

Video

- Samotný program je spuštěn v Matlab App designer, který splňuje lepší uživatelské prostředí
- Popis, jak program funguje je nalezen v příloze Bakalářské práce

Získání pozice

```
[file, path] = uigetfile('*.avi','*.mp4','Select a Text File');
if isequal(file, 0)
    disp('User selected cancel');
else
    disp(['User selected ', fullfile(path,file)]);
    input_video_file = fullfile(path,file);
    vid = VideoReader(input_video_file);
    app.vid=vid %Pro použití v aplikaci
end

kc=50000; %Chceme získat pouze jednu pozici, vysoký krok pro rychlejší vyhodnocení

hasFrame(vid)
    framec=readFrame(vid);
    numberOfFramesc = vid.NumFrames;
    framesToReadc = 1:kc:numberOfFramesc; %Bere každý 50000 snímek
    thisFramec = read(vid,framesToReadc(1));
    imtool(thisFramec) %Vyhodí jeden obrázek, kde dojde k vybrání polohy
```

Start

```
app.Lamp.Color='red' %Chceme detekci červené diody při pokusu (funkce App Designer)
drawnow()
vid=app.vid; %funkce app.vid obsahuje samotné video
k=app.kEditField.Value; %Vložení kroku "k"
u=[]; %Pro ukádání hodnot
while hasFrame(vid)
    frame=readFrame(vid); %Přečte frames a uloží je
    numberOfFrames = vid.NumFrames; %Počet
    framesToRead = 1:k:numberOfFrames; %Rozpočítá frames s určitým krokem k

    for h = 1:length(framesToRead) %h=1 --> první krok, platí od 1 do
                                    % počtu všech frames

        %pokud to platí, aplikuje na každý snímek danou segmentaci níže
        thisFrame = read(vid, framesToRead(h));
        pos1=app.PoziceCTRLVEditField.Value;
        pos=str2num(pos1);
```

```

cr=imcrop(thisFrame,pos); %Zde ořeznutí podle zadané pozice

Rpokus=im2gray(cr); %Převedení obrázku z RGB na obraz
                        % ve stupních šedi
nHood=true(3);
Rpokus=rangefilt(Rpokus,nHood); %Vykreslí obrys jak objektu
                                % tak tekutého robota

Rpokus=medfilt2(Rpokus,[7 7]);
Rpokus=imbinarize(Rpokus); %Prahování
Rpokus=imfill(Rpokus,"holes"); %Vyplní mezery
SE=strel("cube",14);
Rpokus=imclose(Rpokus,SE);
Rpokus=imclearborder(Rpokus); %Zbaví se nepotřebných
                                % pixelů u okrajů
Rpokus = bwareaopen(Rpokus, 300); %vše tvořený méně než
                                % "n" pixely se odstraní

imshow(Rpokus);

%
%určuji interval pravděpodobného průměru tekutého
%robota
    Rmin = 6; %20-40 normálně, pro malou 6-40
    Rmin=app.RminEditField.Value;
    Rmax = 60;
    %přes fci najdu střed a poloměr

[centersBright, radiiBright] = imfindcircles(Rpokus,[Rmin Rmax], 'Object'
                                                %nalezení tekutého robota
a=viscircles(centersBright, radiiBright,'Color','b');
rp=radiiBright;
Sp=centersBright;

%
    if isempty(rp) %určí prázdnou hodnotu a přeskočí ji
        continue
    end

r=rp(1)+2; %Podmínka pro vybrání jednoho kruhu
S=[centersBright(1,1),centersBright(1,2)]; %předešla jsem tomu,
    % aby to bralo dva kruhy, občas totiž viscircles
    % přečetl vlas jako kruh
roi = images.roi.Circle(gca, 'Center' ,S, 'Radius' ,r);
                                %označení kruhu a jeho vymazání - mask
mask=createMask(roi);
Rpokus(mask)=0;
imshow(Rpokus);
[x,y]=find(Rpokus);
xvlas=mean(x);
yvlas=mean(y);
u=[u;h];

```

```
% KAPKA I OBJEKT
```

```
tabulka = [xvlas,yvlas,S];
```

```
%JENOM KAPKA BEZ OBJEKTU
```

```
%                                tabulka=[Sp,Sp,Sp]

    if h == u(1) %vytvoří tabulku, podmínka zapsání první hodnoty
        htabulka = tabulka;
    else
        htabulka = [htabulka; tabulka];
    end

end

    if h==length(framesToRead)
        break
    end
end
htabulka=rmmissing(htabulka); %vymaže nežádoucí hodnoty

table = height(htabulka)-1;
%Výpočet rychlosti za čas
for e=1:table %protože h=velikosti tabulky table
    if e<table
        t=(e-1)*k/20;
        tv=k/20;
        n=e+1;

        dx=htabulka(n,3)-htabulka(e,3);
        dy=htabulka(n,4)-htabulka(e,4);

        r=app.PolomrEditField.Value;
        s=sqrt(dx^2+dy^2);
        v1=s/tv;
        v=v1*24/r/2; %Kalibrace

        tab = [t,v];
        if e == 1
            vtabulka = tab;
        else
            vtabulka = [vtabulka; tab];
        end

    if e==table-1
        break
    end
end
end
end
```

```

vtabulka=rmmissing(vtabulka);
vtabulka;

%Výpočet cosinu za čas
tab = height(htabulka)-1; %podmínka
for x = 1:table
    cas=x*k/20; %jen pro kontrolu
    if x<=tab
        y=x+1;
        Salfa=[htabulka(x,3),htabulka(x,4)]; %střed
        b1=[htabulka(x,1),htabulka(x,2)]; %bod 1
        b2=[htabulka(y,1),htabulka(y,2)]; %bod 2
        atan = atan2(abs(det([b2-Salfa;b1-Salfa])),dot(b2-Salfa,b1-Salfa)); %úhel
        ang= atan*180/pi; %přepočet na stupně
        ang= rmmissing(ang); % Zajišťuje odstranění Nan hodnoty
            % (počáteční položení objektu, tam není)

        cosf=cos(ang);

        tabl =[ang];
        if x == 1
            utabulka = tabl;
        else
            utabulka = [utabulka; tabl];
        end

    end
end
a=height(utabulka);
for tcas=1:a %aby tabulky měly stejný počet řádků jak s časem tak s úhlem
    cas=tcas*k/20; %výpočet času
    tabc =[cas]; %přidávám další tabulku, při první tabulce kvůli nan posunuté hodnoty
    if tcas == 1
        ttabulka = tabc;
    else
        ttabulka = [ttabulka; tabc];
    end
end

podm = height(utabulka);
for z=1:x
    if z<=podm
        w=cumsum(utabulka(:,1)); %sčítá hodnoty (posloupnost)
        cas=ttabulka(:,1); %tabulka s časem
        cosf=cosd(w); %tabulka s úhlem přepočtšným na cosinus

        tab2=([cas,cosf]);
        tab3=([cas,w]);
    end
end

```

```

    if z == 1
        ctabulka = tab2;
        c1tabulka=tab3;
    else
        ctabulka = [ctabulka; tab2];
        c1tabulka= [c1tabulka;tab3];
    end

end

end

end

%Výpočet frakvence za čas
c1tabulka;
ctabulka=rmmissing(ctabulka);
c1tabulka=rmmissing(c1tabulka);
nu=c1tabulka(end,end); %zjistím si poslední hodnotu tabulky
nu_pocet=nu/360 %ted' vím, kolikrát se to otočí
tabnas=[360, 720, 1080, 1440, 1800, 2160, 2520, 2880, 3240, 3600, 3960, 4320, 4680, 5040, 5400];
nuz=floor(nu_pocet); %zaokrouhluju na nejmenší celé číslo
beru=tabnas(1:nuz); %vybere jen počet potřebných násobků

% Najděte nejbližší hodnotu k dané hodnotě
rozdil = abs(c1tabulka(:,2) - беру);
[~, index] = min(rozdil);
nejblizsi_hodnota = c1tabulka(index,2);
ncas = c1tabulka(index,1);
ncas1=[0;ncas];
hodn=nejblizsi_hodnota/360;
for f=1:nuz
    casf=ncas1(f+1)-ncas1(f); %Potřebuju vědět čas otočení
    frek=1/casf;
    omega=frek*2*pi;
    tab4=[omega];
    if f == 1
        frtabulka=tab4;
    else
        frtabulka = [frtabulka; tab4];
    end

end

ftabulka=[ncas,frtabulka];

app.Lamp.Color = 'green';
drawnow()

%hodnoty pro použití aplikace

```

```

app.htabulka5=htabulka;
app.htabulka1=htabulka(:,3);
app.htabulka2=htabulka(:,4);
app.htabulka3=htabulka(:,1);
app.htabulka4=htabulka(:,2);
app.vtabulka1=vtabulka(:,1);
app.vtabulka2=vtabulka(:,2);
app.ctabulka1=ctabulka(:,1);
app.ctabulka2=ctabulka(:,2);
app.ftabulka1=ftabulka(:,1);
app.ftabulka2=ftabulka(:,2);

```

Sekvence

Získání pozice

```

file = uigetdir(); % otevření dialogového okna pro výběr složky
if file ~= 0 % pokud uživatel vybral složku
    app.FolderPathEditField.Value = file; % nastavení hodnoty v editovatelném poli
    b = dir(fullfile(file, '*.jpg')); % uložení seznamu souborů do vlastnosti "FileList"
end

filename=fullfile(file,b(1).name);
x=imread(filename);
imtool(x)

```

Start - Princip je stejný jako u videa, pouze zde chybí krok na rozdělení videa na jednotlivé snímky (frames), ty už samotná sekvence obsahuje

```

app.Lamp_2.Color='red';
drawnow()
file=app.FolderPathEditField.Value;
b = dir(fullfile(file, '*.jpg'));
n = numel(b);
k=app.kEditField.Value;
kr=k/20;
u=[];

for h = 1:kr:n %h=1 --> první krok, platí od 1 do počtu těch framů
    filename=fullfile(file,b(h).name);
    x1=imread(filename);

    pos2=app.PoziceCTRLVEditField.Value;
    pos1=str2num(pos2);
    cr=imcrop(x1,pos1);

    % %imshow(cr)
    RPokus1=im2gray(cr);
    nHood=true(3);

```

```

Rpokus1=rangefilt(RPokus1,nHood);
%       imshow(Rpokus1);
Rpokus1=medfilt2(Rpokus1,[7 7]);
Rpokus1=imbinarize(Rpokus1);
Rpokus1=imfill(Rpokus1,"holes");
SE=strel("cube",14);
Rpokus1=imclose(Rpokus1,SE);
Rpokus1=imclearborder(Rpokus1);
Rpokus1 = bwareaopen(Rpokus1, 300); %vše tvořený méně než "n"
                                     % pixely se odstraní

imshow(Rpokus1);

%
%určuji interval pravděpodobného průměru
Rmin=app.RminEditField.Value; %20-40 normálně, pro malou 6-40
Rmax = 70;
%přes fci najdu střed a poloměr

[centersBright, radiiBright] = imfindcircles(Rpokus1,[Rmin Rmax], 'Obje
a=viscircles(centersBright, radiiBright,'Color','b');
rp=radiiBright; %v pixelech
Sp=centersBright;

    if isempty(rp) %určí prázdnou hodnotu a přeskočí ji
        continue
    end
    r=rp(1)+2; %PODMÍNKA PRO VYBRÁNÍ JEDNOHO
    %KRUHU, BOHUŽEL NASTANE CHYBA POKUD NĚKTERÁ HODNOTA
    %r=rp;
    S1=[centersBright(1,1),centersBright(1,2)]; %předešla jsem tomu,
        % aby to bralo dva kruhy, občas totiž viscircles
        % přečetl vlas jako kruh
    roi = images.roi.Circle(gca, 'Center' ,S1, 'Radius' ,r);
        %označení kruhu a jeho vymazání - mask
    mask=createMask(roi);
    Rpokus1(mask)=0;
    imshow(Rpokus1);
    [x1,y]=find(Rpokus1);
    xvlas1=mean(x1);
    yvlas1=mean(y);
    u=[u;h];

%KAPKA I OBJEKT
    tabulka1=[xvlas1,yvlas1,S1];

%JENOM KAPKA BEZ OBJEKTU

%
    tabulka1=[Sp,Sp,Sp]

```

```

        if h==u(1)
            if h==1
                htabulka = tabulka1;

            else
                htabulka = [htabulka; tabulka1];
            end

            if h==n
                break
            end
        end
    end
    htabulka=rmmissing(htabulka);
    htabulka;
    table = height(htabulka)-1;
    for e=1:table %protože h=velikosti tabulky table
        if e<table
            t=(e-1)*k/20;
            tv=k/20; %nebo stačí napsat, že je vždy k/20
            n=e+1;

            dx=htabulka(n,3)-htabulka(e,3);
            dy=htabulka(n,4)-htabulka(e,4);

            r=app.PolomrEditField.Value;
            s=sqrt(dx^2+dy^2);
            v1=s/tv;
            v=v1*24/r/2;

            tab = [t,v];
            if e == 1
                vtabulka = tab;
            else
                vtabulka = [vtabulka; tab];
            end

            if e==table-1
                break
            end
        end
    end
    vtabulka=rmmissing(vtabulka);
    vtabulka;

    tab = height(htabulka)-1; %podmínka
    for x1 = 1:table
        cas=x1*k/20; %jen pro kontrolu
        if x1<=tab
            y=x1+1;

```



```

Salfa=[htabulka(x1,3),htabulka(x1,4)]; %střed
b1=[htabulka(x1,1),htabulka(x1,2)]; %bod 1
b2=[htabulka(y,1),htabulka(y,2)]; %bod 2
atan = atan2(abs(det([b2-Salfa;b1-Salfa])),dot(b2-Salfa,b1-Salfa)); %úhel
ang= atan*180/pi; %přepočet na stupně
ang= rmmissing(ang); % Zajišťuje odstranění Nan hodnoty
      % (počáteční položení objektu, tam není)

cosf=cos(ang);

      tabl =[ang];
      if x1 == 1
          utabulka = tabl;
      else
          utabulka = [utabulka; tabl];
      end

end

end
a=height(utabulka);
for tcas=1:a %aby tabulky měly stejný počet řádků jak s časem tak s úhlem
    cas=tcas*k/20; %výpočet času
    tabc =[cas]; %přidávám další tabulku, při první tabulce kvůli nan posunuté hodnoty
    if tcas == 1
        ttabulka = tabc;
    else
        ttabulka = [ttabulka; tabc];
    end
end
utabulka;
ttabulka;
% sum=sum(utabulka(:,2), 'omitnan' );
podm = height(utabulka);
for z=1:x1
    if z<=podm;
        w=cumsum(utabulka(:,1)); %sčítá hodnoty (posloupnost)
        cas=ttabulka(:,1); %tabulka s časem
        cosf=cosd(w); %tabulka s úhlem přepočtšným na cosinus

        tab2=([cas,cosf]);
        tab3=([cas,w]);
        if z == 1
            ctabulka = tab2;
            c1tabulka=tab3;
        else
            ctabulka = [ctabulka; tab2];
            c1tabulka= [c1tabulka;tab3];
        end
    end
end

```

```

end

end
c1tabulka
% ctabulka=rmmissing(ctabulka);
% c1tabulka=rmmissing(c1tabulka);
nu=c1tabulka(end,end); %zjistím si poslední hodnotu tabulky
nu_pocet=nu/360 %ted' vím, kolikrát se to otočí
tabnas=[360, 720, 1080, 1440, 1800, 2160, 2520, 2880, 3240, 3600, 3960, 4320, 4680, 5040, 5400];
nuz=floor(nu_pocet); %zaokrouhluju na nejmenší celé číslo
beru=tabnas(1:nuz); %vybere jen počet potřebných násobků

% Najděte nejbližší hodnotu k dané hodnotě
rozdil = abs(c1tabulka(:,2) - beru);
[~, index] = min(rozdil);
nejblizsi_hodnota = c1tabulka(index,2);
% ncas = c1tabulka(index,1);
[~, idx] = ismember(nejblizsi_hodnota, c1tabulka);
% získání odpovídajících časů z matice cas
myRows = c1tabulka(ismember(c1tabulka(:,2), nejblizsi_hodnota), :);
ncas2=myRows(:,1);
ncas = ncas2(1:nuz,:);
ncas1=[0;ncas];
hodn=nejblizsi_hodnota/360;
for f=1:nuz
    casf=ncas1(f+1)-ncas1(f); %Potřebuju vědět čas otočení
    frek=1/casf;
    omega=frek*2*pi;
    tab4=[omega];
    if f == 1
        frtabulka=tab4;
    else
        frtabulka = [frtabulka; tab4];
    end
end

end

ftabulka=[ncas,frtabulka];
ftabulka;
filename = 'pokus.xlsx';
writematrix(ftabulka,filename,'Sheet',1,'Range','G56')

app.Lamp.Color = 'green';
drawnow()

app.Lamp_2.Color='green';
drawnow()

app.htabulka5=htabulka;
app.htabulka1=htabulka(:,3);

```

```

app.htabulka2=htabulka(:,4);
app.htabulka3=htabulka(:,1);
app.htabulka4=htabulka(:,2);
app.vtabulka1=vtabulka(:,1);
app.vtabulka2=vtabulka(:,2);
app.ctabulka1=ctabulka(:,1);
app.ctabulka2=ctabulka(:,2);
app.ftabulka1=ftabulka(:,1);
app.ftabulka2=ftabulka(:,2);

```

Grafy

Trajektorie kapky v reálném měřítku 24x24

```

r=app.PolomrEditField.Value;
axes_x = app.htabulka1;
axes_y = app.htabulka2;

plot(app.UIAxes, axes_x, axes_y)
daspect([24 24 24])
p=2;
m1=mean(app.htabulka1)+p; %abych zaručila skutečně střed celkové rotace
m2=mean(app.htabulka2)+p;%Lze přičít hodnotu - navýšení pixelů středu, ne zápor

Dolx=m1-r; %91 je poloměr kapky --> při cropování stačí změřit
Dopx=m1+r;
Doly=m2-r;
Dopy=m2+r;
PrumD1=[Dolx,Doly];
PrumD2=[Dopx,Dopy];
D1=mean(PrumD1); %průměr hodnot, stejná délka os
D2=mean(PrumD2);
x1=[D1 D2];
y1=[D1 D2];
xlim(app.UIAxes,x1)
ylim(app.UIAxes,y1)

```

Trajektorie kapky zblízka

```

z=app.hodnotazEditField.Value;
axes_x = app.htabulka1;
axes_y = app.htabulka2;

plot(app.UIAxes,axes_x, axes_y)
p=2;

```

```

m3=mean(app.htabulka1)+p; %abych zaručila skutečně střed celkové rotace
m4=mean(app.htabulka2)+p; %Lze přičít hodnotu - navýšení pixelů středu, ne zápor
Dolx1=m3-z; %90 je poloměr kapky --> při cropování stačí změřit (imtool)
Dopx1=m3+z;
Doly1=m4-z;
Dopy1=m4+z;
PrumD3=[Dolx1,Doly1];
PrumD4=[Dopx1,Dopy1];
D3=mean(PrumD3); %průměr hodnot, stejná délka os
D4=mean(PrumD4);
x2=[D3 D4];
y2=[D3 D4];
xlim(app.UIAxes,x2)
ylim(app.UIAxes,y2)

```

Trajektorie objektu v reálném měřítku 24x24

```

r=app.PolomrEditField.Value;
axes_x1 = app.htabulka3;
axes_y1 = app.htabulka4;

plot(app.UIAxes,axes_x1, axes_y1)
p=2;
m5=mean(app.htabulka3)+p; %abych zaručila skutečně střed celkové rotace
m6=mean(app.htabulka4)+p; %Lze přičít hodnotu - navýšení pixelů středu, ne zápor
Dolx2=m5-r; %91 je poloměr kapky --> při cropování stačí změřit
Dop2=m5+r;
Doly2=m6-r;
Dopy2=m6+r;
PrumD5=[Dolx2,Doly2];
PrumD6=[Dop2,Dopy2];
D5=mean(PrumD5); %průměr hodnot, stejná délka os
D6=mean(PrumD6);
x3=[D5 D6];
y3=[D5 D6];
xlim(app.UIAxes,x3)
ylim(app.UIAxes,y3)

```

Trajektorie objektu zblízka

```

z=app.hodnotazEditField.Value;
axes_x1 = app.htabulka3;
axes_y1 = app.htabulka4;
plot(app.UIAxes,axes_x1, axes_y1)
p=2;
m7=mean(app.htabulka3)+p; %abych zaručila skutečně střed celkové rotace
m8=mean(app.htabulka4)+p; %Lze přičít hodnotu - navýšení pixelů středu, ne zápor
Dolx3=m7-z; %91 je poloměr kapky --> při cropování stačí změřit
Dopx3=m7+z;
Doly3=m8-z;

```

```

Dopy3=m8+z;
PrumD7=[Dolx3,Doly3];
PrumD8=[Dopx3,Dopy3];
D7=mean(PrumD7); %průměr hodnot, stejná délka os
D8=mean(PrumD8);
x4=[D7 D8];
y4=[D7 D8];
xlim(app.UIAxes,x4)
ylim(app.UIAxes,y4)

```

Rychlost kapky - graf

```

axes_x = app.vtabulka1;
axes_y = app.vtabulka2;
plot(app.UIAxes2,axes_x, axes_y)
% xlabel("t [s]");
% ylabel("v [mm/s]"); %Nastavuje se již v aplikaci, zde není třeba
xl=app.XlimEditField.Value;
xl=str2num(xl);
yl=app.YlimEditField.Value;
yl=str2num(yl);
xlim(app.UIAxes2,xl)
ylim(app.UIAxes2,yl)

```

Cosinus objektu - graf

```

axes_x = app.ctabulka1;
axes_y = app.ctabulka2;
[axes_x, dum] = sort(axes_x); %aby se nepropojovala první a poslední hodnota
axes_y = axes_y(dum);
plot(app.UIAxes3,axes_x, axes_y)
xlabel("t [s]")
ylabel("Cos [alfa]")
ylim(app.UIAxes3,[-1 1])
xlc=app.XlimEditField_2.Value;
xlc=str2num(xlc);
xlim(app.UIAxes3,xlc);

```

Frakvence objektu - graf

```

axes_x4 = app.ftabulka1;
    axes_y4 = app.ftabulka2;
    plot(app.UIAxes4,axes_x4, axes_y4)
    xl=app.XlimEditField_3.Value;
    xl=str2num(xl);
    yl=app.YlimEditField_2.Value;
    yl=str2num(yl);
    xlim(app.UIAxes4,xl)
    ylim(app.UIAxes4,yl)

```

