Příloha A

Počítačový program IFTAmaster

```
function IFTAmaster(path, file_in, ER_if, AA_if, alpha, max_iter_if, max_iter, ZB_if, ZB )

cesta1 = [path file_in];
V=imread(cesta1); %nacte pozadovany design
sz=size(V); %velikost vstupni osvetlovaci matice
B_re(1:sz(2), 1:sz(1))=double(256);
B_im(1:sz(2), 1:sz(1))=pi*rand(sz(1));
B=complex(B_re,B_im);
Fa_ra(1:sz(2),1:sz(1))=pi;
C_A(1:sz(2),1:sz(1))=double(0); %prealokace matice amplitud
C_A_narm(1:sz(2),1:sz(1))=double(0); %prealokace matice amplitud
C_f(1:sz(2),1:sz(1))=double(0); %prealokace matice fazi
C_P_norm(1:sz(2),1:sz(1))=double(0);
BB_fa(1:sz(2),1:sz(1))=double(0);
BB_v(1:sz(2),1:sz(1))=double(0);
CC(1:sz(2),1:sz(1))=double(0);
RMS(1:sz(2),1:sz(1))=double(0);
V_norm_sum(1:sz(2),1:sz(1))=double8(0);
BB_v_write(1:sz(2),1:sz(1))=uint8(0);
V_norm(1:sz(2),1:sz(1))=double(0);
D(1:sz(2), 1:sz(1))=double(0); % prealokace matice chyb
max_V=max(max(V)); %urceni maximalni intenzity
min_V=min(min(V)); %urceni minimalni intenzity
if ZB ==1              %hledani minimalniho oramovani motivu
  bo_leva=sz(2);
  bo_prava=1;
  bo_horni=sz(1);
  bo_dolni=1;
  for m=1:sz(2);
     for n=1:sz(1);
       if V(m,n)==0
         if m<bo_horni
           bo_horni=m;
         end;
         if m>bo_dolni
           bo_dolni=m;
         end;
         if n<bo_leva
           bo_leva=n;
         end;
         if n>bo_prava
           bo_prava=n;
         end;
       end;
     end;
  end;
```

```
end;
if max_iter_if==1
   for x=1:max_iter %pro prvni az maximalni iteraci
      C=fft2(B); %FFT matice B
      C=fftshift(C); % zamena stredu za periferie
      C_f=angle(C); %hodnoty faze
      C_A=abs(C);  % hodnoty amplitudy
      C_A_real=real(C_A);
      C_A_imag=imag(C_A);
      C_P=C_A_real.^2 + C_A_imag.^2;   %power spektrum~intenzita
      max_C_P=max(max(C_P));  %urceni maximalni amplitudy
      min_C_P=min(min(C_P));  %urceni minimalni amplitudy
      roz= max_C_P-min_C_P;   %rozsah amplitud
      max_C_f=max(max(C_f));  %urceni maximalni faze
      min_C_f=min(min(C_f));  %urceni minimalni faze
      if ZB_if==1
        for m=max(max(bo_horni-ZB,1)):min(min(bo_dolni+ZB,sz(1)));
            for n=max(max(bo_leva-ZB,1)):min(min(bo_prava+ZB, sz(2)));
               C_P_norm(m,n)=(255*(C_P(m,n)-min_C_P)/(max_C_P-min_C_P));
               V_norm(m,n)=(255*(V(m,n)-min_V)/(max_V-min_V));
               if ER_if==1
                 V_norm(m,n)=(255*(V(m,n)-min_V)/(max_V-min_V));
               end;
               if AA_if==1
                  V_norm(m,n)=alpha*V(m,n)+(1-alpha)*C_P_norm(m,n);
               end;
               CC(m,n)=sqrt(V_norm(m,n))*exp(1i*angle(C(m,n)));
               RMS(m,n)=((C_P_norm(m,n)-V_norm(m,n))^2);
               %zapisovani
            end;
        end;
        V_norm_sum=sum(sum(V_norm));
        RMS_disp=sum(sum(RMS));
        RMS_disp=RMS_disp/V_norm_sum;
      else
         for m=1:sz(2);
            for n=1 : sz(1);
               C_P_norm(m,n)=(255*(C_P(m,n)-min_C_P)/(max_C_P-min_C_P));
               V_norm(m,n)=(255*(V(m,n)-min_V)/(max_V-min_V));
               if ER_if==1
                 V_norm(m,n)=(255*(V(m,n)-min_V)/(max_V-min_V));
               end;
               if AA_if==1
                  V_norm(m,n)=alpha*V(m,n)+(1-alpha)*C_P_norm(m,n);
               end;
               CC(m,n)=sqrt(V_norm(m,n))*exp(1i*angle(C(m,n)));
               RMS(m,n)=((C_P_norm(m,n)-V_norm(m,n))^2);
               %zapisovani
            end;
         end;
```

```
            V_norm_sum=sum(sum(V_norm));
            RMS_disp=sum(sum(RMS));
            RMS_disp=RMS_disp/V_norm_sum;
        end;
        %disp (RMS_disp);
        BB=ifft2(CC);
        BB=fftshift(BB);
        BB_fa=angle(BB);
        max_BB_fa=max(max(BB_fa));  %urceni maximalni amplitudy
        min_BB_fa=min(min(BB_fa));  %urceni minimalni amplitudy
        %disp (max_BB_fa)
        %disp (min_BB_fa)
        BB_fa_abs=sqrt(BB_fa.^2);
        for m= 1 : sz(2);
            for n=1 : sz(1);
                if BB_fa_abs(m,n)<=pi/2
                    BB_v_write(m,n)=0;
                    BB_v(m,n)=0;
                else
                    BB_v_write(m,n)=255;
                    BB_v(m,n)=pi;
                end;
            end;
        end;
        %figure
        %imshow (BB_v_write);
        B(:,:)=double(0);
        C(:,:)=double(0);
        B_fa(:,:)=double(0);
        for m= 1 : sz(2);
            for n=1 : sz(1);
                B(m,n)=255*exp(1i*BB_v(m,n));
            end;
        end;
        tag_x=num2str(x);
        jms=[tag_x '-iterace'];
        imwrite(uint8(BB_v_write),gray(256),[path 'zpracovano\' jms '.bmp']);
        disp(tag_x);
    end;
else
    for x=1:max_iter %pro prvni az maximalni iteraci
        C=fft2(B); %FFT matice B
        C=fftshift(C); % zamena stredu za periferie
        C_f=angle(C); %hodnoty faze
        C_A=abs(C);  % hodnoty amplitudy
        C_A_real=real(C_A);
        C_A_imag=imag(C_A);
        C_P=C_A_real.^2 + C_A_imag.^2;   %power spektrum~intenzita
        max_C_P=max(max(C_P));  %urceni maximalni amplitudy
        min_C_P=min(min(C_P));  %urceni minimalni amplitudy
```

```
roz= max_C_P-min_C_P;   %rozsah amplitud
max_C_f=max(max(C_f));  %urceni maximalni faze
min_C_f=min(min(C_f));  %urceni minimalni faze
if ZB_if==1
  for m=max(max(bo_horni-ZB,1)):min(min(bo_dolni+ZB,sz(1)));
      for n=max(max(bo_leva-ZB,1)):min(min(bo_prava+ZB, sz(2)));
         C_P_norm(m,n)=(255*(C_P(m,n)-min_C_P)/(max_C_P-min_C_P));
         V_norm(m,n)=(255*(V(m,n)-min_V)/(max_V-min_V));
         if ER_if==1
           V_norm(m,n)=(255*(V(m,n)-min_V)/(max_V-min_V));
         end;
         if AA_if==1
            V_norm(m,n)=alpha*V(m,n)+(1-alpha)*C_P_norm(m,n);
         end;
         CC(m,n)=sqrt(V_norm(m,n))*exp(1i*angle(C(m,n)));
         RMS(m,n)=((C_P_norm(m,n)-V_norm(m,n))^2);
      end;
  end;
  V_norm_sum=sum(sum(V_norm));
  RMS_disp=sum(sum(RMS));
  RMS_disp=RMS_disp/V_norm_sum;
else
   for m=1:sz(2);
      for n=1 : sz(1);
         C_P_norm(m,n)=(255*(C_P(m,n)-min_C_P)/(max_C_P-min_C_P));
         V_norm(m,n)=(255*(V(m,n)-min_V)/(max_V-min_V));
         if ER_if==1
           V_norm(m,n)=(255*(V(m,n)-min_V)/(max_V-min_V));
         end;
         if AA_if==1
            V_norm(m,n)=alpha*V(m,n)+(1-alpha)*C_P_norm(m,n);
         end;
         CC(m,n)=sqrt(V_norm(m,n))*exp(1i*angle(C(m,n)));
         RMS(m,n)=((C_P_norm(m,n)-V_norm(m,n))^2);
      end;
   end;
   V_norm_sum=sum(sum(V_norm));
   RMS_disp=sum(sum(RMS));
   RMS_disp=RMS_disp/V_norm_sum;
end;
disp (RMS_disp);
BB=ifft2(CC);
BB=fftshift(BB);
BB_fa=angle(BB);
max_BB_fa=max(max(BB_fa));  %urceni maximalni amplitudy
min_BB_fa=min(min(BB_fa));  %urceni minimalni amplitudy
BB_fa_abs=sqrt(BB_fa.^2);
for m= 1 : sz(2);
   for n=1 : sz(1);
      if BB_fa_abs(m,n)<=pi/2
```

```
            BB_v_write(m,n)=0;
            BB_v(m,n)=0;
          else
            BB_v_write(m,n)=255;
            BB_v(m,n)=pi;
          end;
        end;
      end;
      %figure
      %imshow (BB_v_write);
      B(:,:)=double(0);
      C(:,:)=double(0);
      B_fa(:,:)=double(0);
      for m= 1 : sz(2);
        for n=1 : sz(1);
          B(m,n)=255*exp(1i*BB_v(m,n));
        end;
      end;
      tag_x=num2str(x);
      jms=[tag_x '-iterace'];
      imwrite(uint8(BB_v_write),gray(256),[path 'zpracovano\' jms '.bmp']);
      disp(tag_x);
    end;
end;
```