
            if(l==1 || l==o)
                n3=0;
            else
                n3 = double(data2(i,j,l+1)) - double(data2(i,j,l-1));
            end
            if(n1~=0 || n2~=0 || n3~=0 )
                norma = norm([n1, n2, n3]);
                 n1 = n1/norma;
                 n2 = n2/norma;
                 n3= n3/norma;
            end

            r = 2*dot(L,[n1,n2,n3])*[n1,n2,n3] - L;
            r = r/norm(r);
            %difuzni ambientni a spekulani slozka
            I8(i,j) = rd*dot(L,[n1,n2,n3]) + ra + rs*(dot(v,r)^h);
        end
    end
end

figure, imshow(I8);
```



```
imwrite(I8,'volume_gls.png');
```