## Analyzer.m

classdef Analyzer < handle

%UNTITLED9 Summary of this class goes here

% Detailed explanation goes here

properties (Access = private)

ap Analyzer\_params

rp Record\_params

pp PIV\_params

pg PIV\_grid

pf PIV\_frames

fi\_processor\_\_data Array\_temp\_storage

end

properties (SetAccess = protected, GetAccess = public)

mean\_field\_xd

mean\_field\_yd

end

methods

function obj = Analyzer( ap, rp, pp, pf, pg, fi\_processor\_\_data)

obj.ap = ap;

obj.rp = rp;

obj.pp = pp;

obj.pf = pf;

obj.pg = pg;

obj.fi\_processor\_\_data = fi\_processor\_\_data;

end

function [ts, pos\_ij\_arr] = Get\_time\_series(obj, data, pos\_ij\_arr)

%METHOD1 Summary of this method goes here

% Detailed explanation goes here

% pos\_ij\_arr - индексы на сетке в 2 колонки, если пустое - то

% для всех точек

% [10 12

% 15 46

% 13 14]

% ts - 3д матрица, склеенные по третей размерности

% трехколоночные матрица (время, xd, yd)

% if pos\_i > size(pg.xMat,1) || pos\_j > size(pg.xMat,2), return; end

if isempty(pos\_ij\_arr) %выбираем все

[pos\_i, pos\_j] = meshgrid( 1:size(obj.pg.xMat,1), 1:size(obj.pg.xMat,2) );

pos\_ij\_arr = [ pos\_i(:), pos\_j(:) ];

end

ts = nan( obj.pf.frame\_count, 4, size(pos\_ij\_arr,1) );

for ti = 1:obj.pf.frame\_count

t = ( obj.pf.first\_frames(ti) - 1 ) / obj.rp.fps;

d = data.Get(ti);

if isempty(d), continue; end

for pos\_ij\_arr\_i = 1:size(pos\_ij\_arr,1)

pos\_i = pos\_ij\_arr(pos\_ij\_arr\_i,1);

pos\_j = pos\_ij\_arr(pos\_ij\_arr\_i,2);

xd = d.xdispl(pos\_i,pos\_j);

yd = d.ydispl(pos\_i,pos\_j);

ts(ti, 1, pos\_ij\_arr\_i) = ti;

ts(ti, 2, pos\_ij\_arr\_i) = t;

ts(ti, 3, pos\_ij\_arr\_i) = xd;

ts(ti, 4, pos\_ij\_arr\_i) = yd;

end

end

%сглаживание

if round(obj.ap.ts\_smoothing\_window\_size) > 1

for pos\_ij\_arr\_i = 1:size(pos\_ij\_arr,1)

ts(:,3,pos\_ij\_arr\_i) = nanfastsmooth( ts(:,3,pos\_ij\_arr\_i), round(obj.ap.ts\_smoothing\_window\_size) );

ts(:,4,pos\_ij\_arr\_i) = nanfastsmooth( ts(:,4,pos\_ij\_arr\_i), round(obj.ap.ts\_smoothing\_window\_size) );

end

end

end

function [ xd, yd ] = Calc\_mean\_field(obj)

[xd, yd] = Analyzer.Get\_mean\_field( obj.fi\_processor\_\_data );

obj.mean\_field\_xd = xd;

obj.mean\_field\_yd = yd;

end

end

methods(Static)

function [xd, yd] = Get\_mean\_field( data, ti\_arr, ti\_center\_window )

% ti\_arr - индексы по которым считать

% ti\_center\_window - центр и размер окна индексов

% если не указать, то считает по всем (TODO жрет много памяти)

% xd, yd - 2д матрицs, смещений

if exist('ti\_arr','var') && ~isempty(ti\_arr)

[fi\_output\_arr, data\_is\_empty\_arr] = data.Get\_as\_cell(ti\_arr);

elseif exist('ti\_center\_window','var') && ~isempty(ti\_center\_window)

ti\_center = ti\_center\_window(1);

ti\_window = ti\_center\_window(2);

ti\_arr\_s = ti\_center - ceil (ti\_window/2);

ti\_arr\_e = ti\_center + floor(ti\_window/2);

ti\_arr\_s = max(ti\_arr\_s,1);

ti\_arr\_e = min(ti\_arr\_e, data.data\_size);

ti\_arr = ti\_arr\_s : ti\_arr\_e;

[fi\_output\_arr, data\_is\_empty\_arr] = data.Get\_as\_cell(ti\_arr);

else

[fi\_output\_arr, data\_is\_empty\_arr] = data.Get\_all\_as\_cell();

end

xd\_mat3 = cellfun(@(x) x.xdispl, fi\_output\_arr(~data\_is\_empty\_arr),'uni',false);

yd\_mat3 = cellfun(@(x) x.ydispl, fi\_output\_arr(~data\_is\_empty\_arr),'uni',false);

xd\_mat3 = cat(3,xd\_mat3{:});

yd\_mat3 = cat(3,yd\_mat3{:});

xd = mean(xd\_mat3,3,'omitnan');

yd = mean(yd\_mat3,3,'omitnan');

end

function [xd, yd] = Get\_median\_field( data, ti\_arr, ti\_center\_window )

% ti\_arr - индексы по которым считать

% ti\_center\_window - центр и размер окна индексов

% если не указать, то считает по всем (TODO жрет много памяти)

% xd, yd - 2д матрицs, смещений

if exist('ti\_arr','var') && ~isempty(ti\_arr)

[fi\_output\_arr, data\_is\_empty\_arr] = data.Get\_as\_cell(ti\_arr);

elseif exist('ti\_center\_window','var') && ~isempty(ti\_center\_window)

ti\_center = ti\_center\_window(1);

ti\_window = ti\_center\_window(2);

ti\_arr\_s = ti\_center - ceil (ti\_window/2);

ti\_arr\_e = ti\_center + floor(ti\_window/2);

ti\_arr\_s = max(ti\_arr\_s,1);

ti\_arr\_e = min(ti\_arr\_e, data.data\_size);

ti\_arr = ti\_arr\_s : ti\_arr\_e;

[fi\_output\_arr, data\_is\_empty\_arr] = data.Get\_as\_cell(ti\_arr);

else

[fi\_output\_arr, data\_is\_empty\_arr] = data.Get\_all\_as\_cell();

end

xd\_mat3 = cellfun(@(x) x.xdispl, fi\_output\_arr(~data\_is\_empty\_arr),'uni',false);

yd\_mat3 = cellfun(@(x) x.ydispl, fi\_output\_arr(~data\_is\_empty\_arr),'uni',false);

xd\_mat3 = cat(3,xd\_mat3{:});

yd\_mat3 = cat(3,yd\_mat3{:});

xd = median(xd\_mat3,3,'omitnan');

yd = median(yd\_mat3,3,'omitnan');

end

end

end

## Analyzer\_params.m

classdef Analyzer\_params

%UNTITLED10 Summary of this class goes here

% Detailed explanation goes here

properties

ts\_smoothing\_window\_size = 1; %целое

end

end

## Array\_temp\_storage.m

classdef Array\_temp\_storage < handle

%Array\_temp\_storage Cell массив с автоматическим сохранением данных на диск

% A = Array\_temp\_storage(101, 10, "temp\_storage");

% В него можно сохранять и извлекать данные как в cell array.

% На ХДД данные сохраняются сегментами. Автоматически сегменты

% сохраняются на ХДД в момент добавления/изменения элементов когда

% они заполнены целиком но при этом не являются текущими. Если писать

% данные в случайном порядке, сегменты будут заполняться медленно и не

% будут сохраняться на ХДД. При чтении весь сегмент извлекается из

% памяти, неактивные полные сегменты при этом убираются

% data\_size - размер массива с данными

% seg\_size - разммер сегмента. Может быть не

% кратен размеру данных, может превышать размер данных или быть Inf

% (тогда будет один сегмент, который сохранится на ХДД только по

% требованию через Move\_all\_to\_hdd)

% storage\_file\_path\_base - основа пути к файлам с данными. она добивается " ats\_seg0005"

% rw\_options:

% 'replace' - если есть старые файлы, удалить их

% 'read' - прочитать старые файлы, если из не достаотчно, выдать

% предупреждение

% 'new' - если есть старые файлы выдать ошибку

%%

properties (SetAccess = protected, GetAccess = public)

seg\_in\_mem

seg\_count

seg\_i\_arr %для каждого элемента номер сегмента

data\_i\_in\_seg %для каждого сегмента номера элементов

data\_is\_empty\_arr

data\_size

seg\_size

storage\_file\_path\_base

storage\_file\_path\_arr

data

version = 1;

end

%%

methods

function obj = Array\_temp\_storage(data\_size, seg\_size, storage\_file\_path\_base)

obj.data\_size = data\_size;

obj.seg\_size = seg\_size;

obj.data = cell(obj.data\_size,1);

% распределяем элементы по сегментам

obj.seg\_i\_arr = zeros(obj.data\_size,1);

seg\_i = 1;

items\_in\_current\_seg = 0;

for i = 1:obj.data\_size

if items\_in\_current\_seg >= seg\_size

seg\_i = seg\_i + 1;

items\_in\_current\_seg = 0;

end

obj.seg\_i\_arr(i) = seg\_i;

items\_in\_current\_seg = items\_in\_current\_seg + 1;

end

obj.seg\_count = obj.seg\_i\_arr(end);

obj.data\_i\_in\_seg = cell(obj.seg\_count,1);

for i = 1:obj.seg\_count

obj.data\_i\_in\_seg{i} = find(obj.seg\_i\_arr == i);

end

obj.seg\_in\_mem = true(obj.seg\_count,1);

% файлы для хранения. Если существуют, то удалить все файлы с

% такой базой

obj.storage\_file\_path\_base = storage\_file\_path\_base;

delete( [ obj.storage\_file\_path\_base '\*' ] );

obj.storage\_file\_path\_arr = cell(obj.seg\_count,1);

for seg\_i = 1:obj.seg\_count

obj.storage\_file\_path\_arr{seg\_i} = sprintf( '%s ats\_seg%04d.mat', obj.storage\_file\_path\_base, seg\_i );

end

obj.data\_is\_empty\_arr = true(obj.data\_size,1);

end

function Set(obj, i, item)

if i < 1 || i > obj.data\_size, return; end

% если сегмент на хдд, то возвращаем его в память

current\_segment = obj.seg\_i\_arr(i);

if ~obj.seg\_in\_mem( current\_segment )

obj.Seg\_move\_from\_hdd( current\_segment );

end

obj.data{i} = item;

obj.data\_is\_empty\_arr(i) = false;

% проверим есть ли полные сегменты кроме текущего и их

% переместим на хдд

seg\_is\_full\_arr = obj.Get\_seg\_is\_full();

seg\_is\_full\_arr(current\_segment) = false;

seg\_to\_move = find( seg\_is\_full\_arr & obj.seg\_in\_mem );

for seg\_to\_move\_i = 1:numel(seg\_to\_move)

obj.Seg\_move\_to\_hdd( seg\_to\_move(seg\_to\_move\_i) );

end

end

function item = Get(obj, i)

if i < 1 || i > obj.data\_size

item = [];

return;

end

% если сегмент на хдд, то возвращаем его в память

current\_segment = obj.seg\_i\_arr(i);

if ~obj.seg\_in\_mem(current\_segment)

obj.Seg\_move\_from\_hdd( current\_segment );

end

item = obj.data{i};

% проверим есть ли полные сегменты кроме текущего и их

% переместим на хдд

seg\_is\_full\_arr = obj.Get\_seg\_is\_full();

seg\_is\_full\_arr(current\_segment) = false;

seg\_to\_move = find( seg\_is\_full\_arr & obj.seg\_in\_mem );

for seg\_to\_move\_i = 1:numel(seg\_to\_move)

obj.Seg\_move\_to\_hdd( seg\_to\_move(seg\_to\_move\_i) );

end

end

function [output, data\_is\_empty\_arr] = Get\_all\_as\_cell(obj)

output = cell(obj.data\_size,1);

for i = 1:obj.data\_size

output{i} = obj.Get(i);

end

data\_is\_empty\_arr = obj.data\_is\_empty\_arr;

end

function [output, data\_is\_empty\_arr] = Get\_as\_cell(obj, ti\_arr)

output = cell(numel(ti\_arr),1);

for i = 1:numel(ti\_arr)

output{i} = obj.Get(ti\_arr(i));

end

data\_is\_empty\_arr = obj.data\_is\_empty\_arr(ti\_arr(i));

end

function Remove\_all\_files(obj)

for i = 1:obj.seg\_count

if exist(obj.storage\_file\_path\_arr{i},'file')

delete(obj.storage\_file\_path\_arr{i});

end

end

end

function Move\_all\_to\_hdd(obj)

seg\_in\_mem\_arr = find(obj.seg\_in\_mem);

for i = 1:numel(seg\_in\_mem\_arr)

obj.Seg\_move\_to\_hdd(seg\_in\_mem\_arr(i));

end

end

function Clear(obj)

obj.Remove\_all\_files();

obj.data = cell(obj.data\_size,1);

obj.data\_is\_empty\_arr = true(obj.data\_size,1);

obj.seg\_in\_mem = true(obj.seg\_count,1);

end

function Export\_to\_file(obj, file\_path\_base) % сохраняет все данные в зип файл и файл с инфой

% файл с информацией:

info\_file\_path = sprintf( '%s info.mat', file\_path\_base );

data\_size = obj.data\_size;

seg\_size = obj.seg\_size;

version = obj.version;

data\_is\_empty\_arr = obj.data\_is\_empty\_arr;

if ~exist(fileparts(info\_file\_path),'dir'), mkdir(fileparts(info\_file\_path)); end

save(info\_file\_path, 'version', 'data\_size', 'seg\_size', 'data\_is\_empty\_arr');

% файлы с данными

obj.Move\_all\_to\_hdd();

data\_file\_path = sprintf( '%s data.zip', file\_path\_base );

zip( data\_file\_path, obj.storage\_file\_path\_arr );

end

function Import\_from\_file(obj, file\_path\_base)

info\_file\_path = sprintf( '%s info.mat', file\_path\_base ); % файл с информацией

data\_file\_path = sprintf( '%s data.zip', file\_path\_base ); % файлы с данными

if ~exist(info\_file\_path,'file')

warning('Ошибка загрузки: нет файла с данными %s', file\_path\_base );

return;

end

if ~exist(data\_file\_path,'file')

warning('Ошибка загрузки: нет файла с данными %s', data\_file\_path );

return;

end

% очистим временные файлы

obj.Clear();

% файл с информацией

f\_info = load(info\_file\_path, 'version', 'data\_size', 'seg\_size', 'data\_is\_empty\_arr');

if obj.version ~= f\_info.version

error('Ошибка загрузки: не совпадающая версия');

end

if obj.data\_size ~= f\_info.data\_size

error('Ошибка загрузки: неправильный размер: %s', info\_file\_path);

end

if obj.seg\_size ~= f\_info.seg\_size

obj = Array\_temp\_storage(obj.data\_size, f\_info.seg\_size, obj.storage\_file\_path\_base);

warning('Данные с другим размером сегмента, размер сегмента изменен на %d: %s', obj.seg\_size, info\_file\_path);

end

obj.data\_is\_empty\_arr = f\_info.data\_is\_empty\_arr;

% файлы с данными распакуем во временную директорию

filenames = unzip( data\_file\_path, obj.storage\_file\_path\_base );

%проверяем файлы

if numel(filenames) ~= numel(obj.storage\_file\_path\_arr)

warning('Файл с данными не может быть загружен, неправильное количество файлов: %s', data\_file\_path );

delete(filenames{:});

rmdir(temp\_folder);

return

end

obj.seg\_in\_mem = false(obj.seg\_count,1); %все данные только на винчестере

end

end

%%

methods (Access = private)

function Seg\_move\_to\_hdd(obj, seg\_i)

% disp(['Seg\_move\_to\_hdd ' num2str(seg\_i)])

if ~obj.seg\_in\_mem(seg\_i)

warinig('Seg\_move\_to\_hdd: segment is not in memory');

return;

end

data\_in\_seg\_i = obj.data\_i\_in\_seg{seg\_i};

data\_in\_seg = obj.data(data\_in\_seg\_i);

file\_path = obj.storage\_file\_path\_arr{seg\_i};

if ~exist(fileparts(file\_path),'dir')

mkdir(fileparts(file\_path));

end

save(file\_path,'data\_in\_seg');

for i =1:numel(data\_in\_seg\_i)

obj.data{data\_in\_seg\_i(i)} = [];

end

obj.seg\_in\_mem(seg\_i) = false;

end

function Seg\_move\_from\_hdd(obj, seg\_i)

% disp(['Seg\_move\_from\_hdd ' num2str(seg\_i)])

data\_in\_seg\_i = obj.data\_i\_in\_seg{seg\_i};

file\_path = obj.storage\_file\_path\_arr{seg\_i};

f = load(file\_path,'data\_in\_seg');

data\_in\_seg = f.data\_in\_seg;

for i =1:numel(data\_in\_seg\_i)

obj.data{data\_in\_seg\_i(i)} = data\_in\_seg{i};

end

obj.seg\_in\_mem(seg\_i) = true;

end

function seg\_is\_full\_arr = Get\_seg\_is\_full(obj)

seg\_is\_full\_arr = false(obj.seg\_count,1);

for seg\_i = 1:obj.seg\_count

is\_full = ~any( obj.data\_is\_empty\_arr( obj.data\_i\_in\_seg{seg\_i} ) );

seg\_is\_full\_arr(seg\_i) = is\_full;

end

end

end

end

## Array\_temp\_storage\_test.m

obj = Array\_temp\_storage(11, 3, 'temp\_storage');

%%

for i = 1:101

obj.Set(i,-i);

end

for i = 1:101

obj.Set(i,i);

end

%%

% obj.Get(11)

%%

obj.Get\_all\_as\_cell()

%%

obj.Export\_to\_file('test ats\1');

%%

clear all

obj = Array\_temp\_storage(11, 5, 'temp\_storage');

obj.Import\_from\_file('test ats\1');

obj.Get\_all\_as\_cell()

## CalcBgMedian\_v2019\_10.m

function [ bg ] = CalcBgMedian\_v2019\_10( imageLoader, ti\_bg, percentage, color\_channel )

% считает медианное значение каждого пикселя из набора номеров кадров

% imageLoader - imLoader\_v2019\_1

% ti\_bg - номера кадров, могуть выходить за границы, могут быть не целыми

% percentage - от 0 до 100 перцентиль

if isempty(ti\_bg)

ti\_bg = 1:imageLoader.imNum;

end

if numel(ti\_bg) == 1

ti\_bg = round( linspace( 1, imageLoader.imNum, ti\_bg) );

end

ti\_bg(ti\_bg < 1 | ti\_bg > imageLoader.imNum) = [];

if isempty(ti\_bg)

warning('no frames');

bg = zeros(imageLoader.imSize(2),imageLoader.imSize(1));

else

bgall = zeros(imageLoader.imSize(2),imageLoader.imSize(1),numel(ti\_bg));

for i = 1:numel(ti\_bg)

im = imageLoader.getImage(ti\_bg(i));

im = im(:,:,color\_channel);

bgall(:,:,i) = im;

end

bgr = reshape(permute(bgall,[3 2 1]),[numel(ti\_bg), size(bgall,1) \* size(bgall,2)]);

p = prctile(bgr,percentage,1);

% p = median(bgr,1);

bg = reshape(p,[size(bgall,2), size(bgall,1)])';

end

end

## Core.m

classdef Core < handle

%UNTITLED6 Summary of this class goes here

% Detailed explanation goes here

%

properties

il Image\_loader

ipp Image\_preprocessor

piv\_processor PIV\_processor

fi\_processor Filter\_and\_interpolation\_processor

transform\_processor Transform\_processor

exporter Exporter

analyzer Analyzer

end

%

methods

function obj = Core(il, ipp, piv\_processor, fi\_processor, transform\_processor, exporter, analyzer )

obj.il = il;

obj.ipp = ipp;

obj.piv\_processor = piv\_processor;

obj.fi\_processor = fi\_processor;

obj.transform\_processor = transform\_processor;

obj.exporter = exporter;

obj.analyzer = analyzer;

end

end

end

## DoCC10.m

function [ PIV\_output ] = DoCC10( ...

im1,im2,...

pp, pg)

iteration\_tic = tic;

%DoCC6 Вычисление PIV над двумя изображениями

% pp PIV\_params

% pg PIV\_grid

%% eval params

s2e = pp;

clear fieldnames fieldnames\_i;

fieldnames\_my = fieldnames(s2e);

for fieldnames\_i = 1:numel(fieldnames\_my)

eval([fieldnames\_my{fieldnames\_i} '=s2e.' fieldnames\_my{fieldnames\_i} ';']);

end

clear fieldnames fieldnames\_i;

%% grid

xcArr = pg.xMat;

ycArr = pg.yMat;

doCCArr = pg.do\_PIV\_mat;

%%

if numel(xIDArr) == 1 && (isnan(xIDArr) || xIDArr == 0)

xIDArr = zeros(size(xcArr));

end

if numel(yIDArr) == 1 && (isnan(yIDArr) || yIDArr == 0)

yIDArr = zeros(size(xcArr));

end

xIDArr = round(xIDArr);

yIDArr = round(yIDArr);

%%

if any(size(xcArr) ~= size(ycArr) | size(xcArr) ~= size(doCCArr))

error 'DoCC: Разные размеры матриц координат';

end

if any(size(im1) ~= size(im2))

error 'DoCC: Разные размеры изображений';

end

xFieldCount = size(xcArr,1);

yFieldCount = size(xcArr,2);

imSize = fliplr(size(im1));

%%

status = nan\*zeros(size(xcArr));

xdispl = nan\*zeros(size(xcArr));

ydispl = nan\*zeros(size(xcArr));

CC\_maxValue = nan\*zeros(size(xcArr));

CC\_maxRaitio = nan\*zeros(size(xcArr));

mean\_im1 = nan\*zeros(size(xcArr));

% mean\_im2 = nan\*zeros(size(xcArr));

clear CC\_infoArr = cell(size(xcArr));

if (saveCCandIMC)

CC\_infoArr = cell(size(xcArr));

else

CC\_infoArr = [];

end

% if useBiasCorrection || saveCCandIMC

% [r,s] = meshgrid( -wSize(1)/2:wSize(1)/2 - 1, -wSize(2)/2:wSize(2)/2 - 1 );

% biasCorrMatrix = ( 1-abs(r) / wSize(1) ).\*( 1-abs(s) / wSize(2) );

% end

%% основной цикл

for xi = 1:xFieldCount

for yi = 1:yFieldCount

xc = xcArr(xi,yi);

yc = ycArr(xi,yi);

xID = xIDArr(xi,yi);

yID = yIDArr(xi,yi);

doCC = doCCArr(xi,yi);

status(xi,yi) = 0;

if ~doCC

status(xi,yi) = 85;

continue;

end

if isnan(xc) || isnan(yc) || isnan(xID) || isnan(yID)

status(xi,yi) = 84;

continue;

end

xp\_sp = NaN; yp\_sp = NaN;

%%

c = [xc, yc]; %центр

dc0 = [xID, yID]; % предсмещение

%% первый проход, если удастся, то он подменит xID, yID на уточненные

if doFirstPass

imArea1 = RectArea(c - dc0/2,wSize1);

imArea2 = RectArea(c + dc0/2,wSize1);

if ~(RectFitsin(imArea1,imSize) && RectFitsin(imArea2,imSize))

status(xi,yi) = 101;

continue;

end

[xS1, xE1, yS1, yE1] = BoardersOfArea(imArea1);

[xS2, xE2, yS2, yE2] = BoardersOfArea(imArea2);

im1c=im1(yS1:yE1, xS1:xE1);

im2c=im2(yS2:yE2, xS2:xE2);

result\_conv =fftshift(real(ifft2(conj(fft2(im1c)).\*fft2(im2c))));

mean\_im1(xi,yi) = mean(im1c(:));

% mean\_im2(xi,yi) = mean(im2c(:));

% %bias correction:

% if useBiasCorrection

% result\_conv = result\_conv./biasCorrMatrix;

% end

% максимумы

if pp.doMultiMax

minMatrix = imregionalmax(result\_conv);

maximums\_i = find(minMatrix); %индексы

maximums\_v = result\_conv(maximums\_i); %значения

if numel(maximums\_v > 1)

[maximums\_v1, maximums\_i1] = max(maximums\_v); %индекс и значение в списке максимумов

maximums\_v(maximums\_i1) = 0;

[maximums\_v2, maximums\_i2] = max(maximums\_v);

maximums\_v(maximums\_i1) = maximums\_v1;

CC\_maxRaitio(xi,yi) = maximums\_v1/maximums\_v2;

CC\_maxValue(xi,yi) = maximums\_v1;

else

CC\_maxRaitio(xi,yi) = Inf;

CC\_maxValue(xi,yi) = maximums\_v;

end

location = maximums\_i(maximums\_i1);

else

[~, location] = max(result\_conv(:));

maximums\_v1 = nan; maximums\_i1 = nan; maximums\_v2 = nan; maximums\_i2 = nan;

CC\_maxRaitio(xi,yi) = Inf;

CC\_maxValue(xi,yi) = maximums\_v1;

end

[yp,xp] = ind2sub(size(result\_conv),location);

CC\_xdispl = xp - wSize1(1)/2 - 1;

CC\_ydispl = yp - wSize1(2)/2 - 1;

if CC\_maxRaitio(xi,yi) < minS2NRatio

status(xi,yi) = 177;

elseif ~(abs(CC\_xdispl) < maxDispl(1) && abs(CC\_ydispl) < maxDispl(2))

status(xi,yi) = 178;

else

xID = xID + CC\_xdispl;

yID = yID + CC\_ydispl;

end

if (saveCCandIMC)

xd\_FirstPass = CC\_xdispl;

yd\_FirstPass = CC\_ydispl;

result\_conv\_firstPass = result\_conv;

end

else

if (saveCCandIMC)

xd\_FirstPass = nan;

yd\_FirstPass = nan;

result\_conv\_firstPass = nan;

end

end

%% основной проход

if status(xi,yi) == 0 % на случай что на первом этапе вылетели с ошибкой

dc0 = [xID, yID]; % предсмещение

imArea1 = RectArea(c - dc0/2,wSize);

imArea2 = RectArea(c + dc0/2,wSize);

if ~(RectFitsin(imArea1,imSize) && RectFitsin(imArea2,imSize))

status(xi,yi) = 1;

continue;

end

[xS1, xE1, yS1, yE1] = BoardersOfArea(imArea1);

[xS2, xE2, yS2, yE2] = BoardersOfArea(imArea2);

im1c=im1(yS1:yE1, xS1:xE1);

im2c=im2(yS2:yE2, xS2:xE2);

result\_conv =fftshift(real(ifft2(conj(fft2(im1c)).\*fft2(im2c))));

% назуляем ККФ за пределами максимального смещения (2019 11), плюс по одному пикселю на подпиксельный поиск

if ~isinf(maxDispl(2))

result\_conv( 1 : -maxDispl(2) -1 + wSize(2)/2 + 1 , : ) = 0;

result\_conv( maxDispl(2) + wSize(2)/2 + 1 + 1: end, : ) = 0;

end

if ~isinf(maxDispl(1))

result\_conv( :, 1 : -maxDispl(1) -1 + wSize(1)/2 + 1 ) = 0;

result\_conv( :, maxDispl(1) + wSize(1)/2 + 1 +1 : end ) = 0;

end

mean\_im1(xi,yi) = mean(im1c(:));

% mean\_im2(xi,yi) = mean(im2c(:));

% %bias correction:

% if useBiasCorrection

% result\_conv = result\_conv./biasCorrMatrix;

% end

% максимумы

if pp.doMultiMax

minMatrix = imregionalmax(result\_conv);

maximums\_i = find(minMatrix); %индексы

maximums\_v = result\_conv(maximums\_i); %значения

if numel(maximums\_v > 1)

[maximums\_v1, maximums\_i1] = max(maximums\_v); %индекс и значение в списке максимумов

maximums\_v(maximums\_i1) = 0;

[maximums\_v2, maximums\_i2] = max(maximums\_v);

maximums\_v(maximums\_i1) = maximums\_v1;

CC\_maxRaitio(xi,yi) = maximums\_v1/maximums\_v2;

CC\_maxValue(xi,yi) = maximums\_v1;

else

CC\_maxRaitio(xi,yi) = Inf;

CC\_maxValue(xi,yi) = maximums\_v;

end

location = maximums\_i(maximums\_i1);

else

[maximums\_v, location] = max(result\_conv(:));

maximums\_v1 = nan; maximums\_i1 = nan; maximums\_v2 = nan; maximums\_i2 = nan;

CC\_maxRaitio(xi,yi) = Inf;

CC\_maxValue(xi,yi) = maximums\_v1;

end

[yp,xp] = ind2sub(size(result\_conv),location);

CC\_xdispl = xp - wSize(1)/2 - 1;

CC\_ydispl = yp - wSize(2)/2 - 1;

if CC\_maxRaitio(xi,yi) < minS2NRatio

status(xi,yi) = 77;

elseif ~(abs(CC\_xdispl) < maxDispl(1) && abs(CC\_ydispl) < maxDispl(2))

status(xi,yi) = 78;

else

xdispl(xi,yi) = xID + CC\_xdispl;

ydispl(xi,yi) = yID + CC\_ydispl;

%добавка от подпиксельного поиска

if ~useSubpixel

status(xi,yi) = 0;

elseif ~(xp > 1 && xp < wSize(1) && yp > 1 && yp < wSize(2))%проверка на правильное положение максимума

status(xi,yi) = 4;

else

Cx = result\_conv(yp,xp-1:xp+1);

Cy = result\_conv(yp-1:yp+1,xp)';

Cx23 = Cx(2)/Cx(3);Cx12 = Cx(1)/Cx(2);Cy23 = Cy(2)/Cy(3);Cy12 = Cy(1)/Cy(2);

minC = 0;

maxC = 1e10;

if (Cx12 > maxC || Cx23 > maxC || Cy12 > maxC || Cy23 > maxC) ||...

(Cx12 <= minC || Cx23 <= minC || Cy12 <= minC || Cy23 <= minC)

status(xi,yi) = 41;

else

xp\_sp = -0.5\*(log( Cx23 )+log( Cx12 )) / (log( Cx23 )-log( Cx12 ));

yp\_sp = -0.5\*(log( Cy23 )+log( Cy12 ))/ (log( Cy23 )-log( Cy12 ));

if ~(abs(xp\_sp)<1 && abs(yp\_sp)<1)

error ('DoCC: subpixel: Ошибка положения максимума');

end

status(xi,yi) = 0;

xdispl(xi,yi) = xID + CC\_xdispl + xp\_sp;

ydispl(xi,yi) = yID + CC\_ydispl + yp\_sp;

end

end

end

end

%%

if (saveCCandIMC)

clear CC\_info;

CC\_info.xi = xi;

CC\_info.yi = yi;

CC\_info.xc = xc;

CC\_info.yc = yc;

CC\_info.status = status(xi,yi);

CC\_info.imc1 = im1c;

CC\_info.imc2 = im2c;

CC\_info.result\_conv = result\_conv;

CC\_info.result\_conv\_firstPass = result\_conv\_firstPass;

% CC\_info.r = r;

% CC\_info.s = s;

CC\_info.CC\_xdispl = CC\_xdispl;

CC\_info.CC\_ydispl = CC\_ydispl;

CC\_info.xd\_FirstPass = xd\_FirstPass;

CC\_info.yd\_FirstPass = yd\_FirstPass;

CC\_info.xp\_sp = xp\_sp;

CC\_info.yp\_sp = yp\_sp;

CC\_info.xdispl = xdispl(xi,yi);

CC\_info.ydispl = ydispl(xi,yi);

CC\_info.maximums\_v = maximums\_v;

CC\_info.maximums\_v1 = maximums\_v1;

CC\_info.maximums\_i1 = maximums\_i1;

CC\_info.maximums\_v2 = maximums\_v2;

CC\_info.maximums\_i2 = maximums\_i2;

CC\_info.CC\_maxRaitio = CC\_maxRaitio(xi,yi);

CC\_infoArr{xi,yi} = CC\_info;

end

end

end

PIV\_output.status = status;

PIV\_output.xdispl = xdispl;

PIV\_output.ydispl = ydispl;

PIV\_output.CC\_maxValue = CC\_maxValue;

PIV\_output.CC\_maxRaitio = CC\_maxRaitio;

PIV\_output.CC\_infoArr = CC\_infoArr;

PIV\_output.pp = pp;

PIV\_output.pg = pg;

PIV\_output.imSize = imSize;

PIV\_output.mean\_im1 = mean\_im1;

% PIV\_output.mean\_im2 = mean\_im2;

PIV\_output.time\_of\_iteration = toc(iteration\_tic);

end

function result = RectFitsin(imArea,imSize)

[xS, xE, yS, yE] = BoardersOfArea(imArea);

imSizeX = imSize(1);

imSizeY = imSize(2);

result = (xS >= 1 && xE <= imSizeX && yS >= 1 && yE <= imSizeY);

end

function imArea = RectArea(c,wSize)

xc = c(1);

yc = c(2);

xS = xc - round(wSize(1)/2);

xE = xc + round(wSize(1)/2)-1;

yS = yc - round(wSize(2)/2);

yE = yc + round(wSize(2)/2)-1;

imArea = round([xS xE yS yE]);

end

function [xS, xE, yS, yE] = BoardersOfArea(imArea)

xS = imArea(1);

xE = imArea(2);

yS = imArea(3);

yE = imArea(4);

end

## Exporter.m

classdef Exporter < handle

%UNTITLED8 Summary of this class goes here

% Detailed explanation goes here

properties

decimal\_delimeter = '.';

column\_delimeter = '\t';

% save\_header = true; TODO

ommit\_nans = false;

end

methods

function Export\_all\_fields\_to\_one\_file( obj, file\_path, data, transform\_processor )

% data - что-то типа Array\_temp\_storage, имеет метод Get(ti) и размером с pf

% Get возвращает какую-то структуру, содержащую матрицы xdispl, ydispl и status (все числовые), размером с pg

if exist(file\_path,'file')

delete(file\_path);

end

if ~exist(fileparts(file\_path),'dir'), mkdir(fileparts(file\_path)); end

fileID = fopen(file\_path,'w');

for ti = 1:numel(transform\_processor.time)

t = transform\_processor.time(ti);

x = transform\_processor.xMat;

y = transform\_processor.yMat;

d = data.Get(ti);

if ~isempty(d)

xd = transform\_processor.Convert\_xdispl( d.xdispl );

yd = transform\_processor.Convert\_ydispl( d.ydispl );

else

xd = nan;

yd = nan;

end

% status = d.status;

for i = 1:numel(x)

if obj.ommit\_nans && (isnan(xd(i)) || isnan(yd(i)))

continue;

end

output\_string = sprintf('%f%s%f%s%f%s%f%s%f\r\n',...

x (i), obj.column\_delimeter,...

y (i), obj.column\_delimeter,...

t, obj.column\_delimeter,...

xd(i), obj.column\_delimeter,...

yd(i));

if obj.decimal\_delimeter ~= '.'

output\_string(output\_string == '.') = obj.decimal\_delimeter;

end

fprintf(fileID,output\_string);

end

end

fclose(fileID);

end

%%

function Export\_all\_fields\_to\_separate\_files( obj, file\_path\_base, data, transform\_processor )

% data - что-то типа Array\_temp\_storage, имеет метод Get(ti) и размером с pf

% Get возвращает какую-то структуру, содержащую матрицы xdispl, ydispl и status (все числовые), размером с pg

if ~exist(fileparts(file\_path\_base),'dir'), mkdir(fileparts(file\_path\_base)); end

for ti = 1:numel(transform\_processor.time)

t = transform\_processor.time(ti);

if ~contains( file\_path\_base, '~info~')

error('Путь должен содержать подстроку ~info~');

end

file\_path = strrep( file\_path\_base, '~info~', ...

sprintf(', time %08.5f', t) );

if exist(file\_path,'file')

delete(file\_path);

end

d = data.Get(ti);

x = transform\_processor.xMat;

y = transform\_processor.yMat;

xd = transform\_processor.Convert\_xdispl( d.xdispl );

yd = transform\_processor.Convert\_ydispl( d.ydispl );

obj.Export\_one\_field\_to\_file( file\_path, x, y, xd, yd );

end

end

%%

function Export\_time\_series\_to\_separate\_files( obj, file\_path\_base, ts, transform\_processor, pos\_ij\_arr )

% file\_path\_base - путь к файлам, должен содержать подстроку '~info~', которая заменится на инфо текущего кадра

% ts - в пикселях на кадр

if ~exist(fileparts(file\_path\_base),'dir'), mkdir(fileparts(file\_path\_base)); end

for pos\_ij\_arr\_i = 1:size(ts,3)

pos\_i = pos\_ij\_arr(pos\_ij\_arr\_i,1);

pos\_j = pos\_ij\_arr(pos\_ij\_arr\_i,2);

x = transform\_processor.xMat(pos\_i,pos\_j);

y = transform\_processor.yMat(pos\_i,pos\_j);

if ~contains( file\_path\_base, '~info~')

error('Путь должен содержать подстроку ~info~');

end

file\_path = strrep( file\_path\_base, '~info~', ...

sprintf(', pos\_i %d, pos\_j %d, x %.5f, y %.5f', pos\_i, pos\_j, x, y) );

if exist(file\_path,'file')

delete(file\_path);

end

fileID = fopen(file\_path,'w');

for i = 1:size(ts,1)

t = ts(i, 2, pos\_ij\_arr\_i);

xd = ts(i, 3, pos\_ij\_arr\_i);

yd = ts(i, 4, pos\_ij\_arr\_i);

xd\_mps = transform\_processor.Convert\_xdispl( xd );

yd\_mps = transform\_processor.Convert\_ydispl( yd );

if obj.ommit\_nans && (isnan(xd) || isnan(yd))

continue;

end

output\_string = sprintf('%f%s%f%s%f\r\n',...

t , obj.column\_delimeter,...

xd\_mps, obj.column\_delimeter,...

yd\_mps);

if obj.decimal\_delimeter ~= '.'

output\_string(output\_string == '.') = obj.decimal\_delimeter;

end

fprintf(fileID,output\_string);

end

fclose(fileID);

end

end

%%

function Export\_one\_field\_to\_file( obj, file\_path, x\_m, y\_m, xd\_mps, yd\_mps )

if exist(file\_path,'file')

delete(file\_path);

end

fileID = fopen(file\_path,'w');

for i = 1:numel(x\_m)

if obj.ommit\_nans && (isnan(xd\_mps(i)) || isnan(yd\_mps(i)))

continue;

end

output\_string = sprintf('%f%s%f%s%f%s%f\r\n',...

x\_m (i), obj.column\_delimeter,...

y\_m (i), obj.column\_delimeter,...

xd\_mps(i), obj.column\_delimeter,...

yd\_mps(i));

if obj.decimal\_delimeter ~= '.'

output\_string(output\_string == '.') = obj.decimal\_delimeter;

end

fprintf(fileID,output\_string);

end

fclose(fileID);

end

%%

function Export\_profile\_to\_file( obj, file\_path, pos\_m, xd\_mps, yd\_mps)

pos\_m = pos\_m(:);

xd\_mps = xd\_mps(:);

yd\_mps = yd\_mps(:);

if exist(file\_path,'file')

delete(file\_path);

end

fileID = fopen(file\_path,'w');

for i = 1:numel(pos\_m)

if obj.ommit\_nans && (isnan(xd\_mps(i)) || isnan(yd\_mps(i)))

continue;

end

output\_string = sprintf('%f%s%f%s%f\r\n',...

pos\_m(i), obj.column\_delimeter,...

xd\_mps (i), obj.column\_delimeter,...

yd\_mps (i));

if obj.decimal\_delimeter ~= '.'

output\_string(output\_string == '.') = obj.decimal\_delimeter;

end

fprintf(fileID,output\_string);

end

fclose(fileID);

end

%%

end

end

## ExtractNumber.m

function [ n ] = ExtractNumber( str )

%EXTRACTNUMBER Summary of this function goes here

% Detailed explanation goes here

id = isstrprop(str,'digit');

if (~any(id))

n = NaN;

return

end

n\_end = find(id,1,'last');

id(n\_end:end) = true;

n\_beg = find(~id,1,'last')+1;

if (isempty(n\_beg))

n\_beg = 1;

end

n = str2num(str(n\_beg:n\_end));

end

%% test

%

% ExtractNumber('0501') == 501

% ExtractNumber('asd\_0502.jpg') == 502

% ExtractNumber('asdf1231asd2\_0501.jpg') == 501

% isnan(ExtractNumber('sadfasdf.jpg'))

## FilesInFolder.m

function filePaths = FilesInFolder( folderPath , imagesExt, doSort)

if ~exist('imagesExt','var')

imagesExt = 'jpg';

end

if ~exist('doSort','var')

doSort = false;

end

%% чтение папки с изображениями

if (~isempty(folderPath))

dirStructure = dir ([folderPath '\\*.' imagesExt]);

else

dirStructure = dir (['\*.' imagesExt]);

end

imagePathsUnsorted = {dirStructure.name}';

%сортировка

if (nargin > 2 && doSort)

fileNumbers = cellfun(@(x) ExtractNumber(x), imagePathsUnsorted,'UniformOutput', true);

[~, fileNumbersIdexes] = sort(fileNumbers);

filePaths = imagePathsUnsorted(fileNumbersIdexes);

else

filePaths = imagePathsUnsorted;

end

if (~isempty(folderPath))

filePaths = cellfun(@(x) [folderPath '\' x], filePaths,'UniformOutput', false);

end

end

Filter\_and\_interpolation\_params.m

classdef Filter\_and\_interpolation\_params

%UNTITLED3 Summary of this class goes here

% Detailed explanation goes here

properties

version = 1;

%% Фильтрация:

local\_filtering\_\_on = true;

CC\_ratio\_limits = [1.2 inf];

CC\_maxRaitio\_max = inf;

mean\_im1\_limits = [-inf inf];

xd\_limits = [-inf inf];

yd\_limits = [-inf inf];

d\_limits = [-inf inf];

%медианная фильтрация:

median\_space\_\_on = false;

median\_space\_\_window\_size = [3 3]; % размер окна (x y)

median\_space\_\_max\_d\_difference = 5; % максимальное отличие от медианного, px/кадр

%медианная фильтрация:

median\_time\_\_on = false;

median\_time\_\_max\_d\_difference = 5; % максимальное отличие от медианного, px/кадр

median\_time\_\_dynamic\_\_on = false;

median\_time\_\_dynamic\_\_window\_size = 5; % размер окна (ti)

%% Интерполяция:

interp\_space\_\_on = false;

interp\_space\_\_Method = 'linear'; % 'linear','nearest' , or 'natural'.

interp\_space\_\_ExtrapolationMethod = 'none'; %'nearest', 'linear', or 'none'.

interp\_time\_\_on = false;

interp\_time\_\_Method = 'linear'; % 'linear' | 'nearest' | 'next' | 'previous' | 'spline' | 'pchip' | 'cubic' | 'makima'

interp\_time\_\_max\_gap\_length = 5; % максимальный промеждуток, который мы интерполируем (в кадрах)

end

end

## Filter\_and\_interpolation\_processor.m

classdef Filter\_and\_interpolation\_processor < handle

%UNTITLED2 Summary of this class goes here

% Detailed explanation goes here

properties (Access = private)

p Filter\_and\_interpolation\_params %можно поменять только пересоздав

pf PIV\_frames

pg PIV\_grid

end

properties

% iteration\_time\_arr % время на итерацию (только фильтрация и интерполяция)

% iteration\_time\_full\_arr % полное время на итерацию (включая время между циклами)

% iteration\_time\_full\_tic % для полного времени

% % compared\_frames\_\_num\_of\_pairs

% is\_processed\_arr

%

% current\_ti = nan;

% current\_stage = 'none';

%%

data Array\_temp\_storage;

end

methods

function obj = Filter\_and\_interpolation\_processor( fi\_params, pf, pg,...

data\_storage\_file\_path\_base, data\_seg\_size )

obj.p = fi\_params;

obj.pf = pf;

obj.pg = pg;

% obj.compared\_frames\_\_num\_of\_pairs = size(obj.pf.compared\_frames\_offset,1);

obj.data = Array\_temp\_storage( ...

obj.pf.frame\_count,...

data\_seg\_size,...

data\_storage\_file\_path\_base);

% obj.Reset\_processing();

end

% function Reset\_processing(obj)

% obj.iteration\_time\_arr = nan(obj.pf.frame\_count,1);

% obj.iteration\_time\_full\_arr = nan(obj.pf.frame\_count,1);

% obj.iteration\_time\_full\_tic = [];

% obj.data.Clear();

% obj.is\_processed\_arr = false(obj.pf.frame\_count,1);

% end

%%

function Process\_all( obj, data\_piv )

for ti = 1:obj.pf.frame\_count

fprintf( 'stage 1 2, ti %d/%d\n', ti, obj.pf.frame\_count )

obj.Process\_one\_\_stage1\_\_local\_filtering( data\_piv, ti );

obj.Process\_one\_\_stage2\_\_median\_space\_filtering( ti );

end

obj.Process\_all\_median\_time();

obj.Process\_all\_\_interpolate\_time();

for ti = 1:obj.pf.frame\_count

fprintf( 'Process\_one\_\_interpolate\_space, ti %d/%d\n', ti, obj.pf.frame\_count )

obj.Process\_one\_\_interpolate\_space( ti );

end

% iteration\_time = tic;

% obj.current\_ti = ti;

% obj.is\_processed\_arr(ti) = true;

% obj.iteration\_time\_arr(ti) = toc(iteration\_time);

% if ~isempty(obj.iteration\_time\_full\_tic)

% obj.iteration\_time\_full\_arr(ti) = toc(obj.iteration\_time\_full\_tic);

% end

% obj.iteration\_time\_full\_tic = tic;

end

%%

function Process\_one\_\_stage1\_\_local\_filtering(obj, data\_piv, ti)

%загружает из piv, фильтрует локально и сохраняет в data (отфильтрованные смещения забивает нанами)

if isempty(data\_piv), return; end

po = data\_piv.Get(ti);

if isempty(po), return; end

xdispl = po.xdispl; %поля после фильтрации и интерполяции

ydispl = po.ydispl; %поля после фильтрации и интерполяции

status = zeros(size(po.status)); % статус фильтрации

%параметры:

local\_filtering\_\_on = obj.p.local\_filtering\_\_on;

CC\_ratio\_limits = obj.p.CC\_ratio\_limits;

CC\_maxRaitio\_max = obj.p.CC\_maxRaitio\_max;

mean\_im1\_limits = obj.p.mean\_im1\_limits;

xd\_limits = obj.p.xd\_limits;

yd\_limits = obj.p.yd\_limits;

d\_limits = obj.p.d\_limits;

if local\_filtering\_\_on

status(po.status ~= 0) = -po.status(po.status ~= 0) ; % отброшена на этапе PIV

status(po.CC\_maxRaitio < CC\_ratio\_limits(1)) = 2; % отброшена по соотношению пиков ККФ

status(po.CC\_maxRaitio > CC\_ratio\_limits(2)) = 2;

status(po.CC\_maxValue > CC\_maxRaitio\_max) = 3; % отброшена по значению пика ККФ

status(po.mean\_im1 < mean\_im1\_limits(1)) = 4; % отброшена по яркости изображения в окне

status(po.mean\_im1 > mean\_im1\_limits(2)) = 4;% если mean\_im1\_limits НаН, то фильтрации не произойдет

status(xdispl < xd\_limits(1)) = 5; % отброшена по превышению смещения

status(xdispl > xd\_limits(2)) = 5;

status(ydispl < yd\_limits(1)) = 5;

status(ydispl > yd\_limits(2)) = 5;

if ~all(isnan(d\_limits) | isinf(d\_limits))

v = sqrt(po.xdispl .^ 2 + po.ydispl .^ 2);

status(v < d\_limits(1)) = 5;

status(v > d\_limits(2)) = 5;

end

xdispl(status ~= 0) = nan;

ydispl(status ~= 0) = nan;

end

fi\_output.status = status;

fi\_output.xdispl = xdispl;

fi\_output.ydispl = ydispl;

obj.data.Set(ti, fi\_output);

end

%%

function Process\_one\_\_stage2\_\_median\_space\_filtering(obj, ti)

% фильтрация по отличию от медианной фильтрации по пространству

median\_space\_\_on = obj.p.median\_space\_\_on;

median\_space\_\_window\_size = obj.p.median\_space\_\_window\_size;

median\_space\_\_max\_d\_difference = obj.p.median\_space\_\_max\_d\_difference;

if median\_space\_\_on

fi\_output = obj.data.Get(ti);

if isempty(fi\_output), return; end

gp = fi\_output.status == 0;

xd = fi\_output.xdispl;

yd = fi\_output.ydispl;

xd(~gp) = nan;

yd(~gp) = nan;

xd3fs = nanmedfilt2( xd, median\_space\_\_window\_size );

yd3fs = nanmedfilt2( yd, median\_space\_\_window\_size );

bp = ...

abs(xd3fs - xd) > median\_space\_\_max\_d\_difference |...

abs(yd3fs - yd) > median\_space\_\_max\_d\_difference;

fi\_output.status(bp) = 6; % фильтрация по отличию от медианной фильтрации по пространству

fi\_output.xdispl( fi\_output.status ~= 0 ) = nan;

fi\_output.ydispl( fi\_output.status ~= 0 ) = nan;

obj.data.Set(ti, fi\_output);

end

end

function Process\_all\_median\_time(obj )

% работает с уже отфильтрованными данными

median\_time\_\_on = obj.p.median\_time\_\_on;

median\_time\_\_max\_d\_difference = obj.p.median\_time\_\_max\_d\_difference;

median\_time\_\_dynamic\_\_on = obj.p.median\_time\_\_dynamic\_\_on; %динамическая намного медленнее, но не требует много памяти

median\_time\_\_dynamic\_\_window\_size = obj.p.median\_time\_\_dynamic\_\_window\_size;

if median\_time\_\_on

if ~median\_time\_\_dynamic\_\_on

% вычисляем медианное поле для всей записи

fprintf( 'Get\_median\_field all...\n' );

[xd\_median, yd\_median] = Analyzer.Get\_median\_field(obj.data); % TODO жрет много памяти, вычислять отдельно по сегментам

% этот кусок уже память не жрет, но можно сделать

% быстрее за счет памяти

for ti = 1:obj.pf.frame\_count

fprintf( 'Process\_all\_median\_time, ti %d/%d\n', ti, obj.pf.frame\_count )

fi\_output = obj.data.Get(ti);

if isempty(fi\_output), continue; end

xd = fi\_output.xdispl;

yd = fi\_output.ydispl;

bp = ...

abs(xd\_median - xd) > median\_time\_\_max\_d\_difference |...

abs(yd\_median - yd) > median\_time\_\_max\_d\_difference;

fi\_output.status(bp) = 7; % фильтрация по отличию от медианной фильтрации по времени

fi\_output.xdispl(bp) = nan;

fi\_output.ydispl(bp) = nan;

obj.data.Set(ti, fi\_output);

end

else

for ti = 1:obj.pf.frame\_count

fprintf( 'Process\_all\_median\_time, ti %d/%d\n', ti, obj.pf.frame\_count )

% ti\_arr\_s = ti - ceil (median\_time\_\_dynamic\_\_window\_size/2);

% ti\_arr\_e = ti + floor(median\_time\_\_dynamic\_\_window\_size/2);

% ti\_arr\_s = max(ti\_arr\_s,1);

% ti\_arr\_e = min(ti\_arr\_e,obj.pf.frame\_count);

% ti\_arr = ti\_arr\_s : ti\_arr\_e;

[xd\_median, yd\_median] = Analyzer.Get\_median\_field(obj.data, [], [ ti median\_time\_\_dynamic\_\_window\_size] );

fi\_output = obj.data.Get(ti);

if isempty(fi\_output), continue; end

xd = fi\_output.xdispl;

yd = fi\_output.ydispl;

bp = ...

abs(xd\_median - xd) > median\_time\_\_max\_d\_difference |...

abs(yd\_median - yd) > median\_time\_\_max\_d\_difference;

fi\_output.status(bp) = 7; % фильтрация по отличию от медианной фильтрации по времени

fi\_output.xdispl(bp) = nan;

fi\_output.ydispl(bp) = nan;

obj.data.Set(ti, fi\_output);

end

end

end

end

%%

function Process\_all\_\_interpolate\_time(obj)

% интерполяция по времени на основе всех не нан смещений

interp\_time\_\_on = obj.p.interp\_time\_\_on;

interp\_time\_\_Method = obj.p.interp\_time\_\_Method;

interp\_time\_\_max\_gap\_length = obj.p.interp\_time\_\_max\_gap\_length;

if interp\_time\_\_on

% TODO жрет много памяти, вычислять отдельно по

% сегментам

[fi\_output\_all, data\_is\_empty\_arr] = obj.data.Get\_all\_as\_cell();

gp\_t = ~data\_is\_empty\_arr; %номера полей, для которые поля посчитаны

xd\_mat3 = cellfun(@(x) x.xdispl, fi\_output\_all(gp\_t),'uni',false); % склеиваем только подсчитанные поля

yd\_mat3 = cellfun(@(x) x.ydispl, fi\_output\_all(gp\_t),'uni',false);

xd\_mat3 = cat(3,xd\_mat3{:});

yd\_mat3 = cat(3,yd\_mat3{:});

bp\_mat3 = isnan(xd\_mat3) | isnan(yd\_mat3);

t = obj.pf.first\_frames(gp\_t); % на случай неэквидистанстный фреймов или недосчитанных данных

for i = 1:size(xd\_mat3,1) %todo parfor

for j = 1:size(xd\_mat3,2)

fprintf( 'Process\_all\_\_interpolate\_time, i %d/%d, j %d/%d\n', i, size(xd\_mat3,1), j, size(xd\_mat3,2) )

xd\_ts = squeeze(xd\_mat3(i,j,:));

yd\_ts = squeeze(yd\_mat3(i,j,:));

bp = squeeze(bp\_mat3(i,j,:));

gp = ~bp;

if ~any(bp(:)), continue; end % ничего не надо интерполировать

if sum(gp(:)) < 3, continue; end % не с чего интерполировать

xd\_ts(bp) = interp1gap(t(gp), xd\_ts(gp), t(bp), interp\_time\_\_max\_gap\_length, interp\_time\_\_Method);

yd\_ts(bp) = interp1gap(t(gp), yd\_ts(gp), t(bp), interp\_time\_\_max\_gap\_length, interp\_time\_\_Method);

% xd\_ts(bp) = interp1(t(gp), xd\_ts(gp), t(bp), interp\_time\_\_Method, 'extrap');

% yd\_ts(bp) = interp1(t(gp), yd\_ts(gp), t(bp), interp\_time\_\_Method, 'extrap');

xd\_mat3(i,j,:) = reshape( xd\_ts, [ 1 1 numel(xd\_ts) ] );

yd\_mat3(i,j,:) = reshape( yd\_ts, [ 1 1 numel(yd\_ts) ] );

end

end

ti\_arr = find(gp\_t);

for ti\_arr\_i = 1:numel(ti\_arr)% только для подсчитанных полей

ti = ti\_arr(ti\_arr\_i);

fi\_output = obj.data.Get(ti);

if isempty(fi\_output), continue; end

fi\_output.xdispl = xd\_mat3(:,:,ti\_arr\_i);

fi\_output.ydispl = yd\_mat3(:,:,ti\_arr\_i);

% точки, которые были интерполированы:

interp\_points = bp\_mat3(:,:,ti\_arr\_i) & ~isnan(fi\_output.xdispl) & ~isnan(fi\_output.ydispl);

fi\_output.status(interp\_points) = 10; % точки, вычисленные в результате интерполяции по времени

obj.data.Set( ti, fi\_output );

end

end

end

function Process\_one\_\_interpolate\_space(obj, ti)

% интерполяция по пространству на основе всех не нан смещений

interp\_space\_\_on = obj.p.interp\_space\_\_on;

interp\_space\_\_Method = obj.p.interp\_space\_\_Method;

interp\_space\_\_ExtrapolationMethod = obj.p.interp\_space\_\_ExtrapolationMethod;

if interp\_space\_\_on

fi\_output = obj.data.Get(ti);

if isempty(fi\_output), return; end

xdispl = fi\_output.xdispl;

ydispl = fi\_output.ydispl;

gp = ~isnan(xdispl) & ~isnan(ydispl);

% if sum(gp(:)) > 2

bp = ~gp;

if ~any(bp(:)), return; end % ничего не надо интерполировать

if ~any(gp(:)), return; end % не с чего интерполировать

SI\_xdispl = scatteredInterpolant( obj.pg.xMat(gp), obj.pg.yMat(gp), fi\_output.xdispl(gp));

SI\_ydispl = scatteredInterpolant( obj.pg.xMat(gp), obj.pg.yMat(gp), fi\_output.ydispl(gp));

SI\_xdispl.Method = interp\_space\_\_Method;

SI\_ydispl.Method = interp\_space\_\_Method;

SI\_xdispl.ExtrapolationMethod = interp\_space\_\_ExtrapolationMethod;

SI\_ydispl.ExtrapolationMethod = interp\_space\_\_ExtrapolationMethod;

xdispl\_q = SI\_xdispl( obj.pg.xMat(bp), obj.pg.yMat(bp) );

ydispl\_q = SI\_ydispl( obj.pg.xMat(bp), obj.pg.yMat(bp) );

if isempty(xdispl\_q) || isempty(ydispl\_q)

warning('Process\_one\_\_interpolate\_space: Интерполяция не удалась')

return;

end

xdispl(bp) = xdispl\_q;

ydispl(bp) = ydispl\_q;

% точки, которые были интерполированы:

interp\_points = bp & ~isnan(xdispl) & ~isnan(ydispl);

fi\_output.status(interp\_points) = 11; % точки, вычисленные в результате интерполяции по пространству

fi\_output.xdispl = xdispl;

fi\_output.ydispl = ydispl;

obj.data.Set(ti, fi\_output);

end

end

%%

function info = GetInfo(obj)

% TODO жрет много памяти, вычислять отдельно по

% сегментам

[fi\_output\_all, data\_is\_empty\_arr] = obj.data.Get\_all\_as\_cell();

status = cellfun(@(x) x.status, fi\_output\_all(~data\_is\_empty\_arr),'uni',false);

status = cat(3,status{:});

a = status(:);

info.status\_list = unique(a);

info.status\_histc = histc(a, info.status\_list);

end

end

end

## Image\_loader.m

classdef Image\_loader < handle

%IMLOADER Класс загрузки изображений

% bufferSize - размер буфера, можно не указывать

properties

path;

imagePaths;

bufferSize;

indexArray;

imArray;

imSize;

imNum;

% свойста, которые стоит менять после конструктора:

use\_ind2gray; %индексное изображение, для файлов которые пишет TimeBench

end

methods

function il = Image\_loader( impath, imagesExt, bufferSize )

if nargin > 0

% disp('Reading Folder....')

if ~exist(impath,'dir'), error('=== no folder === '); end

tic

il.path = impath;

if ~exist('imagesExt','var') || isempty(imagesExt)

d = dir ([impath '\\*']);

d = d(4);

[~,~,imagesExt] = fileparts(d.name);

end

if imagesExt(1) == '.', imagesExt = imagesExt(2:end); end

if ~exist('bufferSize','var') || isempty(bufferSize), bufferSize = 10; end

il.bufferSize = bufferSize;

%% чтение папки с изображениями

% imagesExt = 'jpg';

il.imagePaths = FilesInFolder( il.path , imagesExt, true);

if isempty(il.imagePaths), error('Пустая папка'); end

il.indexArray = zeros(il.bufferSize,1);

il.imArray = cell(il.bufferSize,1);

il.imNum = numel(il.imagePaths);

im = imread(il.imagePaths{1});

il.imSize = fliplr(size(im(:,:,1)));

% fprintf('File info loaded in %g sec\nTotal number of frames %d\n',toc, il.imNum);

else

error('!!!! need path in constructor');

end

end

function im = getImage(obj, i)

if (obj.bufferSize > 0)

ii = find(obj.indexArray == i);

if ~isempty(ii)

im = obj.imArray{ii};

return;

end

end

if isempty(obj.use\_ind2gray)

try

im = imread(obj.imagePaths{i});

catch

im = [];

return;

end

else

[im, rgbmap] = imread(obj.imagePaths{i});

im = ind2gray(im,rgbmap);

end

% буфферизация:

if (obj.bufferSize > 0)

obj.indexArray = circshift(obj.indexArray,1);

obj.imArray = circshift(obj.imArray,1);

obj.indexArray(1) = i;

obj.imArray{1} = im;

end

end

function imPath = getImagePath(obj,i)

imPath = obj.imagePaths{i};

end

function ResetCache(obj)

obj.indexArray = zeros(obj.bufferSize,1);

obj.imArray = cell(obj.bufferSize,1);

end

end

end

## Image\_loader\_params.m

classdef Image\_loader\_params

%Rec\_params Summary of this class goes here

% Detailed explanation goes here

properties

version = 1;

images\_folder

images\_ext = 'jpg';

end

end

## Image\_preprocessor.m

classdef Image\_preprocessor < handle

%IMLOADER Класс загрузки изображений

% bufferSize - размер буфера, можно не указывать

properties (Access = private)

ip

imSize;

imNum;

bufferSize

indexArray;

imArray;

params Image\_preprocessor\_params

end

methods

function obj = Image\_preprocessor( ip, params, bufferSize )

obj.ip = ip;

obj.imNum = ip.imNum;

obj.imSize = ip.imSize;

obj.params = params;

obj.bufferSize = bufferSize;

% очистить буффер

obj.indexArray = zeros(obj.bufferSize,1);

obj.imArray = cell(obj.bufferSize,1);

end

%%

function im = getImage(obj, i)

if (obj.bufferSize > 0)

ii = find(obj.indexArray == i);

if ~isempty(ii)

im = obj.imArray{ii};

return;

end

end

im = obj.ip.getImage(i);

im = im(:,:,obj.params.color\_channel);

if obj.params.bg\_on && ~isempty(obj.params.bg)

switch obj.params.bg\_cut\_type

case 'max'

im = max(obj.params.bg\_cut\_val, double(im) - double(obj.params.bg));

case 'min'

im = min(obj.params.bg\_cut\_val, double(im) - double(obj.params.bg));

case 'none'

im = double(im) - double(obj.bg);

otherwise

warning('Unexpected bg\_cut\_type');

im = double(im) - double(obj.bg);

end

end

if obj.params.mask\_on && ~isempty(obj.params.mask)

im( obj.params.mask ) = obj.params.mask\_set\_value;

end

if obj.params.levels\_on

im = mat2gray(im,obj.params.levels\_limits);

end

% буфферизация:

if (obj.bufferSize > 0)

obj.indexArray = circshift(obj.indexArray,1);

obj.imArray = circshift(obj.imArray,1);

obj.indexArray(1) = i;

obj.imArray{1} = im;

end

end

end

end

## Image\_preprocessor\_params.m

classdef Image\_preprocessor\_params

%UNTITLED Summary of this class goes here

% Detailed explanation goes here

properties

version = 1;

%%

color\_channel = 1;

%% bg

bg

bg\_on = false;

bg\_cut\_type = 'max';% min или max или другое (тогда не будет отрезаться)

bg\_cut\_val = 0;

bg\_source\_file\_path = "";

bg\_source\_auto\_prctile = 1;

bg\_source\_auto\_N = 10; %число сэмплов

%% mask

mask

mask\_on = false;

mask\_set\_value = 0; % пиксели заменяются на это значение

mask\_source\_path = "";

mask\_source\_color = [255 0 0]; % пиксели загружаемой картинки с таким цветом считаются маской

%% levels

levels\_on = false;

levels\_limits = [0 100];

end

end

## My\_importfile\_xls\_with\_headers\_to\_table\_v1.m

function [table\_data, data, raw\_data] = My\_importfile\_xls\_with\_headers\_to\_table\_v1(workbookFile, xlRange, row\_num\_headers, row\_num\_data\_start)

% reads xlsx file with headers

% workbookFile - file path

% example: rec\_list = My\_importfile\_xls\_with\_headers\_to\_table\_v1('..\список записей 1.xlsx',[],1,3));

%%

[data, headers, raw\_data] = My\_importfile\_xls\_with\_headers\_v1(workbookFile, xlRange, row\_num\_headers, row\_num\_data\_start);

%где стринги заменим наны на пустую строку

d\_is\_str = cellfun(@(x) ischar(x), data);

d\_is\_nan = cellfun(@(x) any(isnan(x)), data);

d\_is\_num = cellfun(@(x) isnumeric(x), data);

d\_col\_str = repmat(any(d\_is\_str,1),[size(data,1),1]); %если в колонке хотя бы одна текстовая строка

data( d\_col\_str & d\_is\_nan ) = {''}; % если в колонке есть текст, то всё будет текстом

data( d\_col\_str & d\_is\_num ) = cellfun(@(x) num2str(x), data( d\_col\_str & d\_is\_num ), 'uni', false); % если в колонке есть текст, то всё будет текстом

try

table\_data = cell2table( data, 'variablenames', headers );

catch

disp(headers);

error('в первой строке файла должны быть валидные имена переменных для матлаба')

end

My\_importfile\_xls\_with\_headers\_v1.m

function [data, headers, raw\_data] = My\_importfile\_xls\_with\_headers\_v1(workbookFile, xlRange, row\_num\_headers, row\_num\_data\_start)

% reads xlsx file with headers

% workbookFile - file path

% example: rec\_list = importfile\_reclist\_v2('..\ADC records list.xlsx');

sheetName = 1;

if exist('xlRange','var') && ~isempty(xlRange)

[~, ~, raw\_data] = xlsread(workbookFile, sheetName, xlRange);

else

[~, ~, raw\_data] = xlsread(workbookFile, sheetName);

end

headers = raw\_data(row\_num\_headers,:);

data = raw\_data(row\_num\_data\_start:end,:);

## PIV\_frames.m

classdef PIV\_frames

%UNTITLED2 Summary of this class goes here

% Detailed explanation goes here

% compared\_frames\_offset %сравниваемые кадры: например [ 0 1; 0 2; 0 3 ] - первый со вторым, первый с третьим, первый с четвертым

% firstFrames % номера первых кадров сравнения

properties (SetAccess = protected, GetAccess = public)

version = 1;

frame\_start

frame\_end

frame\_skip

frame\_step

first\_frames

compared\_frames\_offset

frame\_count

end

methods

function obj = PIV\_frames(frame\_start, frame\_end, frame\_skip, frame\_step)

% frame\_start - с какого кадра начинать

% frame\_end - до какого кадра считать (полей будет меньше, т.к. он будет вторым сравниваемым кадров)

% frame\_skip: 1 - каждый первый, 2 - каждый второй....

% frame\_step: через сколько кадров сравнивать, 1 - соседние, 2 - через 1

obj.frame\_start = frame\_start;

obj.frame\_end = frame\_end;

obj.frame\_skip = frame\_skip;

obj.frame\_step = frame\_step;

obj.first\_frames = frame\_start : frame\_skip : ( frame\_end - frame\_step );

obj.compared\_frames\_offset = [0 frame\_step];

obj.frame\_count = numel(obj.first\_frames);

end

end

end

% if (max(firstFrames(:)) + max(compared\_frames\_offset(:)) > ip.imNum)

% error('надо будет брать кадры с номером больше чем число изображений');

% end

## PIV\_grid.m

classdef PIV\_grid

%квадратная сетка для PIV

% Detailed explanation goes here

properties

version = 1;

xMat;

yMat; %(xi yi)

xArr;

yArr;

xStep;

yStep; %auto

do\_PIV\_mat;

end

methods

function obj = PIV\_grid(xArr, yArr)

if nargin == 2

[obj.yMat,obj.xMat] = meshgrid(yArr,xArr);

obj.xArr = xArr;

obj.yArr = yArr;

obj.do\_PIV\_mat = true(size(obj.xMat));

if numel(xArr) > 1

d = diff(xArr);

if any(d ~= d(1))

obj.xStep = NaN;

else

obj.xStep = d(1);

end

else

obj.xStep = NaN;

end

if numel(yArr) > 1

d = diff(yArr);

if any(d ~= d(1))

obj.yStep = NaN;

else

obj.yStep = d(1);

end

else

obj.yStep = NaN;

end

else

error ('need 2 arguments')

end

end

function s = Get\_size(obj)

s = size(obj.xMat);

end

end

methods(Static)

function obj = Get\_max\_grid(im\_size\_xy, step\_xy)

im\_size\_x = im\_size\_xy(1);

im\_size\_y = im\_size\_xy(2);

st\_x = step\_xy(1);

st\_y = step\_xy(2);

s\_x = ceil(st\_x + 1);

s\_y = ceil(st\_y + 1);

e\_x = floor(im\_size\_x - st\_x + 1);

e\_y = floor(im\_size\_y - st\_y + 1);

obj = PIV\_grid(...

s\_x : st\_x : e\_x,...

s\_y : st\_y : e\_y);

end

function result = RectFitsin(imArea,imSize)

[xS, xE, yS, yE] = PIV\_grid.BoardersOfArea(imArea);

imSizeX = imSize(1);

imSizeY = imSize(2);

result = (xS >= 1 && xE <= imSizeX && yS >= 1 && yE <= imSizeY);

end

function imArea = RectArea(c,wSize)

xc = c(1);

yc = c(2);

xS = xc - round(wSize(1)/2);

xE = xc + round(wSize(1)/2)-1;

yS = yc - round(wSize(2)/2);

yE = yc + round(wSize(2)/2)-1;

imArea = round([xS xE yS yE]);

end

function [xS, xE, yS, yE] = BoardersOfArea(imArea)

xS = imArea(1);

xE = imArea(2);

yS = imArea(3);

yE = imArea(4);

end

end

methods

function obj = Disable\_from\_mask(obj, mask, wSize, max\_prc)

%Disable\_from\_mask выключает узлы, в которых слишком много

%маски

imSize = fliplr(size(mask));

xcArr = obj.xMat;

ycArr = obj.yMat;

xFieldCount = size(xcArr,1);

yFieldCount = size(xcArr,2);

for xi = 1:xFieldCount

for yi = 1:yFieldCount

xc = xcArr(xi,yi);

yc = ycArr(xi,yi);

if isnan(xc) || isnan(yc)

continue;

end

c = [xc, yc]; %центр

imArea1 = obj.RectArea(c,wSize);

if ~obj.RectFitsin(imArea1,imSize)

continue;

end

[xS1, xE1, yS1, yE1] = obj.BoardersOfArea(imArea1);

im1c = mask(yS1:yE1, xS1:xE1);

mask\_prc = 100 \* sum(im1c(:)) / numel(im1c);

if mask\_prc > max\_prc

obj.do\_PIV\_mat(xi,yi) = false;

end

end

end

end

end

methods (Static)

function overlap\_xy = Get\_overlap(window\_size, step\_xy)

overlap\_xy(1) = 100\* (window\_size(1) - step\_xy(1)) / window\_size(1);

overlap\_xy(2) = 100\*(window\_size(2) - step\_xy(2)) / window\_size(2);

end

function step\_xy = Get\_step\_from\_overlap(window\_size, overlap)

step\_xy(1) = window\_size(1) - overlap/100 \* window\_size(1);

step\_xy(2) = window\_size(2) - overlap/100 \* window\_size(2);

end

end

end

## PIV\_main.m

classdef PIV\_main < handle

%PIV\_APP Summary of this class goes here

% Detailed explanation goes here

properties

core Core

piv\_plot PIV\_plot

%%

p Project\_params

%%

proj\_path %путь к файлу проекта

data\_seg\_size = 100;

current\_frame = 1;

current\_ti = nan;

fig\_fields;

ax\_fields;

ax\_process;

ax\_analyze;

% gui\_app PIV\_main\_App\_v5

%%

selected\_points; %x\_px, y\_px, x\_m, y\_m

end

methods

function obj = PIV\_main()

%PIV\_APP Construct an instance of this class

% Detailed explanation goes here

obj.piv\_plot = PIV\_plot();

obj.p = Project\_params();

obj.proj\_path = '';

end

function delete(obj) %при удалении объекта удалять временные данные, закрывать фигуры

if ~isempty(obj.fig\_fields) && ishandle(obj.fig\_fields)

close(obj.fig\_fields);

end

if ~isempty(obj.core) && ~isempty(obj.core.piv\_processor) && ~isempty(obj.core.piv\_processor.data)

obj.core.piv\_processor.data.Clear();

obj.core.piv\_processor.ID\_data.Clear();

end

if ~isempty(obj.core) && ~isempty(obj.core.fi\_processor) && ~isempty(obj.core.fi\_processor.data)

obj.core.fi\_processor.data.Clear();

end

end

%%

function Proj\_new( obj, images\_folder, images\_ext, proj\_path )

obj.Add\_to\_log(sprintf('Создание нового проекта...'));

obj.proj\_path = proj\_path;

obj.p.rp = Record\_params();

obj.p.il = Image\_loader\_params();

obj.p.il.images\_folder = images\_folder;

obj.p.il.images\_ext = images\_ext;

il = Image\_loader( obj.p.il.images\_folder, obj.p.il.images\_ext, 10 );

obj.p.ipp = Image\_preprocessor\_params();

ipp = Image\_preprocessor( il, obj.p.ipp, 10 );

obj.p.pp = PIV\_params();

obj.p.pg = PIV\_grid.Get\_max\_grid(il.imSize,obj.p.pp.wSize/2);

obj.p.pf = PIV\_frames(1,il.imNum,1,1);

obj.p.fi = Filter\_and\_interpolation\_params();

obj.p.ap = Analyzer\_params();

obj.p.plot\_params = Plot\_params();

piv\_processor = PIV\_processor( obj.p.pp, obj.p.pf, obj.p.pg,...

[ obj.proj\_path ' - temp storage\piv\'], obj.data\_seg\_size);

fi\_processor = Filter\_and\_interpolation\_processor( obj.p.fi, obj.p.pf, obj.p.pg,...

[ obj.proj\_path ' - temp storage\fi\' ], obj.data\_seg\_size);

transform\_processor = Transform\_processor( obj.p.rp, obj.p.pg, obj.p.pf );

exporter = Exporter();

analyzer = Analyzer( obj.p.ap, obj.p.rp, obj.p.pp, obj.p.pf, obj.p.pg, fi\_processor.data );

obj.core = Core( il, ipp, piv\_processor, fi\_processor, transform\_processor, exporter, analyzer );

obj.Go\_to\_frame(1);

obj.selected\_points = [];

obj.Add\_to\_log(sprintf('Создание нового проекта завершено'));

end

function Proj\_save(obj, proj\_path)

obj.Add\_to\_log(sprintf('Сохранение проекта...'));

if exist('proj\_path','var')

obj.proj\_path = proj\_path;

end

p = obj.p;

if ~exist(fileparts(obj.proj\_path),'dir'), mkdir(fileparts(obj.proj\_path)); end

save(obj.proj\_path, 'p','-mat');

obj.core.piv\_processor.data.Export\_to\_file( [obj.proj\_path ' data piv'] );

obj.core.fi\_processor. data.Export\_to\_file( [obj.proj\_path ' data fi'] );

obj.Add\_to\_log(sprintf('Сохранение проекта завершено'));

end

function Proj\_load(obj, proj\_path)

obj.Add\_to\_log(sprintf('Загрузка проекта...'));

obj.proj\_path = proj\_path;

f = load(proj\_path,'-mat');

obj.p = f.p;

il = Image\_loader( obj.p.il.images\_folder, obj.p.il.images\_ext, 10 );

ipp = Image\_preprocessor( il, obj.p.ipp, 10 );

piv\_processor = PIV\_processor( obj.p.pp, obj.p.pf, obj.p.pg,...

[ obj.proj\_path ' - temp storage\piv\' ], obj.data\_seg\_size);

fi\_processor = Filter\_and\_interpolation\_processor( obj.p.fi, obj.p.pf, obj.p.pg,...

[ proj\_path ' - temp storage\fi\' ], obj.data\_seg\_size);

transform\_processor = Transform\_processor( obj.p.rp, obj.p.pg, obj.p.pf );

exporter = Exporter();

analyzer = Analyzer( obj.p.ap, obj.p.rp, obj.p.pp, obj.p.pf, obj.p.pg, fi\_processor.data );

obj.core = Core( il, ipp, piv\_processor, fi\_processor, transform\_processor, exporter, analyzer );

obj.core.piv\_processor.data.Import\_from\_file( [obj.proj\_path ' data piv'] );

obj.core.fi\_processor. data.Import\_from\_file( [obj.proj\_path ' data fi' ] );

obj.Go\_to\_frame(1);

obj.selected\_points = [];

obj.Add\_to\_log(sprintf('Загрузка проекта завершена'));

end

%%

function Set\_il\_params(obj,il\_params)

if ... %если все совпадает

strcmp( obj.p.il.images\_folder, il\_params.images\_folder) && ...

strcmp( obj.p.il.images\_ext, il\_params.images\_ext)

return;

end

obj.p.il = il\_params;

obj.core.il = Image\_loader( obj.p.il.images\_folder, obj.p.il.images\_ext, 10 );

obj.core.ipp = Image\_preprocessor( obj.core.il, obj.p.ipp, 10 );

if obj.p.pf.frame\_end > obj.core.il.imNum

pf = PIV\_frames(obj.p.pf.frame\_start, obj.core.il.imNum, obj.p.pf.frame\_skip, obj.p.pf.frame\_step);

obj.Set\_pf(pf);

end

end

function Set\_ipp\_params( obj, ipp\_params )

if ipp\_params.bg\_on && isempty(ipp\_params.bg)

ipp\_params.bg\_on = false;

end

if ipp\_params.mask\_on && isempty(ipp\_params.mask)

ipp\_params.mask\_on = false;

end

obj.p.ipp = ipp\_params;

obj.core.ipp = Image\_preprocessor( obj.core.il, obj.p.ipp, 10 );

end

function Set\_rp\_params( obj, rp )

obj.p.rp = rp;

obj.core.transform\_processor = Transform\_processor( obj.p.rp, obj.p.pg, obj.p.pf );

obj.core.analyzer = Analyzer( obj.p.ap, obj.p.rp, obj.p.pp, obj.p.pf, obj.p.pg, obj.core.fi\_processor.data );

end

function Set\_pp( obj, pp )

obj.p.pp = pp;

obj.core.piv\_processor = PIV\_processor( obj.p.pp, obj.p.pf, obj.p.pg,...

[ obj.proj\_path ' - temp storage\piv\' ], obj.data\_seg\_size );

obj.core.fi\_processor = Filter\_and\_interpolation\_processor( obj.p.fi, obj.p.pf, obj.p.pg,...

[ obj.proj\_path ' - temp storage\fi\' ], obj.data\_seg\_size);

obj.core.analyzer = Analyzer( obj.p.ap, obj.p.rp, obj.p.pp, obj.p.pf, obj.p.pg, obj.core.fi\_processor.data );

end

function Set\_pg( obj, p\_pg )

obj.p.pg = p\_pg;

obj.core.piv\_processor = PIV\_processor( obj.p.pp, obj.p.pf, obj.p.pg,...

[ obj.proj\_path ' - temp storage\piv\' ], obj.data\_seg\_size );

obj.core.fi\_processor = Filter\_and\_interpolation\_processor( obj.p.fi, obj.p.pf, obj.p.pg,...

[ obj.proj\_path ' - temp storage\fi\' ], obj.data\_seg\_size);

obj.core.transform\_processor = Transform\_processor( obj.p.rp, obj.p.pg, obj.p.pf );

obj.core.analyzer = Analyzer( obj.p.ap, obj.p.rp, obj.p.pp, obj.p.pf, obj.p.pg, obj.core.fi\_processor.data );

end

function Set\_pf( obj, p\_pf )

obj.p.pf = p\_pf;

obj.core.piv\_processor = PIV\_processor( obj.p.pp, obj.p.pf, obj.p.pg,...

[ obj.proj\_path ' - temp storage\piv\' ], obj.data\_seg\_size );

obj.core.fi\_processor = Filter\_and\_interpolation\_processor( obj.p.fi, obj.p.pf, obj.p.pg,...

[ obj.proj\_path ' - temp storage\fi\' ], obj.data\_seg\_size);

obj.core.transform\_processor = Transform\_processor( obj.p.rp, obj.p.pg, obj.p.pf );

obj.core.analyzer = Analyzer( obj.p.ap, obj.p.rp, obj.p.pp, obj.p.pf, obj.p.pg, obj.core.fi\_processor.data );

end

function Set\_fi\_params( obj, fi\_params )

obj.p.fi = fi\_params;

obj.core.fi\_processor = Filter\_and\_interpolation\_processor( obj.p.fi, obj.p.pf, obj.p.pg,...

[ obj.proj\_path ' - temp storage\fi\' ], obj.data\_seg\_size);

obj.core.analyzer = Analyzer( obj.p.ap, obj.p.rp, obj.p.pp, obj.p.pf, obj.p.pg, obj.core.fi\_processor.data );

end

function Set\_ap( obj, ap )

obj.p.ap = ap;

obj.core.analyzer = Analyzer( obj.p.ap, obj.p.rp, obj.p.pp, obj.p.pf, obj.p.pg, obj.core.fi\_processor.data );

end

function Set\_plot\_params( obj, plot\_params )

obj.p.plot\_params = plot\_params;

end

%TODO остальные сеты

%%

function Add\_to\_log(obj, text)

fprintf('%s > %s\n', datestr(datetime('now'),'HH:MM:SS'), text);

end

%%

function Go\_to\_frame(obj,frame)

if isempty(obj.core), return; end

obj.current\_frame = max(1,min(obj.core.il.imNum,frame));

ti = find(obj.p.pf.first\_frames == obj.current\_frame,1);

if ~isempty(ti)

obj.current\_ti = ti;

else

obj.current\_ti = nan;

end

end

function Go\_to\_ti(obj,ti)

if isempty(obj.core), return; end

obj.current\_ti = max(1,min(obj.p.pf.frame\_count, ti));

obj.current\_frame = obj.p.pf.first\_frames(obj.current\_ti);

end

function n = ti\_count(obj)

n = 0;

if isempty(obj.core), return; end

if isempty(obj.p), return; end

n = obj.p.pf.frame\_count;

end

function n = im\_count(obj)

n = 0;

if isempty(obj.core), return; end

if isempty(obj.core.il), return; end

n = obj.core.il.imNum;

end

function n = im\_size(obj, dim)

if ~exist('dim','var')

n = obj.core.il.imSize;

else

n = obj.core.il.imSize(dim);

end

end

%%

function Process\_piv\_current(obj)

if isempty(obj.core), return; end

if isnan(obj.current\_ti), return; end

% obj.Add\_to\_log('Вычисление...');

% obj.core.piv\_processor.iteration\_time\_full\_tic = tic;

obj.core.piv\_processor.Process\_one( obj.core.ipp, obj.current\_ti );

obj.Add\_to\_log(obj.core.piv\_processor.Get\_info());

end

function Process\_piv\_all(obj,...

redraw\_fields\_on, redraw\_fields\_step,...

redraw\_process\_on, redraw\_process\_step,...

autosave\_on, autosave\_step)

if ~exist('redraw\_fields\_on', 'var'), redraw\_fields\_on = false; end

if ~exist('redraw\_fields\_step', 'var'), redraw\_fields\_step = 1; end

if ~exist('redraw\_process\_on', 'var'), redraw\_process\_on = false; end

if ~exist('redraw\_process\_step','var'), redraw\_process\_step = 1; end

if ~exist('autosave\_on', 'var'), autosave\_on = false; end

if ~exist('autosave\_step', 'var'), autosave\_step = 100; end

obj.Process\_piv\_reset();

for ti = 1:obj.p.pf.frame\_count

obj.Go\_to\_ti(ti);

obj.Process\_piv\_current();

if redraw\_fields\_on && rem(ti, redraw\_fields\_step) == 0

obj.Redraw\_fields();

drawnow();

end

if redraw\_process\_on && rem(ti, redraw\_process\_step) == 0

obj.Redraw\_process('piv\_processor');

drawnow();

end

if autosave\_on && rem(ti, autosave\_step) == 0

obj.Proj\_save();

end

end

end

function Process\_piv\_reset(obj)

obj.core.piv\_processor.Reset\_processing();

% obj.Redraw\_fields();

% obj.Redraw\_process('piv\_processor');

drawnow();

end

function is\_processed\_arr = Process\_piv\_\_is\_processed\_arr(obj)

is\_processed\_arr = obj.core.piv\_processor.is\_processed\_arr;

end

function info = Process\_piv\_\_Get\_info\_progress(obj)

info = obj.core.piv\_processor.Get\_info\_progress();

end

function info = Process\_piv\_\_Get\_info(obj)

info = obj.core.piv\_processor.Get\_info();

end

%%

function Process\_fi\_all(obj)

obj.Add\_to\_log('Запущена фильтрация и интерполяция');

obj.core.fi\_processor.Process\_all(obj.core.piv\_processor.data);

obj.Add\_to\_log('Фильтрация и интерполяция завершена');

end

%%

function New\_fig\_fields(obj)

obj.fig\_fields = figure('name','Поля','WindowButtonDownFcn',@wbdcb);

% ah = axes('SortMethod','childorder');

obj.ax\_fields = gca;

function wbdcb(src,~)

seltype = src.SelectionType;

if strcmp(seltype,'normal')

% src.Pointer = 'circle';

cp = obj.ax\_fields.CurrentPoint;

xinit = cp(1,1);

yinit = cp(1,2);

if ~obj.p.plot\_params.plot\_in\_scale

x\_px = xinit;

y\_px = yinit;

x\_m = obj.core.transform\_processor.Convert\_x(xinit);

y\_m = obj.core.transform\_processor.Convert\_y(yinit);

else

x\_px = obj.core.transform\_processor.Convert\_x\_to\_px(xinit);

y\_px = obj.core.transform\_processor.Convert\_y\_to\_px(yinit);

x\_m = xinit;

y\_m = yinit;

end

text\_m = sprintf('%f, %f m', x\_m, y\_m );

text\_px = sprintf('%.0f, %.0f px', x\_px, y\_px);

obj.Add\_to\_log(sprintf('(%s) (%s)', text\_m, text\_px));

obj.selected\_points = [ obj.selected\_points; x\_px, y\_px, x\_m, y\_m ];

obj.Redraw\_fields();

end

if strcmp(seltype,'alt')

cp = obj.ax\_fields.CurrentPoint;

xinit = cp(1,1);

yinit = cp(1,2);

if ~obj.p.plot\_params.plot\_in\_scale

x\_px = xinit;

y\_px = yinit;

x\_m = obj.core.transform\_processor.Convert\_x(xinit);

y\_m = obj.core.transform\_processor.Convert\_y(yinit);

else

x\_px = obj.core.transform\_processor.Convert\_x\_to\_px(xinit);

y\_px = obj.core.transform\_processor.Convert\_y\_to\_px(yinit);

x\_m = xinit;

y\_m = yinit;

end

selected\_points\_\_dsitance = sqrt( (obj.selected\_points(:,1) - x\_px) .^2 + (obj.selected\_points(:,2) - y\_px) .^2 );

selected\_points\_\_to\_delete = selected\_points\_\_dsitance < 10;

obj.selected\_points(selected\_points\_\_to\_delete,:) = [];

obj.Add\_to\_log(sprintf('Удалено точек: (%d)', selected\_points\_\_to\_delete));

obj.Redraw\_fields();

end

if strcmp(seltype,'extend')

obj.selected\_points = [ ];

obj.Redraw\_fields();

end

end

end

%%

function Redraw\_fields(obj)

if isempty(obj.core), return; end

if isempty(obj.fig\_fields) || ~ishandle(obj.fig\_fields)

obj.New\_fig\_fields();

end

if isempty(obj.ax\_fields), return; end

if ~ishandle(obj.ax\_fields), return; end

if isempty(obj.p.plot\_params), return; end

ax = obj.ax\_fields;

cla(ax,'reset');

reset(ax);

% пределы построения (края изображения)

XData = [1 obj.im\_size(1)];

YData = [1 obj.im\_size(2)];

if obj.p.plot\_params.plot\_in\_scale

XData = obj.core.transform\_processor.Convert\_x(XData);

YData = obj.core.transform\_processor.Convert\_y(YData);

end

x = obj.p.pg.xMat;

y = obj.p.pg.yMat;

frame = obj.current\_frame;

ti = obj.current\_ti;

time = obj.core.transform\_processor.Convert\_t(frame);

if obj.p.plot\_params.plot\_in\_scale

x = obj.core.transform\_processor.Convert\_x(x);

y = obj.core.transform\_processor.Convert\_y(y);

end

if obj.p.plot\_params.image\_on

if ~obj.p.plot\_params.image\_processed

im = obj.core.il.getImage(obj.current\_frame);

imshow(im,'Parent',ax,'XData', XData,'YData', YData); hold(ax, 'on');

else

im = obj.core.ipp.getImage(obj.current\_frame);

imagesc(XData,YData,im,'Parent',ax); hold(ax, 'on');

colormap(ax, obj.p.plot\_params.image\_colormap);

end

end

obj.piv\_plot.ax = ax;

obj.piv\_plot.plot\_params = obj.p.plot\_params;

if isnan(obj.current\_ti)

text(ax, 0, 0,'Кадр не выбран для вычислений','color','r','VerticalAlignment','bottom','units','normalized');

else

if obj.p.plot\_params.grid\_on

gp = obj.p.pg.do\_PIV\_mat;

obj.piv\_plot.plot\_centers(x,y,gp); hold(ax, 'on');

end

if obj.p.plot\_params.grid\_show\_windows

wSize = obj.p.pp.wSize;

if obj.p.plot\_params.plot\_in\_scale

wSize = wSize \* obj.p.rp.scale \* 1e-6;

end

obj.piv\_plot.plot\_windows(x,y,wSize); hold(ax, 'on'); %TODO какой из wSize? TODO работает только для простого масштаба

end

piv\_output = obj.core.piv\_processor.data.Get(obj.current\_ti);

fi\_output = obj.core.fi\_processor. data.Get(obj.current\_ti);

if isempty(piv\_output)

text(ax, 0,0,'Нет данных PIV', 'color','r','VerticalAlignment','bottom','units','normalized');

else

u\_piv = piv\_output.xdispl;

v\_piv = piv\_output.ydispl;

gp\_piv = piv\_output.status == 0;

u\_piv(~gp\_piv) = nan;

v\_piv(~gp\_piv) = nan;

if obj.p.plot\_params.plot\_in\_scale

u\_piv = obj.core.transform\_processor.Convert\_xdispl(u\_piv);

v\_piv = obj.core.transform\_processor.Convert\_ydispl(v\_piv);

end

if obj.p.plot\_params.grid\_show\_status

obj.piv\_plot.plot\_status( x, y, piv\_output.status, 'right'); hold(ax, 'on');

end

end

if isempty(fi\_output)

text(ax, 0,0,'Нет данных фильтрации','color','r','VerticalAlignment','bottom','units','normalized');

else

u\_fi = fi\_output.xdispl;

v\_fi = fi\_output.ydispl;

gp\_fi = ~isnan(u\_fi) & ~isnan(v\_fi);

if obj.p.plot\_params.plot\_in\_scale

u\_fi = obj.core.transform\_processor.Convert\_xdispl(u\_fi);

v\_fi = obj.core.transform\_processor.Convert\_ydispl(v\_fi);

end

if obj.p.plot\_params.grid\_show\_status\_filtering

obj.piv\_plot.plot\_status( x, y, fi\_output.status, 'left'); hold(ax, 'on');

end

end

if ~isempty( obj.core.analyzer.mean\_field\_xd ) && ~isempty( obj.core.analyzer.mean\_field\_yd )

u\_mean = obj.core.analyzer.mean\_field\_xd;

v\_mean = obj.core.analyzer.mean\_field\_yd;

gp\_mean = ~isnan(u\_mean) & ~isnan(v\_mean);

if obj.p.plot\_params.plot\_in\_scale

u\_mean = obj.core.transform\_processor.Convert\_xdispl(u\_mean);

v\_mean = obj.core.transform\_processor.Convert\_ydispl(v\_mean);

end

end

if obj.p.plot\_params.pcolor\_on

eqvidistant = ~isnan(obj.p.pg.xStep) && ~isnan(obj.p.pg.yStep);

if strcmp( obj.p.plot\_params.pcolor\_source, 'piv' ) && ~isempty(piv\_output)

u = u\_piv;

v = v\_piv;

gp = gp\_piv;

obj.piv\_plot.plot\_pcolor( x, y, u, v, gp, eqvidistant ); hold(ax, 'on');

end

if strcmp( obj.p.plot\_params.pcolor\_source, 'fi' ) && ~isempty(fi\_output)

u = u\_fi;

v = v\_fi;

gp = gp\_fi;

obj.piv\_plot.plot\_pcolor( x, y, u, v, gp, eqvidistant ); hold(ax, 'on');

end

if strcmp( obj.p.plot\_params.pcolor\_source, 'mean' ) &&...

~isempty( obj.core.analyzer.mean\_field\_xd ) && ~isempty( obj.core.analyzer.mean\_field\_yd )

u = u\_mean;

v = v\_mean;

gp = gp\_mean;

obj.piv\_plot.plot\_pcolor( x, y, u, v, gp, eqvidistant ); hold(ax, 'on');

end

end

if obj.p.plot\_params.quiver\_on\_ID

ID\_data = obj.core.piv\_processor.ID\_data.Get(ti);

if ~isempty(ID\_data)

u = ID\_data.xIDArr;

v = ID\_data.yIDArr;

if obj.p.plot\_params.plot\_in\_scale

u = obj.core.transform\_processor.Convert\_xdispl(u);

v = obj.core.transform\_processor.Convert\_ydispl(v);

end

gp = ~isnan(u);

obj.piv\_plot.plot\_quiver( x, y, u, v, gp, obj.p.plot\_params.quiver\_color\_ID ); hold(ax, 'on');

end

end

if obj.p.plot\_params.quiver\_on\_mean && ~isempty( obj.core.analyzer.mean\_field\_xd ) && ~isempty( obj.core.analyzer.mean\_field\_yd )

u = u\_mean;

v = v\_mean;

gp = gp\_mean;

obj.piv\_plot.plot\_quiver( x, y, u, v, gp, obj.p.plot\_params.quiver\_color\_mean ); hold(ax, 'on');

end

if obj.p.plot\_params.quiver\_on\_piv && ~isempty(piv\_output)

u = u\_piv;

v = v\_piv;

gp = gp\_piv;

obj.piv\_plot.plot\_quiver( x, y, u, v, gp, obj.p.plot\_params.quiver\_color\_piv ); hold(ax, 'on');

end

if obj.p.plot\_params.quiver\_on\_fi && ~isempty(fi\_output)

u = u\_fi;

v = v\_fi;

gp = gp\_fi;

obj.piv\_plot.plot\_quiver( x, y, u, v, gp, obj.p.plot\_params.quiver\_color\_fi ); hold(ax, 'on');

end

%выбранные точки

for selected\_points\_i = 1:size(obj.selected\_points,1)

xinit = obj.selected\_points(selected\_points\_i,1); % в px

yinit = obj.selected\_points(selected\_points\_i,2);

if obj.p.plot\_params.plot\_in\_scale

xinit = obj.core.transform\_processor.Convert\_x(xinit);

yinit = obj.core.transform\_processor.Convert\_y(yinit);

end

line(obj.ax\_fields, 'XData',xinit,'YData',yinit,...

'Marker','+','color','w','markersize',20);

text(obj.ax\_fields, xinit, yinit,...

num2str(selected\_points\_i),...

'color','w','VerticalAlignment','bottom','units','data','fontsize',14);

end

end

axis(ax, 'equal');

ax.YDir = 'Reverse';

ax.XLim = XData;

ax.YLim = YData;

if obj.p.plot\_params.show\_grid

grid(ax, 'on');

end

if obj.p.plot\_params.show\_ticks

if obj.p.plot\_params.plot\_in\_scale

xlabel(ax, 'x, м');

ylabel(ax, 'y, м');

else

xlabel(ax, 'x, px');

ylabel(ax, 'y, px');

end

else

ax.XTick = [];

ax.YTick = [];

set(ax,'LooseInset',get(ax,'TightInset'))

end

if obj.p.plot\_params.show\_title

title( ax, sprintf('Кадр %d, поле %d, время %.4f', frame, ti, time ) );

end

end

%%

function Redraw\_process(obj, processor\_name)

if isempty(obj.core), return; end

if isempty(obj.ax\_process), return; end

if ~ishandle(obj.ax\_process), return; end

if isempty(obj.core.piv\_processor), return; end

switch processor\_name

case 'piv\_processor'

processor = obj.core.piv\_processor;

case 'fi\_processor'

processor = obj.core.fi\_processor;

otherwise

return

end

ax = obj.ax\_process;

cla(ax);

reset(ax);

frame\_count = obj.p.pf.frame\_count;

plot( ax, processor.iteration\_time\_arr, '.-');

hold( ax, 'on');

plot( ax, processor.iteration\_time\_full\_arr, '.-r');

ti = processor.current\_ti;

ax.YLim = [0 ax.YLim(2)];

if ~isnan(ti)

ylims = ax.YLim;

plot( ax, [ti ti], ylims);

end

ti\_arr = 1 : frame\_count;

ti\_arr\_pr = ax.YLim(2) \* processor.is\_processed\_arr;

plot( ax, ti\_arr, ti\_arr\_pr,'.g');

xlabel( ax, 'Номер поля');

ylabel( ax, 'Время на вычисление (с)');

ax.XLim = [1 frame\_count];

end

%%

function Plot\_hist\_ti(obj)

if isempty(obj.core), return; end

if isempty(obj.p.plot\_params), return; end

clf

ax = gca;

frame = obj.current\_frame;

ti = obj.current\_ti;

time = obj.core.transform\_processor.Convert\_t(frame);

if isnan(obj.current\_ti)

text(ax, 0, 0,'Кадр не выбран для вычислений','color','r','VerticalAlignment','bottom','units','normalized');

else

piv\_output = obj.core.piv\_processor.data.Get(obj.current\_ti);

ID = obj.core.piv\_processor.ID\_data.Get(obj.current\_ti);

fi\_output = obj.core.fi\_processor. data.Get(obj.current\_ti);

if isempty(piv\_output)

text(ax, 0,0,'Нет данных PIV', 'color','r','VerticalAlignment','bottom','units','normalized');

else

u = piv\_output.xdispl;

v = piv\_output.ydispl;

cc = piv\_output.CC\_maxRaitio;

gp = piv\_output.status == 0;

[u\_counts,u\_centers] = hist(u(gp));

[v\_counts,v\_centers] = hist(v(gp));

[cc\_counts,cc\_centers] = hist(cc(gp & cc < 2));

subplot(311);

plot(u\_centers, u\_counts); hold on;

plot(v\_centers, v\_counts);

subplot(313);

plot(cc\_centers, cc\_counts); hold on;

end

if isempty(fi\_output)

text(ax, 0,0,'Нет данных фильтрации','color','r','VerticalAlignment','bottom','units','normalized');

else

u = fi\_output.xdispl;

v = fi\_output.ydispl;

gp = ~isnan(u) & ~isnan(v);

[u\_counts,u\_centers] = hist(u(gp));

[v\_counts,v\_centers] = hist(v(gp));

subplot(312);

plot(u\_centers, u\_counts); hold on;

plot(v\_centers, v\_counts);

end

if isempty(ID)

text(ax, 0,0,'Нет данных предсмещений','color','r','VerticalAlignment','bottom','units','normalized');

else

u = piv\_output.xdispl;

v = piv\_output.ydispl;

gp = piv\_output.status == 0;

u\_id = ID.xIDArr;

v\_id = ID.yIDArr;

u = u - u\_id;

v = v - v\_id;

[u\_counts,u\_centers] = hist(u(gp));

[v\_counts,v\_centers] = hist(v(gp));

subplot(312);

plot(u\_centers, u\_counts); hold on;

plot(v\_centers, v\_counts);

end

xlabel(ax, 'x, м');

ylabel(ax, 'y, м');

end

end

%% Предобработка изображений

function Calc\_and\_use\_bg(obj, bg\_source\_auto\_N, bg\_source\_auto\_prctile) %вычисляет и устанавливает фон

if isempty(obj.core), return; end

if isempty(obj.core.il), return; end

if exist('bg\_source\_auto\_N','var')

obj.p.ipp.bg\_source\_auto\_N = bg\_source\_auto\_N;

end

if exist('bg\_source\_auto\_prctile','var')

obj.p.ipp.bg\_source\_auto\_prctile = bg\_source\_auto\_prctile;

end

disp('Вычисление фона...');

obj.p.ipp.bg = CalcBgMedian\_v2019\_10(obj.core.il, obj.p.ipp.bg\_source\_auto\_N, obj.p.ipp.bg\_source\_auto\_prctile, obj.p.ipp.color\_channel);

disp('вычисление фона завершено');

obj.p.ipp.bg\_on = true;

obj.Set\_ipp\_params( obj.p.ipp );

end

function Load\_and\_use\_bg(obj, file\_path)

if isempty(obj.core), return; end

if isempty(obj.core.il), return; end

if exist('file\_path','var')

obj.p.ipp.bg\_source\_file\_path = file\_path;

end

if ~exist(obj.p.ipp.bg\_source\_file\_path,'file')

obj.Add\_to\_log('Файл не найден');

return;

end

im = imread(obj.p.ipp.bg\_source\_file\_path);

if size(im,2) ~= obj.core.il.imSize(1) || size(im,1) ~= obj.core.il.imSize(2)

obj.Add\_to\_log('Размер изображения фона должен совпадать с размером изображений. Фон не вычитается');

return;

end

bg = im(:,:,obj.p.ipp.color\_channel);

p\_ipp = obj.p.ipp;

p\_ipp.bg = bg;

p\_ipp.bg\_on = true;

obj.Set\_ipp\_params( p\_ipp );

end

function Load\_and\_use\_mask(obj, file\_path, mask\_source\_color)

if isempty(obj.core), return; end

if isempty(obj.core.il), return; end

if ~exist('file\_path','var')

obj.Add\_to\_log('Файл не найден');

return;

end

im = imread(file\_path);

if size(im,2) ~= obj.im\_size(1) || size(im,1) ~= obj.im\_size(2)

obj.Add\_to\_log('Размер изображения маски должен совпадать с размером изображений. Маска не используется');

return;

end

obj.p.ipp.mask\_source\_path = file\_path;

if exist('mask\_source\_color','var')

obj.p.ipp.mask\_source\_color = mask\_source\_color;

end

mask = ...

im(:,:,1) == obj.p.ipp.mask\_source\_color(1) &...

im(:,:,2) == obj.p.ipp.mask\_source\_color(2) &...

im(:,:,3) == obj.p.ipp.mask\_source\_color(3);

obj.p.ipp.mask = mask;

obj.p.ipp.mask\_on = true;

obj.Set\_ipp\_params( obj.p.ipp );

end

%% Экспорт

function Export\_all\_fields\_to\_one\_file( obj, file\_path )

% Сохранение всех полей скорости в один файл

obj.Add\_to\_log(sprintf('Сохранение в %s', file\_path));

obj.core.exporter.Export\_all\_fields\_to\_one\_file(...

file\_path, obj.core.fi\_processor.data, obj.core.transform\_processor );

obj.Add\_to\_log(sprintf('Сохранение завершено'));

end

function Export\_all\_fields\_to\_separate\_files( obj, file\_path\_base )

% Сохранение всех полей скорости в набор файлов

obj.Add\_to\_log(sprintf('Сохранение в %s', file\_path\_base));

obj.core.exporter.Export\_all\_fields\_to\_separate\_files(...

file\_path\_base, obj.core.fi\_processor.data, obj.core.transform\_processor);

obj.Add\_to\_log(sprintf('Сохранение завершено'));

end

function Export\_time\_series( obj, file\_path\_base, pos\_ij\_arr )

% Сохранение временных зависимостей в набор файлов

obj.Add\_to\_log(sprintf('Сохранение в %s', file\_path\_base));

[ts, pos\_ij\_arr] = obj.Get\_time\_series( pos\_ij\_arr );

obj.core.exporter.Export\_time\_series\_to\_separate\_files( file\_path\_base, ts, obj.core.transform\_processor, pos\_ij\_arr );

obj.Add\_to\_log(sprintf('Сохранение завершено'));

end

function Export\_mean\_field\_to\_files( obj, file\_path, profile\_y\_\_pos\_i, profile\_x\_\_pos\_j )

% Сохранение среднего поля в файл

% profile\_x\_\_pos\_j - набор номеров координат по y для профилей вдоль x

% profile\_x\_\_pos\_j - набор номеров координат по x для профилей вдоль y

obj.Add\_to\_log(sprintf('Сохранение в %s', file\_path));

[xd, yd] = obj.Update\_mean\_field();

x = obj.p.pg.xMat;

y = obj.p.pg.yMat;

x = obj.core.transform\_processor.Convert\_x(x);

y = obj.core.transform\_processor.Convert\_y(y);

xd = obj.core.transform\_processor.Convert\_xdispl(xd);

yd = obj.core.transform\_processor.Convert\_ydispl(yd);

file\_path\_mean = strrep( file\_path, '~info~', ' среднее поле' );

obj.core.exporter.Export\_one\_field\_to\_file( file\_path\_mean, x, y, xd, yd)

for profile\_x\_\_pos\_j\_\_i = 1:numel(profile\_x\_\_pos\_j)

pos\_j = profile\_y\_\_pos\_i(profile\_x\_\_pos\_j\_\_i);

x\_profile = obj.p.pg.xMat(:,pos\_j);

xd\_profile = xd(:,pos\_j);

yd\_profile = yd(:,pos\_j);

y = obj.core.transform\_processor.Convert\_y(y);

x\_profile = obj.core.transform\_processor.Convert\_x(x\_profile);

file\_path\_profile = strrep( file\_path, '~info~', sprintf(' профиль по x, pos\_j %d, y %.5f', pos\_j, y) );

obj.core.exporter.Export\_profile\_to\_file( file\_path\_profile, x\_profile, xd\_profile, yd\_profile );

end

for profile\_y\_\_pos\_i\_\_i = 1:numel(profile\_y\_\_pos\_i)

pos\_i = profile\_y\_\_pos\_i(profile\_y\_\_pos\_i\_\_i);

x = unique(obj.p.pg.xMat(pos\_i,:));

y\_profile = obj.p.pg.yMat(pos\_i,:);

xd\_profile = xd(pos\_i,:);

yd\_profile = yd(pos\_i,:);

x = obj.core.transform\_processor.Convert\_x(x);

y\_profile = obj.core.transform\_processor.Convert\_y(y\_profile);

file\_path\_profile = strrep( file\_path, '~info~', sprintf(' профиль по y, pos\_i %d, x %.5f', pos\_i, x) );

obj.core.exporter.Export\_profile\_to\_file( file\_path\_profile, y\_profile, xd\_profile, yd\_profile );

end

obj.Add\_to\_log(sprintf('Сохранение завершено'));

end

function Export\_figure\_current(obj, file\_path)

if ~exist(fileparts(file\_path),'dir'), mkdir(fileparts(file\_path)); end

obj.Redraw\_fields();

saveas(obj.fig\_fields, file\_path);

end

function Export\_figure\_all(obj, file\_path\_base, ti\_arr)

% Сохранение изображений с полями скорости в набор файлов

if ~exist(fileparts(file\_path\_base),'dir'), mkdir(fileparts(file\_path\_base)); end

if ~exist('ti\_arr','var'), ti\_arr = 1:obj.ti\_count; end

for i = 1:numel(ti\_arr)

ti = ti\_arr(i);

obj.Go\_to\_ti(ti);

obj.Redraw\_fields();

drawnow();

file\_path = strrep( file\_path\_base, '~info~', sprintf(' %08d',ti) );

saveas( obj.fig\_fields, file\_path );

obj.Add\_to\_log(sprintf('Сохранен кадр %d в %s', ti, file\_path));

end

end

function Export\_figure\_all\_video(obj, file\_path, frame\_rate, ti\_arr)

% Сохранение изображений с полями скорости в видеофайл

if ~exist(fileparts(file\_path),'dir'), mkdir(fileparts(file\_path)); end

v = VideoWriter(file\_path);

v.FrameRate = frame\_rate;

open(v);

if ~exist('ti\_arr','var'), ti\_arr = 1:obj.ti\_count; end

for i = 1:numel(ti\_arr)

ti = ti\_arr(i);

obj.Go\_to\_ti(ti);

obj.Redraw\_fields();

drawnow();

writeVideo(v, getframe(obj.fig\_fields));

obj.Add\_to\_log(sprintf('Сохранен кадр %d',ti));

end

close(v)

end

function Export\_comment(obj, file\_path)

if ~exist(fileparts(file\_path),'dir'), mkdir(fileparts(file\_path)); end

fileID = fopen(file\_path,'w');

for i = 1:numel(obj.p.rp.comment)

fprintf(fileID, '%s\r\n', obj.p.rp.comment{i});

end

fclose(fileID);

end

%% Анализ

function pos\_ij\_arr = Get\_pos\_ij\_arr\_\_of\_selected\_point(obj)

if isempty(obj.selected\_points)

pos\_ij\_arr = [];

return;

end

pos\_ij\_arr = zeros( size(obj.selected\_points,1), 2 );

for i = 1:size(obj.selected\_points,1)

sp\_x = obj.selected\_points(i,1);

sp\_y = obj.selected\_points(i,2);

d = sqrt( (obj.p.pg.xMat - sp\_x).^2 + (obj.p.pg.yMat - sp\_y).^2 );

[d\_min, d\_min\_i] = min(d(:));

if d\_min > 100, warning('Выбранная точка дальше 100 px от ближайшего узла'); end

[pos\_i, pos\_j] = ind2sub(size(d),d\_min\_i);

pos\_ij\_arr(i,:) = [pos\_i, pos\_j];

end

end

function pos = Get\_pos\_by\_pos\_ij\_arr(obj, pos\_ij\_arr)

if isempty(pos\_ij\_arr)

pos = [];

return;

end

pos = nan(size(pos\_ij\_arr,1),4);

for i = 1:size(pos\_ij\_arr,1)

pos\_i = pos\_ij\_arr(i,1);

pos\_j = pos\_ij\_arr(i,2);

x\_px = obj.p.pg.xMat(pos\_i,pos\_j);

y\_px = obj.p.pg.yMat(pos\_i,pos\_j);

x\_m = obj.core.transform\_processor.Convert\_x(x\_px);

y\_m = obj.core.transform\_processor.Convert\_y(y\_px);

pos(i,:) = [x\_px y\_px x\_m y\_m];

end

end

function [ts, pos\_ij\_arr] = Get\_time\_series( obj, pos\_ij\_arr )

[ts, pos\_ij\_arr] = obj.core.analyzer.Get\_time\_series( obj.core.fi\_processor.data, pos\_ij\_arr);

end

function Plot\_time\_series( obj, pos\_ij\_arr )

[ts, pos\_ij\_arr] = obj.Get\_time\_series( pos\_ij\_arr );

for pos\_ij\_arr\_i = 1:size(ts,3)

pos\_i = pos\_ij\_arr(pos\_ij\_arr\_i,1);

pos\_j = pos\_ij\_arr(pos\_ij\_arr\_i,2);

x = obj.p.pg.xMat(pos\_i,pos\_j);

y = obj.p.pg.yMat(pos\_i,pos\_j);

t = ts(:, 2, pos\_ij\_arr\_i);

xd = ts(:, 3, pos\_ij\_arr\_i);

yd = ts(:, 4, pos\_ij\_arr\_i);

subplot(211);

plot( t, xd, '.',...

'displayname',...

sprintf('pos\_i %d, pos\_j %d, x %.5f, y %.5f', pos\_i, pos\_j, x, y) );

hold('on');

xlabel('Время, с');

ylabel('u, м/с');

subplot(212);

plot( t, yd, '.',...

'displayname',...

sprintf('pos\_i %d, pos\_j %d, x %.5f, y %.5f', pos\_i, pos\_j, x, y) );

hold('on');

xlabel('Время, с');

ylabel('v, м/с');

end

end

function Plot\_profiles( obj, profile\_y\_\_pos\_i, profile\_x\_\_pos\_j, legend\_text )

% profile\_x\_\_pos\_j - набор номеров координат по y для профилей вдоль x

% profile\_x\_\_pos\_j - набор номеров координат по x для профилей вдоль y

%TODO сделать Getprofiles, сделать для не среднего

[xd, yd] = obj.Update\_mean\_field();

xd = obj.core.transform\_processor.Convert\_xdispl(xd);

yd = obj.core.transform\_processor.Convert\_ydispl(yd);

if ~isempty(profile\_x\_\_pos\_j)

for profile\_x\_\_pos\_j\_\_i = 1:numel(profile\_x\_\_pos\_j)

pos\_j = profile\_y\_\_pos\_i(profile\_x\_\_pos\_j\_\_i);

y = unique(obj.p.pg.yMat(:,pos\_j));

x\_profile = obj.p.pg.xMat(:,pos\_j);

xd\_profile = xd(:,pos\_j);

yd\_profile = yd(:,pos\_j);

y = obj.core.transform\_processor.Convert\_y(y);

x\_profile = obj.core.transform\_processor.Convert\_x(x\_profile);

subplot(221);

plot( x\_profile, xd\_profile, '.-',...

'displayname',...

sprintf('%s pos\_j %d, y %.5f м', legend\_text, pos\_j, y) );hold('on');

subplot(222);

plot( x\_profile, yd\_profile, '.-',...

'displayname',...

sprintf('%s pos\_j %d, y %.5f м', legend\_text, pos\_j, y) );hold('on');

end

subplot(221);

xlabel('x, м');

ylabel('u, м/с');

legend('show','location','eastoutside','interpreter','none')

subplot(222);

xlabel('x, м');

ylabel('v, м/с');

legend('show','location','eastoutside','interpreter','none')

end

if ~isempty(profile\_y\_\_pos\_i)

for profile\_y\_\_pos\_i\_\_i = 1:numel(profile\_y\_\_pos\_i)

pos\_i = profile\_y\_\_pos\_i(profile\_y\_\_pos\_i\_\_i);

x = unique(obj.p.pg.xMat(pos\_i,:));

y\_profile = obj.p.pg.yMat(pos\_i,:);

xd\_profile = xd(pos\_i,:);

yd\_profile = yd(pos\_i,:);

x = obj.core.transform\_processor.Convert\_x(x);

y\_profile = obj.core.transform\_processor.Convert\_y(y\_profile);

subplot(223);

plot( xd\_profile, y\_profile, '.-',...

'displayname',...

sprintf('%s pos\_i %d, x %.5f м', legend\_text, pos\_i, x) );hold('on');

subplot(224);

plot( yd\_profile, y\_profile, '.-',...

'displayname',...

sprintf('%s pos\_i %d, x %.5f м', legend\_text, pos\_i, x) );hold('on');

end

subplot(223);

ylabel('y, м');

xlabel('u, м/с');

legend('show','location','eastoutside','interpreter','none')

subplot(224);

ylabel('y, м');

xlabel('v, м/с');

legend('show','location','eastoutside','interpreter','none')

end

end

function [xd, yd] = Update\_mean\_field( obj )

obj.Add\_to\_log('Вычисление среднего поля...')

[xd, yd] = obj.core.analyzer.Calc\_mean\_field();

obj.Add\_to\_log('Среднее поле вычислено')

end

function [xd\_mat3, yd\_mat3] = Get\_displ\_mat3( obj, stage)

switch stage

case 'piv'

data = obj.core.piv\_processor.data;

case 'fi'

data = obj.core.fi\_processor.data;

end

[data\_arr, data\_is\_empty\_arr] = data.Get\_all\_as\_cell();

xd\_mat3 = cellfun(@(x) x.xdispl, data\_arr(~data\_is\_empty\_arr),'uni',false);

yd\_mat3 = cellfun(@(x) x.ydispl, data\_arr(~data\_is\_empty\_arr),'uni',false);

xd\_mat3 = cat(3,xd\_mat3{:});

yd\_mat3 = cat(3,yd\_mat3{:});

end

%%

function Add\_selected\_point\_px(obj,x\_px,y\_px)

x\_m = obj.core.transform\_processor.Convert\_x(x\_px);

y\_m = obj.core.transform\_processor.Convert\_y(y\_px);

obj.selected\_points = [ obj.selected\_points; x\_px, y\_px, x\_m, y\_m ];

end

function Add\_selected\_point\_m(obj,x\_m,y\_m)

x\_px = obj.core.transform\_processor.Convert\_x\_to\_px(x\_m);

y\_px = obj.core.transform\_processor.Convert\_y\_to\_px(y\_m);

obj.selected\_points = [ obj.selected\_points; x\_px, y\_px, x\_m, y\_m ];

end

function Clear\_selected\_point\_m(obj)

obj.selected\_points = [ ];

end

%%

function [fi\_output, pg] = Get\_field\_intrept(obj, ti)

if isempty(obj.core) || isempty(obj.core.fi\_processor), return; end

fi\_output = obj.core.fi\_processor.data.Get(ti);

pg = obj.p.pg;

end

%% Предсмещения

function Import\_initial\_displacements(obj, input\_piv\_main)

% загружает начальные смещения из проекта

% input\_piv\_main - типа PIV\_main из которого берутся смещения

if ~isequal( obj.p.pf.first\_frames, input\_piv\_main.p.pf.first\_frames )

error('pf совпадают');

end

obj.Add\_to\_log('Начало импорта смещений...');

for ti = 1:obj.ti\_count

[ fi\_input, pg\_input ] = input\_piv\_main.Get\_field\_intrept(ti);

if isempty(fi\_input)

warning('Нет данных предсмещения');

continue; % ID останется нулевым

else

xdispl = fi\_input.xdispl;

ydispl = fi\_input.ydispl;

x = pg\_input.xMat;

y = pg\_input.yMat;

gp = ~isnan(xdispl) & ~isnan(ydispl);

if ~any(gp(:)), continue; end % ID останется нулевым

SI\_xdispl = scatteredInterpolant( x(gp), y(gp), xdispl(gp));

SI\_ydispl = scatteredInterpolant( x(gp), y(gp), ydispl(gp));

SI\_xdispl.Method = 'linear';

SI\_ydispl.Method = 'linear';

SI\_xdispl.ExtrapolationMethod = 'nearest';

SI\_ydispl.ExtrapolationMethod = 'nearest';

xdispl\_q = SI\_xdispl( obj.p.pg.xMat, obj.p.pg.yMat );

ydispl\_q = SI\_ydispl( obj.p.pg.xMat, obj.p.pg.yMat );

if isempty(xdispl\_q) || isempty(ydispl\_q)

warning('Process\_one\_\_interpolate\_space: Интерполяция не удалась')

continue;

end

frame\_step\_ratio = obj.p.pf.frame\_step / input\_piv\_main.p.pf.frame\_step; % на случай если считалось с разными шагами

ID.xIDArr = xdispl\_q \* frame\_step\_ratio;

ID.yIDArr = ydispl\_q \* frame\_step\_ratio;

obj.core.piv\_processor.ID\_data.Set( ti, ID );

end

end

obj.Add\_to\_log('Импорт смещений завершен');

end

%% Импорт параметров из проекта

function Import\_params(obj, input\_piv\_main, params\_list)

% params\_list - список имен параметров как cell массив строк {'pp', 'pg'}

if any(strcmp(params\_list,'rp'))

obj.Set\_rp\_params(input\_piv\_main.p.rp);

end

if any(strcmp(params\_list,'il'))

obj.Set\_il\_params(input\_piv\_main.p.il);

end

if any(strcmp(params\_list,'ipp'))

obj.Set\_ipp\_params(input\_piv\_main.p.ipp);

end

if any(strcmp(params\_list,'pp'))

obj.Set\_pp(input\_piv\_main.p.pp);

end

if any(strcmp(params\_list,'pf'))

obj.Set\_pf(input\_piv\_main.p.pf);

end

if any(strcmp(params\_list,'pg'))

obj.Set\_pg(input\_piv\_main.p.pg);

end

if any(strcmp(params\_list,'fi'))

obj.Set\_fi\_params(input\_piv\_main.p.fi);

end

if any(strcmp(params\_list,'ap'))

obj.Set\_ap(input\_piv\_main.p.ap);

end

if any(strcmp(params\_list,'plot\_params'))

obj.Set\_plot\_params(input\_piv\_main.p.plot\_params);

end

end

end

end

## PIV\_params.m

classdef PIV\_params

%UNTITLED Summary of this class goes here

% pp.wSize = [32 32];

% pp.maxDispl = pp.wSize\*Inf;

% pp.minS2NRatio = 1;

% pp.useSubpixel = true;

% pp.useBiasCorrection = false;

% pp.saveCCandIMC = false;

% pp.xIDArr%- массив(можно двумерный) координат центров

% pp.yIDArr%- массив(можно двумерный) начальных смещений

% pp.doCCArr%- массив делать/не делать КК (может быть [] - делать все)

properties

version = 1;

wSize = [64 64];

doFirstPass = true;

wSize1 = [128 128];

maxDispl = [Inf Inf];

minS2NRatio = 1;

useSubpixel = true;

doMultiMax = true;

xIDArr = 0; % массив(можно двумерный) начальных смещений

yIDArr = 0;

saveCCandIMC = false; % сохранять ккф и элементы изображений для анализа

max\_mask\_pixels\_prc = 50; %максимальный процент пикселей маски в окне. При превышении ККФ не вычисляется

useBiasCorrection = false; % использовать Bias сorrection

end

end

## PIV\_plot.m

classdef PIV\_plot < handle

%PIV\_PLOT Summary of this class goes here

% Detailed explanation goes here

properties

ax %оси на которых всё рисуется

plot\_params Plot\_params

end

methods

function plot\_centers( obj, x, y, gp )

plot(obj.ax, x(gp), y(gp),...

obj.plot\_params.grid\_symbol,...

'markersize',obj.plot\_params.grid\_symbol\_size,...

'color',obj.plot\_params.grid\_symbol\_color);

end

%%

function plot\_status( obj, x, y, s, horizontalAlignment)

for i = 1:numel(s)

text(obj.ax, x(i), y(i), num2str(s(i)), ...

'fontsize', 6, ...

'HorizontalAlignment', horizontalAlignment,...

'VerticalAlignment', 'baseline','color','r');

end

end

%%

function plot\_windows(obj, x, y, wSize )

if numel(x) >= 2

i = 2;

rectangle(obj.ax,'Position',[x(i)-wSize(1)/2,y(i)-wSize(2)/2,wSize(1),wSize(2)],'edgecolor', 'g');

end

if numel(x) >= 1

i = 1;

rectangle(obj.ax,'Position',[x(i)-wSize(1)/2,y(i)-wSize(2)/2,wSize(1),wSize(2)],'edgecolor', 'r');

end

end

%%

function plot\_pcolor(obj, x, y, u, v, gp, eqvidistant )

switch obj.plot\_params.pcolor\_value

case 'u'

value = u;

case 'v'

value = v;

case 'V'

value = sqrt(u.^2 + v.^2);

end

% value\_q = griddata(x(gp),y(gp),value(gp),x,y);

value\_q = value; value(~gp) = nan;

% в случае эквидистантной сетки:

if eqvidistant

x\_limits = [ x(1) x(end) ]; %вроде с матрицами должно правильно работать

y\_limits = [ y(1) y(end) ];

b = imagesc( obj.ax, ...

x\_limits, y\_limits, value\_q);

set(b,'AlphaData',~isnan(value\_q));

else

pcolor(obj.ax,x,y,value\_q); shading(obj.ax, obj.plot\_params.pcolor\_shading);

end

colormap(obj.ax,obj.plot\_params.pcolor\_colormap);

if obj.plot\_params.pcolor\_caxis\_auto

current\_caxis = prctile(value(:),[5 95]);

else

current\_caxis = obj.plot\_params.pcolor\_caxis;

end

caxis(obj.ax,current\_caxis);

if (obj.plot\_params.pcolor\_colorbar\_on)

colorbar(obj.ax);

else

colorbar(obj.ax,'off');

end

end

%%

function plot\_quiver(obj, x, y, u, v, gp, color )

stepped\_selected = false(size(x)); %точки, выбранные после прореживания

stepped\_selected(...

1 : obj.plot\_params.quiver\_step : end,...

1 : obj.plot\_params.quiver\_step : end) = true;

gp(~stepped\_selected) = false;

if any(gp(:))

if obj.plot\_params.quiver\_directions\_only

V = sqrt(u.^2 + v.^2);

u = u./V;

v = v./V;

end

quiver(obj.ax,x(gp),y(gp),...

obj.plot\_params.quiver\_scale \* u(gp),...

obj.plot\_params.quiver\_scale \* v(gp),...

0,...

'Color', color)

end

end

%%

function plot\_hist(obj, PIV\_output, u\_or\_v )

xd\_range = [-1 1] \* PIV\_output.pp.wSize(1)/2;

yd\_range = [-1 1] \* PIV\_output.pp.wSize(2)/2;

if u\_or\_v == 'u'

hist(obj.ax, PIV\_output.xdispl(gp),100)

set(gca,'xlim',xd\_range);

xlabel('u, px/frame')

end

if u\_or\_v == 'v'

hist(obj.ax, PIV\_output.ydispl(gp),100)

set(gca,'xlim',yd\_range);

xlabel('v, px/frame')

end

end

% function plot\_profile(obj, PIV\_output )

% % %

% % subplot(122);

% % % ПРОФИЛИ!

% % scale = 96.2;

% % fps = 2000;

% % p = PIV\_output\_arr{ti,1};

% % pp = p.pp;

% % piv\_grid = pp.piv\_grid;

% % wSize = pp.wSize;

% % y = piv\_grid.yMat(1,:)' - surface\_pos;

% % y = y \* scale / 1000;

% % gp = p.status == 0;

% % xd = p.xdispl;

% % xd(~gp) = nan;

% % xd\_profile = nanmean(xd,1)';

% % % plot(xd\_profile,y);

% % semilogx(y,xd);

% % view([90 -90])

% % set(gca,'ylim',[-10 0]);

% %

% % xlabel('Высота, мм')

% % ylabel('Скорость, см/с')

% % drawnow()

% % % saveas(gcf,[outputFolder '\' expName '\' num2str(ti,'%05d') '.jpg']);

% %

% end

end

end

## PIV\_processor.m

classdef PIV\_processor < handle

%Record\_processor версия 5

% Detailed explanation goes here

properties (Access = private)

% можно задать только при создании

pp PIV\_params %структуры pp для всей записи %TODO поддержку разных pp

pf PIV\_frames

pg PIV\_grid

end

properties (SetAccess = protected, GetAccess = public)

%% Вычисляемые параметры

iteration\_time\_arr % время на итерацию (только PIV)

iteration\_time\_full\_arr % полное время на итерацию (включая время между циклами)

compared\_frames\_\_num\_of\_pairs

is\_processed\_arr

%% временные, работают только для текущей пары

current\_firstFrame\_i %текущий номер первого кадра

current\_compared\_frames %номера текущих обрабатываемых кадров как в compared\_frames\_offset

current\_im\_arr % такого же размера как compared\_frames\_offset, и соответствующие кадры

current\_ti % текущий номер пары

%%

data Array\_temp\_storage; %результаты PIV

ID\_data Array\_temp\_storage; %предсмещения

end

properties

iteration\_time\_full\_tic % для полного времени

end

methods

function obj = PIV\_processor(pp, pf, pg,...

data\_storage\_file\_path\_base, data\_seg\_size)

obj.pp = pp;

obj.pf = pf;

obj.pg = pg;

obj.compared\_frames\_\_num\_of\_pairs = size(obj.pf.compared\_frames\_offset,1);

obj.current\_firstFrame\_i = nan;

obj.current\_compared\_frames = [];

obj.current\_im\_arr = [];

obj.current\_ti = nan;

obj.data = Array\_temp\_storage( ...

obj.pf.frame\_count,...

data\_seg\_size,...

data\_storage\_file\_path\_base);

obj.ID\_data = Array\_temp\_storage( ...

obj.pf.frame\_count,...

data\_seg\_size,...

[ data\_storage\_file\_path\_base ' ID\_data' ]);

obj.Reset\_processing();

end

function Reset\_processing(obj)

obj.iteration\_time\_arr = nan(obj.pf.frame\_count,1);

obj.iteration\_time\_full\_arr = nan(obj.pf.frame\_count,1);

obj.iteration\_time\_full\_tic = [];

obj.data.Clear();

obj.is\_processed\_arr = false(obj.pf.frame\_count,1);

end

function Process\_one(obj, ipp, ti)

piv\_iteration\_time = tic;

if (ti < 1 || ti > obj.pf.frame\_count)

error('Нужен индекс в firstFrames');

end

obj.current\_ti = ti;

obj.current\_firstFrame\_i = obj.pf.first\_frames(ti);

obj.current\_compared\_frames = obj.current\_firstFrame\_i + obj.pf.compared\_frames\_offset;

obj.current\_im\_arr = cell(size(obj.current\_compared\_frames));

for im\_arr\_i = 1:numel(obj.current\_im\_arr)

im = ipp.getImage(obj.current\_compared\_frames(im\_arr\_i));

if isempty(im)

return;

end

obj.current\_im\_arr{im\_arr\_i} = im;

end

% for pair\_i = 1%:obj.compared\_frames\_\_num\_of\_pairs TODO для нескольких кадров

pp\_current = obj.pp;

ID = obj.ID\_data.Get(ti);

if ~isempty(ID)

pp\_current.xIDArr = ID.xIDArr;

pp\_current.yIDArr = ID.yIDArr;

end

pg\_current = obj.pg;

piv\_output = DoCC10(...

obj.current\_im\_arr{1},...

obj.current\_im\_arr{2},...

pp\_current,...

pg\_current);

obj.data.Set(ti, piv\_output);

% end

obj.is\_processed\_arr(ti) = true;

obj.iteration\_time\_arr(ti) = toc(piv\_iteration\_time);

if ~isempty(obj.iteration\_time\_full\_tic)

obj.iteration\_time\_full\_arr(ti) = toc(obj.iteration\_time\_full\_tic);

end

obj.iteration\_time\_full\_tic = tic;

end

function info = Get\_info(obj)

ti = obj.current\_ti ;

if ~isnan(ti)

info = sprintf('осталось %.2f мин., поле %d за, %.4f c (из них PIV %.1f%%)', ...

sum(isnan(obj.iteration\_time\_full\_arr))\*nanmedian(obj.iteration\_time\_full\_arr)/60,...

obj.current\_ti,...

obj.iteration\_time\_full\_arr(ti),...

obj.iteration\_time\_arr(ti) / obj.iteration\_time\_full\_arr(ti) \* 100) ;

else

info = '';

end

end

function info = Get\_info\_progress(obj)

info = sprintf('Обработано %d из %d',...

sum(obj.is\_processed\_arr),...

obj.pf.frame\_count);

end

function Save\_data(obj)

obj.data.Move\_all\_to\_hdd();

end

end

end

## Plot\_params.m

classdef Plot\_params

%PLOT\_PARAMS Summary of this class goes here

% Detailed explanation goes here

properties

image\_on = true;

image\_processed = true;

image\_colormap = 'jet';

grid\_on = true;

grid\_symbol = '+';

grid\_symbol\_size = 5;

grid\_symbol\_color = 'r';

grid\_show\_status = false;

grid\_show\_status\_filtering = false;

grid\_show\_windows = false;

pcolor\_on = false;

pcolor\_source = 'fi'; % врианты 'piv' 'fi' 'mean'

pcolor\_colormap = 'jet';

pcolor\_value = 'V'; %u, v, V

pcolor\_caxis\_auto = true;

pcolor\_caxis = [0 100];

pcolor\_colorbar\_on = true;

pcolor\_shading = 'interp';

quiver\_on\_piv = true; %исходные

quiver\_on\_fi = true; %фильтрованные и интерполированные

quiver\_color\_piv = 'w';

quiver\_color\_fi = 'g';

quiver\_scale = 5;

quiver\_step = 1;

quiver\_directions\_only = false;

quiver\_on\_mean = false;

quiver\_color\_mean = 'y';

quiver\_on\_ID = false; % начальные смещения

quiver\_color\_ID = 'm';

plot\_in\_scale = true;

show\_grid = false;

show\_ticks = false;

show\_title = true;

end

end

## Project\_params.m

classdef Project\_params

%UNTITLED5 Summary of this class goes here

% Detailed explanation goes here

properties

rp Record\_params

il Image\_loader\_params

ipp Image\_preprocessor\_params

pp PIV\_params

pg PIV\_grid

pf PIV\_frames

fi Filter\_and\_interpolation\_params

ap Analyzer\_params

plot\_params Plot\_params

end

end

## Record\_params.m

classdef Record\_params

%UNTITLED3 Summary of this class goes here

% Detailed explanation goes here

properties

version = 1;

comment = {'Описание не заполнено'};

fps = 1;%к/с

scale = 1;%мкм/px

zero\_pos\_px = [0 0];

end

end

## Transform\_processor.m

classdef Transform\_processor < handle

%UNTITLED Summary of this class goes here

% Detailed explanation goes here

% все конвертирует в секунды, метры и м/с

properties

rp Record\_params

pg PIV\_grid

pf PIV\_frames

xMat % преобразованные координаты

yMat

time % преобразованное время

end

methods

function obj = Transform\_processor(rp, pg, pf )

obj.rp = rp;

obj.pg = pg;

obj.pf = pf;

obj.xMat = obj.Convert\_x( obj.pg.xMat );

obj.yMat = obj.Convert\_y( obj.pg.yMat );

obj.time = obj.Convert\_t( obj.pf.first\_frames );

end

%%

% В эти функции можно передавать матрицы любого размера

function u\_mps = Convert\_xdispl(obj, xdispl\_px)

u\_mps = xdispl\_px \* obj.rp.fps \* obj.rp.scale \* 1e-6;

end

function v\_mps = Convert\_ydispl(obj, ydispl\_px)

v\_mps = ydispl\_px \* obj.rp.fps \* obj.rp.scale \* 1e-6;

end

function x\_m = Convert\_x(obj, x\_px)

x\_m = ( x\_px - obj.rp.zero\_pos\_px(1) ) \* obj.rp.scale \* 1e-6;

end

function y\_m = Convert\_y(obj, y\_px)

y\_m = ( y\_px - obj.rp.zero\_pos\_px(2) ) \* obj.rp.scale \* 1e-6;

end

function time\_sec = Convert\_t(obj, frame\_number)

time\_sec = (frame\_number - 1) / obj.rp.fps;

end

function x\_px = Convert\_x\_to\_px(obj, x\_m)

x\_px = x\_m / obj.rp.scale \* 1e6 + obj.rp.zero\_pos\_px(1);

end

function y\_px = Convert\_y\_to\_px(obj, y\_m)

y\_px = y\_m / obj.rp.scale \* 1e6 + obj.rp.zero\_pos\_px(2);

end

end

end

# Графический интерфейс

## PIV\_db\_analyze.mlapp

classdef PIV\_db\_analyze < matlab.apps.AppBase

% Properties that correspond to app components

properties (Access = public)

UIFigure matlab.ui.Figure

Plot\_mean\_field matlab.ui.control.Button

Panel matlab.ui.container.Panel

Plot\_time\_series matlab.ui.control.Button

Label matlab.ui.control.Label

Plot\_time\_series\_pos\_ij\_arr matlab.ui.control.TextArea

end

properties (Access = private)

parent\_app

end

methods (Access = private)

% Code that executes after component creation

function startupFcn(app, parent\_app)

app.parent\_app = parent\_app;

end

% Button pushed function: Plot\_mean\_field

function Plot\_mean\_fieldPushed(app, event)

app.parent\_app.m.Update\_mean\_field();

app.parent\_app.m.Redraw\_fields();

end

% Button pushed function: Plot\_time\_series, Plot\_time\_series,

% Plot\_time\_series, Plot\_time\_series, Plot\_time\_series

function Plot\_time\_seriesButtonPushed(app, event)

pos\_ij\_arr = app.parent\_app.m.Get\_pos\_ij\_arr\_\_of\_selected\_point();

if isempty(pos\_ij\_arr)

app.Plot\_time\_series\_pos\_ij\_arr.Value = {'Точки не выбраны'};

return

end

pos = app.parent\_app.m.Get\_pos\_by\_pos\_ij\_arr( pos\_ij\_arr );

text = cell(size(pos\_ij\_arr,1),1);

for i = 1:size(pos\_ij\_arr,1)

pos\_i = pos\_ij\_arr(i,1);

pos\_j = pos\_ij\_arr(i,2);

x\_px = pos(i,1);

y\_px = pos(i,2);

x\_m = pos(i,3);

y\_m = pos(i,4);

text{i} = sprintf(...

'i %d, j %d, (%d, %d) px, (%.3f, %.3f) м',...

pos\_i, pos\_j,...

x\_px, y\_px,...

x\_m, y\_m);

end

app.Plot\_time\_series\_pos\_ij\_arr.Value = text;

figure();

app.parent\_app.m.Plot\_time\_series( pos\_ij\_arr );

end

end

% App initialization and construction

methods (Access = private)

% Create UIFigure and components

function createComponents(app)

% Create UIFigure

app.UIFigure = uifigure;

app.UIFigure.Position = [100 100 573 362];

app.UIFigure.Name = 'Анализ';

% Create Plot\_mean\_field

app.Plot\_mean\_field = uibutton(app.UIFigure, 'push');

app.Plot\_mean\_field.ButtonPushedFcn = createCallbackFcn(app, @Plot\_mean\_fieldPushed, true);

app.Plot\_mean\_field.Position = [20 327 226 22];

app.Plot\_mean\_field.Text = 'Вычислить среднее поле';

% Create Panel

app.Panel = uipanel(app.UIFigure);

app.Panel.Title = 'Временные реализаци';

app.Panel.Position = [20 77 540 176];

% Create Plot\_time\_series

app.Plot\_time\_series = uibutton(app.Panel, 'push');

app.Plot\_time\_series.ButtonPushedFcn = createCallbackFcn(app, @Plot\_time\_seriesButtonPushed, true);

app.Plot\_time\_series.Position = [6 127 339 22];

app.Plot\_time\_series.Text = 'Построить временные реализации для выбранных точек';

% Create Label

app.Label = uilabel(app.Panel);

app.Label.HorizontalAlignment = 'right';

app.Label.Position = [6 90 108 22];

app.Label.Text = {'Выбранные точки:'; ''};

% Create Plot\_time\_series\_pos\_ij\_arr

app.Plot\_time\_series\_pos\_ij\_arr = uitextarea(app.Panel);

app.Plot\_time\_series\_pos\_ij\_arr.Editable = 'off';

app.Plot\_time\_series\_pos\_ij\_arr.Position = [129 12 392 102];

end

end

methods (Access = public)

% Construct app

function app = PIV\_db\_analyze(varargin)

% Create and configure components

createComponents(app)

% Register the app with App Designer

registerApp(app, app.UIFigure)

% Execute the startup function

runStartupFcn(app, @(app)startupFcn(app, varargin{:}))

if nargout == 0

clear app

end

end

% Code that executes before app deletion

function delete(app)

% Delete UIFigure when app is deleted

delete(app.UIFigure)

end

end

end

## PIV\_db\_export.mlapp

classdef PIV\_db\_export < matlab.apps.AppBase

% Properties that correspond to app components

properties (Access = public)

UIFigure matlab.ui.Figure

fig\_im\_one matlab.ui.control.Button

fig\_im\_all matlab.ui.control.Button

fig\_vid\_all matlab.ui.control.Button

Label matlab.ui.control.Label

fig\_vid\_all\_fps matlab.ui.control.Spinner

comment matlab.ui.control.Button

Export\_all\_fields\_to\_one\_file matlab.ui.control.Button

Export\_all\_fields\_to\_separate\_files matlab.ui.control.Button

Export\_time\_series matlab.ui.control.Button

Export\_mean\_field\_to\_files matlab.ui.control.Button

Export\_mean\_field\_to\_files\_profiles matlab.ui.control.CheckBox

Label\_2 matlab.ui.control.Label

Export\_time\_series\_sp matlab.ui.control.TextArea

Export\_mean\_field\_to\_files\_profiles\_2 matlab.ui.control.CheckBox

end

properties (Access = private)

parent\_app % Description

end

methods (Access = private)

% Code that executes after component creation

function startupFcn(app, parent\_app)

app.parent\_app = parent\_app;

end

% Button pushed function: fig\_im\_one

function fig\_im\_onePushed(app, event)

defname = sprintf('output\\figure %s.png', datestr(now, 'yyyy-mm-dd HH-MM-SS'));

[file,path] = uiputfile( '\*.fig;\*.png;\*.jpg;\*.bmp','Выбор имени файла', defname );

if file == 0, return; end

selpath = fullfile(path,file);

app.parent\_app.m.Export\_figure\_current(selpath);

end

% Button pushed function: fig\_im\_all

function fig\_im\_allButtonPushed(app, event)

defname = sprintf('output\\figures %s.png', datestr(now, 'yyyy-mm-dd HH-MM-SS'));

[file,path] = uiputfile( '\*.fig;\*.png;\*.jpg;\*.bmp','Выбор имени файла', defname );

if file == 0, return; end

selpath = fullfile(path,file);

app.parent\_app.m.Export\_figure\_all(selpath);

end

% Button pushed function: fig\_vid\_all

function fig\_vid\_allButtonPushed(app, event)

defname = sprintf('output\\video %s.avi', datestr(now, 'yyyy-mm-dd HH-MM-SS'));

[file,path] = uiputfile( '\*.avi','Выбор имени файла', defname );

if file == 0, return; end

selpath = fullfile(path,file);

app.parent\_app.m.Export\_figure\_all\_video( selpath, app.fig\_vid\_all\_fps.Value );

end

% Button pushed function: comment

function commentButtonPushed(app, event)

defname = sprintf('output\\comment %s.txt', datestr(now, 'yyyy-mm-dd HH-MM-SS'));

[file,path] = uiputfile( '\*.txt','Выбор имени файла', defname );

if file == 0, return; end

selpath = fullfile(path,file);

app.parent\_app.m.Export\_comment(selpath);

end

% Button pushed function: Export\_all\_fields\_to\_one\_file

function Export\_all\_fields\_to\_one\_fileButtonPushed(app, event)

defname = sprintf('output\\fields %s.dat', datestr(now, 'yyyy-mm-dd HH-MM-SS'));

[file,path] = uiputfile( '\*','Выбор имени файла', defname );

if file == 0, return; end

selpath = fullfile(path,file);

app.parent\_app.m.Export\_all\_fields\_to\_one\_file(selpath);

end

% Button pushed function: Export\_all\_fields\_to\_separate\_files

function Export\_all\_fields\_to\_separate\_filesButtonPushed(app, event)

defname = sprintf('output\\fields %s.dat', datestr(now, 'yyyy-mm-dd HH-MM-SS'));

[file,path] = uiputfile( '\*','Выбор имени файла', defname );

if file == 0, return; end

selpath = fullfile(path,file);

app.parent\_app.m.Export\_all\_fields\_to\_separate\_files(selpath);

end

% Button pushed function: Export\_time\_series

function Export\_time\_seriesButtonPushed(app, event)

defname = sprintf('output\\time series %s.dat', datestr(now, 'yyyy-mm-dd HH-MM-SS'));

[file,path] = uiputfile( '\*','Выбор имени файла', defname );

if file == 0, return; end

selpath = fullfile(path,file);

pos\_ij\_arr = app.parent\_app.m.Get\_pos\_ij\_arr\_\_of\_selected\_point();

if isempty(pos\_ij\_arr)

app.Plot\_time\_series\_pos\_ij\_arr.Value = {'Точки не выбраны'};

return

end

pos = app.parent\_app.m.Get\_pos\_by\_pos\_ij\_arr( pos\_ij\_arr );

text = cell(size(pos\_ij\_arr,1),1);

for i = 1:size(pos\_ij\_arr,1)

pos\_i = pos\_ij\_arr(i,1);

pos\_j = pos\_ij\_arr(i,2);

x\_px = pos(i,1);

y\_px = pos(i,2);

x\_m = pos(i,3);

y\_m = pos(i,4);

text{i} = sprintf(...

'i %d, j %d, (%d, %d) px, (%.3f, %.3f) м',...

pos\_i, pos\_j,...

x\_px, y\_px,...

x\_m, y\_m);

end

app.Export\_time\_series\_sp.Value = text;

app.parent\_app.m.Export\_time\_series( selpath, pos\_ij\_arr );

end

% Button pushed function: Export\_mean\_field\_to\_files

function Export\_mean\_field\_to\_filesButtonPushed(app, event)

defname = sprintf('output\\mean field %s.dat', datestr(now, 'yyyy-mm-dd HH-MM-SS'));

[file,path] = uiputfile( '\*','Выбор имени файла', defname );

if file == 0, return; end

selpath = fullfile(path,file);

profile\_y\_\_pos\_i = [];

profile\_x\_\_pos\_j = [];

pos\_ij\_arr = app.parent\_app.m.Get\_pos\_ij\_arr\_\_of\_selected\_point();

if app.Export\_mean\_field\_to\_files\_profiles.Value

profile\_y\_\_pos\_i = pos\_ij\_arr(:,1);

end

if app.Export\_mean\_field\_to\_files\_profiles\_2.Value

profile\_x\_\_pos\_j = pos\_ij\_arr(:,2);

end

app.parent\_app.m.Export\_mean\_field\_to\_files( selpath, profile\_y\_\_pos\_i, profile\_x\_\_pos\_j );

end

end

% App initialization and construction

methods (Access = private)

% Create UIFigure and components

function createComponents(app)

% Create UIFigure

app.UIFigure = uifigure;

app.UIFigure.Position = [100 100 524 438];

app.UIFigure.Name = 'Экспорт';

% Create fig\_im\_one

app.fig\_im\_one = uibutton(app.UIFigure, 'push');

app.fig\_im\_one.ButtonPushedFcn = createCallbackFcn(app, @fig\_im\_onePushed, true);

app.fig\_im\_one.Position = [16 402 226 22];

app.fig\_im\_one.Text = 'Сохранить поле как изображение';

% Create fig\_im\_all

app.fig\_im\_all = uibutton(app.UIFigure, 'push');

app.fig\_im\_all.ButtonPushedFcn = createCallbackFcn(app, @fig\_im\_allButtonPushed, true);

app.fig\_im\_all.Position = [16 369 226 22];

app.fig\_im\_all.Text = 'Сохранить все поля как изображения';

% Create fig\_vid\_all

app.fig\_vid\_all = uibutton(app.UIFigure, 'push');

app.fig\_vid\_all.ButtonPushedFcn = createCallbackFcn(app, @fig\_vid\_allButtonPushed, true);

app.fig\_vid\_all.Position = [16 336 226 22];

app.fig\_vid\_all.Text = 'Сохранить все поля как видео';

% Create Label

app.Label = uilabel(app.UIFigure);

app.Label.HorizontalAlignment = 'right';

app.Label.Position = [269 336 139 22];

app.Label.Text = 'Частота кадров в видео';

% Create fig\_vid\_all\_fps

app.fig\_vid\_all\_fps = uispinner(app.UIFigure);

app.fig\_vid\_all\_fps.Limits = [0.1 Inf];

app.fig\_vid\_all\_fps.Position = [423 336 76 22];

app.fig\_vid\_all\_fps.Value = 5;

% Create comment

app.comment = uibutton(app.UIFigure, 'push');

app.comment.ButtonPushedFcn = createCallbackFcn(app, @commentButtonPushed, true);

app.comment.Position = [17 282 226 22];

app.comment.Text = 'Сохранить описание как текст';

% Create Export\_all\_fields\_to\_one\_file

app.Export\_all\_fields\_to\_one\_file = uibutton(app.UIFigure, 'push');

app.Export\_all\_fields\_to\_one\_file.ButtonPushedFcn = createCallbackFcn(app, @Export\_all\_fields\_to\_one\_fileButtonPushed, true);

app.Export\_all\_fields\_to\_one\_file.Position = [15 225 226 22];

app.Export\_all\_fields\_to\_one\_file.Text = 'Сохранить все поля в единый файл';

% Create Export\_all\_fields\_to\_separate\_files

app.Export\_all\_fields\_to\_separate\_files = uibutton(app.UIFigure, 'push');

app.Export\_all\_fields\_to\_separate\_files.ButtonPushedFcn = createCallbackFcn(app, @Export\_all\_fields\_to\_separate\_filesButtonPushed, true);

app.Export\_all\_fields\_to\_separate\_files.Position = [13 188 226 22];

app.Export\_all\_fields\_to\_separate\_files.Text = 'Сохранить все поля в набор файлов';

% Create Export\_time\_series

app.Export\_time\_series = uibutton(app.UIFigure, 'push');

app.Export\_time\_series.ButtonPushedFcn = createCallbackFcn(app, @Export\_time\_seriesButtonPushed, true);

app.Export\_time\_series.Position = [17 133 226 22];

app.Export\_time\_series.Text = 'Сохранить временные реализации';

% Create Export\_mean\_field\_to\_files

app.Export\_mean\_field\_to\_files = uibutton(app.UIFigure, 'push');

app.Export\_mean\_field\_to\_files.ButtonPushedFcn = createCallbackFcn(app, @Export\_mean\_field\_to\_filesButtonPushed, true);

app.Export\_mean\_field\_to\_files.Position = [16 31 226 22];

app.Export\_mean\_field\_to\_files.Text = 'Сохранить среднее поле';

% Create Export\_mean\_field\_to\_files\_profiles

app.Export\_mean\_field\_to\_files\_profiles = uicheckbox(app.UIFigure);

app.Export\_mean\_field\_to\_files\_profiles.Text = 'Сохранять вертикальные профили';

app.Export\_mean\_field\_to\_files\_profiles.Position = [269 31 218 22];

% Create Label\_2

app.Label\_2 = uilabel(app.UIFigure);

app.Label\_2.HorizontalAlignment = 'right';

app.Label\_2.Position = [17 103 108 22];

app.Label\_2.Text = {'Выбранные точки:'; ''};

% Create Export\_time\_series\_sp

app.Export\_time\_series\_sp = uitextarea(app.UIFigure);

app.Export\_time\_series\_sp.Editable = 'off';

app.Export\_time\_series\_sp.Position = [140 72 381 55];

% Create Export\_mean\_field\_to\_files\_profiles\_2

app.Export\_mean\_field\_to\_files\_profiles\_2 = uicheckbox(app.UIFigure);

app.Export\_mean\_field\_to\_files\_profiles\_2.Text = 'Сохранять горизонтальные профили';

app.Export\_mean\_field\_to\_files\_profiles\_2.Position = [269 10 230 22];

end

end

methods (Access = public)

% Construct app

function app = PIV\_db\_export(varargin)

% Create and configure components

createComponents(app)

% Register the app with App Designer

registerApp(app, app.UIFigure)

% Execute the startup function

runStartupFcn(app, @(app)startupFcn(app, varargin{:}))

if nargout == 0

clear app

end

end

% Code that executes before app deletion

function delete(app)

% Delete UIFigure when app is deleted

delete(app.UIFigure)

end

end

end

## PIV\_db\_fi\_params.mlapp

classdef PIV\_db\_fi\_params < matlab.apps.AppBase

% Properties that correspond to app components

properties (Access = public)

UIFigure matlab.ui.Figure

Button\_ok matlab.ui.control.Button

Button\_cancel matlab.ui.control.Button

Reset matlab.ui.control.Button

Label matlab.ui.control.Label

CC\_ratio\_limits1 matlab.ui.control.NumericEditField

CC\_ratio\_limits2 matlab.ui.control.NumericEditField

Label\_2 matlab.ui.control.Label

CC\_maxRaitio\_max matlab.ui.control.NumericEditField

Label\_3 matlab.ui.control.Label

mean\_im1\_limits1 matlab.ui.control.NumericEditField

mean\_im1\_limits2 matlab.ui.control.NumericEditField

XEditFieldLabel matlab.ui.control.Label

xd\_limits1 matlab.ui.control.NumericEditField

xd\_limits2 matlab.ui.control.NumericEditField

YEditFieldLabel matlab.ui.control.Label

yd\_limits1 matlab.ui.control.NumericEditField

yd\_limits2 matlab.ui.control.NumericEditField

EditField\_11Label matlab.ui.control.Label

d\_limits1 matlab.ui.control.NumericEditField

d\_limits2 matlab.ui.control.NumericEditField

median\_space\_\_on matlab.ui.control.CheckBox

Label\_5 matlab.ui.control.Label

median\_space\_\_window\_size1 matlab.ui.control.Spinner

median\_space\_\_window\_size2 matlab.ui.control.Spinner

Label\_6 matlab.ui.control.Label

median\_space\_\_max\_d\_difference matlab.ui.control.NumericEditField

interp\_space\_\_on matlab.ui.control.CheckBox

Label\_8 matlab.ui.control.Label

interp\_space\_\_Method matlab.ui.control.DropDown

Label\_9 matlab.ui.control.Label

interp\_space\_\_ExtrapolationMethod matlab.ui.control.DropDown

median\_time\_\_on matlab.ui.control.CheckBox

Label\_11 matlab.ui.control.Label

median\_time\_\_dynamic\_\_window\_size matlab.ui.control.Spinner

Label\_12 matlab.ui.control.Label

median\_time\_\_max\_d\_difference matlab.ui.control.NumericEditField

local\_filtering\_\_on matlab.ui.control.CheckBox

median\_time\_\_dynamic\_\_on matlab.ui.control.CheckBox

interp\_time\_\_on matlab.ui.control.CheckBox

Label\_13 matlab.ui.control.Label

interp\_time\_\_Method matlab.ui.control.DropDown

Label\_14 matlab.ui.control.Label

interp\_time\_\_max\_gap\_length matlab.ui.control.Spinner

end

properties (Access = private)

parent\_app

fi\_params\_backup Filter\_and\_interpolation\_params

end

methods (Access = private)

function Load\_values(app)

fi\_params = app.parent\_app.m.p.fi;

app.local\_filtering\_\_on.Value = fi\_params.local\_filtering\_\_on;

app.CC\_ratio\_limits1.Value = fi\_params.CC\_ratio\_limits(1);

app.CC\_ratio\_limits2.Value = fi\_params.CC\_ratio\_limits(2);

app.CC\_maxRaitio\_max.Value = fi\_params.CC\_maxRaitio\_max;

app.mean\_im1\_limits1.Value = fi\_params.mean\_im1\_limits(1);

app.mean\_im1\_limits2.Value = fi\_params.mean\_im1\_limits(2);

app.xd\_limits1.Value = fi\_params.xd\_limits(1);

app.xd\_limits2.Value = fi\_params.xd\_limits(2);

app.yd\_limits1.Value = fi\_params.yd\_limits(1);

app.yd\_limits2.Value = fi\_params.yd\_limits(2);

app.d\_limits1.Value = fi\_params.d\_limits(1);

app.d\_limits2.Value = fi\_params.d\_limits(2);

%медианная фильтрация по пространству:

app.median\_space\_\_on.Value = fi\_params.median\_space\_\_on;

app.median\_space\_\_window\_size1.Value = fi\_params.median\_space\_\_window\_size(1);

app.median\_space\_\_window\_size2.Value = fi\_params.median\_space\_\_window\_size(2);

app.median\_space\_\_max\_d\_difference.Value = fi\_params.median\_space\_\_max\_d\_difference;

%медианная фильтрация по времени:

app.median\_time\_\_on.Value = fi\_params.median\_time\_\_on;

app.median\_time\_\_max\_d\_difference.Value = fi\_params.median\_time\_\_max\_d\_difference;

app.median\_time\_\_dynamic\_\_on.Value = fi\_params.median\_time\_\_dynamic\_\_on;

app.median\_time\_\_dynamic\_\_window\_size.Value = fi\_params.median\_time\_\_dynamic\_\_window\_size;

%% Интерполяция:

app.interp\_space\_\_on.Value = fi\_params.interp\_space\_\_on;

app.interp\_space\_\_Method.Items = { 'linear', 'nearest', 'natural' };

app.interp\_space\_\_Method.Value = fi\_params.interp\_space\_\_Method;

app.interp\_space\_\_ExtrapolationMethod.Items = { 'nearest', 'linear', 'none' };

app.interp\_space\_\_ExtrapolationMethod.Value = fi\_params.interp\_space\_\_ExtrapolationMethod;

app.interp\_time\_\_on.Value = fi\_params.interp\_time\_\_on;

app.interp\_time\_\_Method.Items = { 'linear', 'nearest', 'next', 'previous', 'spline', 'pchip', 'cubic', 'makima' };

app.interp\_time\_\_Method.Value = fi\_params.interp\_time\_\_Method;

app.interp\_time\_\_max\_gap\_length.Value = fi\_params.interp\_time\_\_max\_gap\_length;

end

end

methods (Access = private)

% Code that executes after component creation

function startupFcn(app, parent\_app)

app.parent\_app = parent\_app;

app.fi\_params\_backup = app.parent\_app.m.p.fi;

app.Load\_values();

end

% Button pushed function: Button\_ok

function Button\_okPushed(app, event)

fi\_params = app.parent\_app.m.p.fi;

fi\_params.local\_filtering\_\_on = app.local\_filtering\_\_on.Value;

fi\_params.CC\_ratio\_limits(1) = app.CC\_ratio\_limits1.Value;

fi\_params.CC\_ratio\_limits(2) = app.CC\_ratio\_limits2.Value;

fi\_params.CC\_maxRaitio\_max = app.CC\_maxRaitio\_max.Value;

fi\_params.mean\_im1\_limits(1) = app.mean\_im1\_limits1.Value;

fi\_params.mean\_im1\_limits(2) = app.mean\_im1\_limits2.Value;

fi\_params.xd\_limits(1) = app.xd\_limits1.Value;

fi\_params.xd\_limits(2) = app.xd\_limits2.Value;

fi\_params.yd\_limits(1) = app.yd\_limits1.Value;

fi\_params.yd\_limits(2) = app.yd\_limits2.Value;

fi\_params.d\_limits(1) = app.d\_limits1.Value;

fi\_params.d\_limits(2) = app.d\_limits2.Value;

%медианная фильтрация:

fi\_params.median\_space\_\_on = app.median\_space\_\_on.Value;

fi\_params.median\_space\_\_window\_size(1) = app.median\_space\_\_window\_size1.Value;

fi\_params.median\_space\_\_window\_size(2) = app.median\_space\_\_window\_size2.Value;

fi\_params.median\_space\_\_max\_d\_difference = app.median\_space\_\_max\_d\_difference.Value;

%медианная фильтрация по времени:

fi\_params.median\_time\_\_on = app.median\_time\_\_on.Value;

fi\_params.median\_time\_\_max\_d\_difference = app.median\_time\_\_max\_d\_difference.Value;

fi\_params.median\_time\_\_dynamic\_\_on = app.median\_time\_\_dynamic\_\_on.Value;

fi\_params.median\_time\_\_dynamic\_\_window\_size = app.median\_time\_\_dynamic\_\_window\_size.Value;

%% Интерполяция:

fi\_params.interp\_space\_\_on = app.interp\_space\_\_on.Value;

fi\_params.interp\_space\_\_Method = app.interp\_space\_\_Method.Value;

fi\_params.interp\_space\_\_ExtrapolationMethod = app.interp\_space\_\_ExtrapolationMethod.Value;

fi\_params.interp\_time\_\_on = app.interp\_time\_\_on.Value;

fi\_params.interp\_time\_\_Method = app.interp\_time\_\_Method.Value;

fi\_params.interp\_time\_\_max\_gap\_length = app.interp\_time\_\_max\_gap\_length.Value;

app.parent\_app.m.Set\_fi\_params(fi\_params);

end

% Button pushed function: Button\_cancel

function Button\_cancelPushed(app, event)

app.parent\_app.m.Set\_fi\_params(app.fi\_params\_backup);

delete(app);

end

% Button pushed function: Reset

function ResetButtonPushed(app, event)

fi\_params = Filter\_and\_interpolation\_params();

app.parent\_app.m.Set\_fi\_params(fi\_params);

app.Load\_values();

end

end

% App initialization and construction

methods (Access = private)

% Create UIFigure and components

function createComponents(app)

% Create UIFigure

app.UIFigure = uifigure;

app.UIFigure.Position = [100 100 716 722];

app.UIFigure.Name = 'Параметры фильтрации и интерполяции';

% Create Button\_ok

app.Button\_ok = uibutton(app.UIFigure, 'push');

app.Button\_ok.ButtonPushedFcn = createCallbackFcn(app, @Button\_okPushed, true);

app.Button\_ok.Position = [583 21 100 22];

app.Button\_ok.Text = 'Применить';

% Create Button\_cancel

app.Button\_cancel = uibutton(app.UIFigure, 'push');

app.Button\_cancel.ButtonPushedFcn = createCallbackFcn(app, @Button\_cancelPushed, true);

app.Button\_cancel.Position = [469 21 100 22];

app.Button\_cancel.Text = 'Отмена';

% Create Reset

app.Reset = uibutton(app.UIFigure, 'push');

app.Reset.ButtonPushedFcn = createCallbackFcn(app, @ResetButtonPushed, true);

app.Reset.Position = [360 21 100 22];

app.Reset.Text = 'Сбросить';

% Create Label

app.Label = uilabel(app.UIFigure);

app.Label.HorizontalAlignment = 'right';

app.Label.Position = [195 652 206 22];

app.Label.Text = 'Пределы соотношения максимумов';

% Create CC\_ratio\_limits1

app.CC\_ratio\_limits1 = uieditfield(app.UIFigure, 'numeric');

app.CC\_ratio\_limits1.Position = [416 652 100 22];

% Create CC\_ratio\_limits2

app.CC\_ratio\_limits2 = uieditfield(app.UIFigure, 'numeric');

app.CC\_ratio\_limits2.Position = [528 653 100 22];

% Create Label\_2

app.Label\_2 = uilabel(app.UIFigure);

app.Label\_2.HorizontalAlignment = 'right';

app.Label\_2.Position = [164 619 237 22];

app.Label\_2.Text = 'Максимальное значение максимума ККФ';

% Create CC\_maxRaitio\_max

app.CC\_maxRaitio\_max = uieditfield(app.UIFigure, 'numeric');

app.CC\_maxRaitio\_max.Position = [416 619 100 22];

% Create Label\_3

app.Label\_3 = uilabel(app.UIFigure);

app.Label\_3.HorizontalAlignment = 'right';

app.Label\_3.Position = [125 586 277 22];

app.Label\_3.Text = 'Пределы средней по окну яркости изображений';

% Create mean\_im1\_limits1

app.mean\_im1\_limits1 = uieditfield(app.UIFigure, 'numeric');

app.mean\_im1\_limits1.Position = [417 586 100 22];

% Create mean\_im1\_limits2

app.mean\_im1\_limits2 = uieditfield(app.UIFigure, 'numeric');

app.mean\_im1\_limits2.Position = [529 586 100 22];

% Create XEditFieldLabel

app.XEditFieldLabel = uilabel(app.UIFigure);

app.XEditFieldLabel.HorizontalAlignment = 'right';

app.XEditFieldLabel.Position = [258 553 144 22];

app.XEditFieldLabel.Text = 'Пределы смещения по X';

% Create xd\_limits1

app.xd\_limits1 = uieditfield(app.UIFigure, 'numeric');

app.xd\_limits1.Position = [417 553 100 22];

% Create xd\_limits2

app.xd\_limits2 = uieditfield(app.UIFigure, 'numeric');

app.xd\_limits2.Position = [529 553 100 22];

% Create YEditFieldLabel

app.YEditFieldLabel = uilabel(app.UIFigure);

app.YEditFieldLabel.HorizontalAlignment = 'right';

app.YEditFieldLabel.Position = [258 520 144 22];

app.YEditFieldLabel.Text = 'Пределы смещения по Y';

% Create yd\_limits1

app.yd\_limits1 = uieditfield(app.UIFigure, 'numeric');

app.yd\_limits1.Position = [417 520 100 22];

% Create yd\_limits2

app.yd\_limits2 = uieditfield(app.UIFigure, 'numeric');

app.yd\_limits2.Position = [529 520 100 22];

% Create EditField\_11Label

app.EditField\_11Label = uilabel(app.UIFigure);

app.EditField\_11Label.HorizontalAlignment = 'right';

app.EditField\_11Label.Position = [240 487 161 22];

app.EditField\_11Label.Text = 'Пределы модуля смещения';

% Create d\_limits1

app.d\_limits1 = uieditfield(app.UIFigure, 'numeric');

app.d\_limits1.Position = [416 487 100 22];

% Create d\_limits2

app.d\_limits2 = uieditfield(app.UIFigure, 'numeric');

app.d\_limits2.Position = [528 487 100 22];

% Create median\_space\_\_on

app.median\_space\_\_on = uicheckbox(app.UIFigure);

app.median\_space\_\_on.Text = 'Медианная фильтрация по пространству';

app.median\_space\_\_on.Position = [64 454 253 22];

% Create Label\_5

app.Label\_5 = uilabel(app.UIFigure);

app.Label\_5.HorizontalAlignment = 'right';

app.Label\_5.Position = [326 421 75 22];

app.Label\_5.Text = 'Размер окна';

% Create median\_space\_\_window\_size1

app.median\_space\_\_window\_size1 = uispinner(app.UIFigure);

app.median\_space\_\_window\_size1.Limits = [1 Inf];

app.median\_space\_\_window\_size1.Position = [416 421 100 22];

app.median\_space\_\_window\_size1.Value = 1;

% Create median\_space\_\_window\_size2

app.median\_space\_\_window\_size2 = uispinner(app.UIFigure);

app.median\_space\_\_window\_size2.Limits = [1 Inf];

app.median\_space\_\_window\_size2.Position = [528 421 100 22];

app.median\_space\_\_window\_size2.Value = 1;

% Create Label\_6

app.Label\_6 = uilabel(app.UIFigure);

app.Label\_6.HorizontalAlignment = 'right';

app.Label\_6.Position = [116 388 285 22];

app.Label\_6.Text = 'Максимальное отличие смещения от медианного';

% Create median\_space\_\_max\_d\_difference

app.median\_space\_\_max\_d\_difference = uieditfield(app.UIFigure, 'numeric');

app.median\_space\_\_max\_d\_difference.Position = [416 388 100 22];

% Create interp\_space\_\_on

app.interp\_space\_\_on = uicheckbox(app.UIFigure);

app.interp\_space\_\_on.Text = 'Интерполяция по пространству';

app.interp\_space\_\_on.Position = [64 223 198 22];

% Create Label\_8

app.Label\_8 = uilabel(app.UIFigure);

app.Label\_8.HorizontalAlignment = 'right';

app.Label\_8.Position = [266 190 129 22];

app.Label\_8.Text = 'Способ интерполяции';

% Create interp\_space\_\_Method

app.interp\_space\_\_Method = uidropdown(app.UIFigure);

app.interp\_space\_\_Method.Items = {};

app.interp\_space\_\_Method.Position = [410 190 100 22];

app.interp\_space\_\_Method.Value = {};

% Create Label\_9

app.Label\_9 = uilabel(app.UIFigure);

app.Label\_9.HorizontalAlignment = 'right';

app.Label\_9.Position = [262 157 133 22];

app.Label\_9.Text = 'Способ экстраполяции';

% Create interp\_space\_\_ExtrapolationMethod

app.interp\_space\_\_ExtrapolationMethod = uidropdown(app.UIFigure);

app.interp\_space\_\_ExtrapolationMethod.Items = {};

app.interp\_space\_\_ExtrapolationMethod.Position = [410 157 100 22];

app.interp\_space\_\_ExtrapolationMethod.Value = {};

% Create median\_time\_\_on

app.median\_time\_\_on = uicheckbox(app.UIFigure);

app.median\_time\_\_on.Text = 'Медианная фильтрация по времени';

app.median\_time\_\_on.Position = [64 355 226 22];

% Create Label\_11

app.Label\_11 = uilabel(app.UIFigure);

app.Label\_11.HorizontalAlignment = 'right';

app.Label\_11.Position = [146 289 255 22];

app.Label\_11.Text = 'Размер окна при динамической фильтрации';

% Create median\_time\_\_dynamic\_\_window\_size

app.median\_time\_\_dynamic\_\_window\_size = uispinner(app.UIFigure);

app.median\_time\_\_dynamic\_\_window\_size.Limits = [1 Inf];

app.median\_time\_\_dynamic\_\_window\_size.Position = [416 289 100 22];

app.median\_time\_\_dynamic\_\_window\_size.Value = 1;

% Create Label\_12

app.Label\_12 = uilabel(app.UIFigure);

app.Label\_12.HorizontalAlignment = 'right';

app.Label\_12.Position = [117 256 285 22];

app.Label\_12.Text = 'Максимальное отличие смещения от медианного';

% Create median\_time\_\_max\_d\_difference

app.median\_time\_\_max\_d\_difference = uieditfield(app.UIFigure, 'numeric');

app.median\_time\_\_max\_d\_difference.Position = [417 256 100 22];

% Create local\_filtering\_\_on

app.local\_filtering\_\_on = uicheckbox(app.UIFigure);

app.local\_filtering\_\_on.Text = 'Локальная фильтрация';

app.local\_filtering\_\_on.Position = [74 685 153 22];

% Create median\_time\_\_dynamic\_\_on

app.median\_time\_\_dynamic\_\_on = uicheckbox(app.UIFigure);

app.median\_time\_\_dynamic\_\_on.Text = 'Динамическая фильтрация';

app.median\_time\_\_dynamic\_\_on.Position = [339 322 174 22];

% Create interp\_time\_\_on

app.interp\_time\_\_on = uicheckbox(app.UIFigure);

app.interp\_time\_\_on.Text = 'Интерполяция по времени';

app.interp\_time\_\_on.Position = [64 125 170 22];

% Create Label\_13

app.Label\_13 = uilabel(app.UIFigure);

app.Label\_13.HorizontalAlignment = 'right';

app.Label\_13.Position = [266 93 129 22];

app.Label\_13.Text = 'Способ интерполяции';

% Create interp\_time\_\_Method

app.interp\_time\_\_Method = uidropdown(app.UIFigure);

app.interp\_time\_\_Method.Items = {};

app.interp\_time\_\_Method.Position = [410 93 100 22];

app.interp\_time\_\_Method.Value = {};

% Create Label\_14

app.Label\_14 = uilabel(app.UIFigure);

app.Label\_14.HorizontalAlignment = 'right';

app.Label\_14.Position = [198 61 197 22];

app.Label\_14.Text = 'Максимальная длина промежутка';

% Create interp\_time\_\_max\_gap\_length

app.interp\_time\_\_max\_gap\_length = uispinner(app.UIFigure);

app.interp\_time\_\_max\_gap\_length.Limits = [1 Inf];

app.interp\_time\_\_max\_gap\_length.Position = [410 61 100 22];

app.interp\_time\_\_max\_gap\_length.Value = 1;

end

end

methods (Access = public)

% Construct app

function app = PIV\_db\_fi\_params(varargin)

% Create and configure components

createComponents(app)

% Register the app with App Designer

registerApp(app, app.UIFigure)

% Execute the startup function

runStartupFcn(app, @(app)startupFcn(app, varargin{:}))

if nargout == 0

clear app

end

end

% Code that executes before app deletion

function delete(app)

% Delete UIFigure when app is deleted

delete(app.UIFigure)

end

end

end

## PIV\_db\_ipp.mlapp

classdef PIV\_db\_ipp < matlab.apps.AppBase

% Properties that correspond to app components

properties (Access = public)

UIFigure matlab.ui.Figure

CheckBox\_bg\_on matlab.ui.control.CheckBox

CheckBox\_mask\_on matlab.ui.control.CheckBox

CheckBox\_levels\_on matlab.ui.control.CheckBox

Button\_Apply matlab.ui.control.Button

Label matlab.ui.control.Label

color\_channel matlab.ui.control.Spinner

SpinnerLabel matlab.ui.control.Label

Spinner\_levels\_min matlab.ui.control.Spinner

Spinner\_2Label matlab.ui.control.Label

Spinner\_levels\_max matlab.ui.control.Spinner

Button\_cancel matlab.ui.control.Button

UIAxes\_bg matlab.ui.control.UIAxes

UIAxes\_mask matlab.ui.control.UIAxes

ButtonReset matlab.ui.control.Button

Panel matlab.ui.container.Panel

Button\_bg\_calc matlab.ui.control.Button

EditField\_2Label matlab.ui.control.Label

EditField\_bg\_calc\_prctile matlab.ui.control.NumericEditField

Label\_3 matlab.ui.control.Label

EditField\_bg\_calc\_N matlab.ui.control.NumericEditField

Panel\_2 matlab.ui.container.Panel

Button\_bg\_path matlab.ui.control.Button

Button\_bg\_load matlab.ui.control.Button

EditFieldLabel matlab.ui.control.Label

EditField\_bg\_path matlab.ui.control.EditField

Panel\_3 matlab.ui.container.Panel

Button\_mask\_path matlab.ui.control.Button

EditField\_4Label matlab.ui.control.Label

EditField\_mask\_path matlab.ui.control.EditField

Label\_4 matlab.ui.control.Label

EditField\_mask\_source\_color1 matlab.ui.control.NumericEditField

EditField\_mask\_source\_color2 matlab.ui.control.NumericEditField

Button\_mask\_load matlab.ui.control.Button

EditField\_mask\_source\_color3 matlab.ui.control.NumericEditField

Button\_mask\_source\_color\_pick matlab.ui.control.Button

Label\_5 matlab.ui.control.Label

EditField\_mask\_pick\_values\_interval\_3 matlab.ui.control.NumericEditField

end

properties (Access = public)

parent\_app

p\_backup Image\_preprocessor\_params

end

methods (Access = private)

function Redraw\_bg\_and\_mask(app)

cla(app.UIAxes\_bg);

cla(app.UIAxes\_mask);

app.UIAxes\_bg.XTick = [];

app.UIAxes\_bg.YTick = [];

app.UIAxes\_mask.XTick = [];

app.UIAxes\_mask.YTick = [];

if ~isempty(app.parent\_app.m.p.ipp.bg)

imagesc(app.parent\_app.m.p.ipp.bg,'Parent',app.UIAxes\_bg);

axis(app.UIAxes\_bg, 'image');

end

if ~isempty(app.parent\_app.m.p.ipp.mask)

imagesc(app.parent\_app.m.p.ipp.mask,'Parent',app.UIAxes\_mask);

axis(app.UIAxes\_mask, 'image');

% colormap(app.UIAxes\_mask, 'grey');

end

end

function Update\_p(app)

p = app.parent\_app.m.p.ipp;

p.color\_channel = app.color\_channel.Value;

p.bg\_on = app.CheckBox\_bg\_on.Value;

p.bg\_source\_file\_path = app.EditField\_bg\_path.Value;

p.bg\_source\_auto\_prctile = app.EditField\_bg\_calc\_prctile.Value;

p.bg\_source\_auto\_N = app.EditField\_bg\_calc\_N.Value;

p.mask\_on = app.CheckBox\_mask\_on.Value;

p.mask\_set\_value = app.EditField\_mask\_pick\_values\_interval\_3.Value;

p.mask\_source\_path = app.EditField\_mask\_path.Value;

p.mask\_source\_color(1) = app.EditField\_mask\_source\_color1.Value;

p.mask\_source\_color(2) = app.EditField\_mask\_source\_color2.Value;

p.mask\_source\_color(3) = app.EditField\_mask\_source\_color3.Value;

p.levels\_on = app.CheckBox\_levels\_on.Value;

p.levels\_limits(1) = app.Spinner\_levels\_min.Value;

p.levels\_limits(2) = app.Spinner\_levels\_max.Value;

app.parent\_app.m.Set\_ipp\_params(p);

app.Load\_values\_from\_parent\_app();

app.parent\_app.m.Redraw\_fields();

end

function Load\_values\_from\_parent\_app(app)

p = app.parent\_app.m.p.ipp;

app.p\_backup = p; %сохраняем на случай отмены

app.color\_channel.Value = p.color\_channel;

app.CheckBox\_bg\_on.Value = p.bg\_on;

app.EditField\_bg\_path.Value = p.bg\_source\_file\_path;

app.EditField\_bg\_calc\_prctile.Value = p.bg\_source\_auto\_prctile;

app.EditField\_bg\_calc\_N.Value = p.bg\_source\_auto\_N;

app.CheckBox\_mask\_on.Value = p.mask\_on;

app.EditField\_mask\_pick\_values\_interval\_3.Value = p.mask\_set\_value;

app.EditField\_mask\_path.Value = p.mask\_source\_path;

app.EditField\_mask\_source\_color1.Value = p.mask\_source\_color(1);

app.EditField\_mask\_source\_color2.Value = p.mask\_source\_color(2);

app.EditField\_mask\_source\_color3.Value = p.mask\_source\_color(3);

app.CheckBox\_levels\_on.Value = p.levels\_on;

app.Spinner\_levels\_min.Value = p.levels\_limits(1);

app.Spinner\_levels\_max.Value = p.levels\_limits(2);

app.Redraw\_bg\_and\_mask();

end

end

methods (Access = private)

% Code that executes after component creation

function startupFcn(app, parent\_app)

app.parent\_app = parent\_app;

app.Load\_values\_from\_parent\_app();

end

% Button pushed function: Button\_bg\_load

function Button\_bg\_loadPushed(app, event)

if ~isempty(app.EditField\_bg\_path.Value)

app.parent\_app.m.Load\_and\_use\_bg(app.EditField\_bg\_path.Value);

app.Load\_values\_from\_parent\_app();

end

end

% Button pushed function: Button\_mask\_load

function Button\_mask\_loadPushed(app, event)

if ~isempty(app.EditField\_mask\_path.Value)

app.parent\_app.m.Load\_and\_use\_mask(...

app.EditField\_mask\_path.Value,...

[app.EditField\_mask\_source\_color1.Value,...

app.EditField\_mask\_source\_color2.Value,...

app.EditField\_mask\_source\_color3.Value]);

app.Load\_values\_from\_parent\_app();

end

end

% Button pushed function: Button\_bg\_path

function Button\_bg\_pathPushed(app, event)

[file,path] = uigetfile('\*');

if file ~= 0

selpath = fullfile(path,file);

app.EditField\_bg\_path.Value = selpath;

app.parent\_app.m.Load\_and\_use\_bg(app.EditField\_bg\_path.Value);

app.Load\_values\_from\_parent\_app();

end

end

% Button pushed function: Button\_mask\_path

function Button\_mask\_pathPushed(app, event)

[file,path] = uigetfile('\*');

if file ~= 0

selpath = fullfile(path,file);

app.EditField\_mask\_path.Value = selpath;

end

end

% Button pushed function: Button\_cancel

function Button\_cancelPushed(app, event)

app.parent\_app.m.Set\_ipp\_params( app.p\_backup );

delete(app);

end

% Button pushed function: Button\_Apply

function Button\_ApplyPushed(app, event)

app.Update\_p();

end

% Button pushed function: Button\_bg\_calc

function Button\_bg\_calcPushed(app, event)

app.parent\_app.m.Calc\_and\_use\_bg( app.EditField\_bg\_calc\_N.Value, app.EditField\_bg\_calc\_prctile.Value );

app.parent\_app.m.Redraw\_fields();

app.Load\_values\_from\_parent\_app();

end

% Button pushed function: ButtonReset

function ButtonResetButtonPushed(app, event)

app.parent\_app.m.Set\_ipp\_params(Image\_preprocessor\_params());

app.parent\_app.m.Redraw\_fields();

app.Load\_values\_from\_parent\_app();

end

% Button pushed function: Button\_mask\_source\_color\_pick

function Button\_mask\_source\_color\_pickPushed(app, event)

im\_path = app.EditField\_mask\_path.Value;

if ~isempty(im\_path)

f = figure('name','Выберите мышкой пиксель, принадлежащий маске');

im = imread(im\_path);

imshow(im);

p = round(ginput(1));

close(f);

c = double(squeeze(im(p(2),p(1),:)));

app.EditField\_mask\_source\_color1.Value = c(1);

app.EditField\_mask\_source\_color2.Value = c(2);

app.EditField\_mask\_source\_color3.Value = c(3);

end

end

end

% App initialization and construction

methods (Access = private)

% Create UIFigure and components

function createComponents(app)

% Create UIFigure

app.UIFigure = uifigure;

app.UIFigure.Position = [100 100 658 715];

app.UIFigure.Name = 'Параметры изображений';

% Create CheckBox\_bg\_on

app.CheckBox\_bg\_on = uicheckbox(app.UIFigure);

app.CheckBox\_bg\_on.Text = 'Вычитать фон';

app.CheckBox\_bg\_on.Position = [19 636 102 22];

% Create CheckBox\_mask\_on

app.CheckBox\_mask\_on = uicheckbox(app.UIFigure);

app.CheckBox\_mask\_on.Text = 'Использовать маску';

app.CheckBox\_mask\_on.Position = [21 360 135 22];

% Create CheckBox\_levels\_on

app.CheckBox\_levels\_on = uicheckbox(app.UIFigure);

app.CheckBox\_levels\_on.Text = 'Использовать уровни';

app.CheckBox\_levels\_on.Position = [23 110 141 22];

% Create Button\_Apply

app.Button\_Apply = uibutton(app.UIFigure, 'push');

app.Button\_Apply.ButtonPushedFcn = createCallbackFcn(app, @Button\_ApplyPushed, true);

app.Button\_Apply.Position = [544 14 100 22];

app.Button\_Apply.Text = 'Применить';

% Create Label

app.Label = uilabel(app.UIFigure);

app.Label.HorizontalAlignment = 'right';

app.Label.Position = [19 680 39 22];

app.Label.Text = 'Канал';

% Create color\_channel

app.color\_channel = uispinner(app.UIFigure);

app.color\_channel.Limits = [1 3];

app.color\_channel.Position = [73 680 41 22];

app.color\_channel.Value = 1;

% Create SpinnerLabel

app.SpinnerLabel = uilabel(app.UIFigure);

app.SpinnerLabel.HorizontalAlignment = 'right';

app.SpinnerLabel.Position = [42 78 58 22];

app.SpinnerLabel.Text = 'Минимум';

% Create Spinner\_levels\_min

app.Spinner\_levels\_min = uispinner(app.UIFigure);

app.Spinner\_levels\_min.Position = [115 78 100 22];

% Create Spinner\_2Label

app.Spinner\_2Label = uilabel(app.UIFigure);

app.Spinner\_2Label.HorizontalAlignment = 'right';

app.Spinner\_2Label.Position = [37 47 63 22];

app.Spinner\_2Label.Text = 'Максимум';

% Create Spinner\_levels\_max

app.Spinner\_levels\_max = uispinner(app.UIFigure);

app.Spinner\_levels\_max.Position = [115 47 100 22];

% Create Button\_cancel

app.Button\_cancel = uibutton(app.UIFigure, 'push');

app.Button\_cancel.ButtonPushedFcn = createCallbackFcn(app, @Button\_cancelPushed, true);

app.Button\_cancel.Position = [427 14 100 22];

app.Button\_cancel.Text = 'Отмена';

% Create UIAxes\_bg

app.UIAxes\_bg = uiaxes(app.UIFigure);

title(app.UIAxes\_bg, 'Фон')

app.UIAxes\_bg.Position = [359 507 285 203];

% Create UIAxes\_mask

app.UIAxes\_mask = uiaxes(app.UIFigure);

title(app.UIAxes\_mask, 'Маска')

app.UIAxes\_mask.Position = [360 302 284 203];

% Create ButtonReset

app.ButtonReset = uibutton(app.UIFigure, 'push');

app.ButtonReset.ButtonPushedFcn = createCallbackFcn(app, @ButtonResetButtonPushed, true);

app.ButtonReset.Position = [318 14 100 22];

app.ButtonReset.Text = 'Сбросить';

% Create Panel

app.Panel = uipanel(app.UIFigure);

app.Panel.Title = 'Вычисление фона';

app.Panel.Position = [21 393 274 125];

% Create Button\_bg\_calc

app.Button\_bg\_calc = uibutton(app.Panel, 'push');

app.Button\_bg\_calc.ButtonPushedFcn = createCallbackFcn(app, @Button\_bg\_calcPushed, true);

app.Button\_bg\_calc.Position = [102 7 160 22];

app.Button\_bg\_calc.Text = 'Вычислить фон';

% Create EditField\_2Label

app.EditField\_2Label = uilabel(app.Panel);

app.EditField\_2Label.HorizontalAlignment = 'right';

app.EditField\_2Label.Position = [70 68 73 22];

app.EditField\_2Label.Text = 'Перцентиль';

% Create EditField\_bg\_calc\_prctile

app.EditField\_bg\_calc\_prctile = uieditfield(app.Panel, 'numeric');

app.EditField\_bg\_calc\_prctile.Limits = [0 100];

app.EditField\_bg\_calc\_prctile.ValueDisplayFormat = '%.0f';

app.EditField\_bg\_calc\_prctile.Position = [158 68 100 22];

app.EditField\_bg\_calc\_prctile.Value = 5;

% Create Label\_3

app.Label\_3 = uilabel(app.Panel);

app.Label\_3.HorizontalAlignment = 'right';

app.Label\_3.Position = [61 39 82 22];

app.Label\_3.Text = 'Число кадров';

% Create EditField\_bg\_calc\_N

app.EditField\_bg\_calc\_N = uieditfield(app.Panel, 'numeric');

app.EditField\_bg\_calc\_N.Limits = [1 Inf];

app.EditField\_bg\_calc\_N.ValueDisplayFormat = '%.0f';

app.EditField\_bg\_calc\_N.Position = [158 39 100 22];

app.EditField\_bg\_calc\_N.Value = 100;

% Create Panel\_2

app.Panel\_2 = uipanel(app.UIFigure);

app.Panel\_2.Title = 'Загрузить фон';

app.Panel\_2.Position = [19 528 276 99];

% Create Button\_bg\_path

app.Button\_bg\_path = uibutton(app.Panel\_2, 'push');

app.Button\_bg\_path.ButtonPushedFcn = createCallbackFcn(app, @Button\_bg\_pathPushed, true);

app.Button\_bg\_path.Position = [209 43 56 22];

app.Button\_bg\_path.Text = 'Указать';

% Create Button\_bg\_load

app.Button\_bg\_load = uibutton(app.Panel\_2, 'push');

app.Button\_bg\_load.ButtonPushedFcn = createCallbackFcn(app, @Button\_bg\_loadPushed, true);

app.Button\_bg\_load.Position = [104 10 161 22];

app.Button\_bg\_load.Text = 'Загрузить фон из файла';

% Create EditFieldLabel

app.EditFieldLabel = uilabel(app.Panel\_2);

app.EditFieldLabel.HorizontalAlignment = 'right';

app.EditFieldLabel.Position = [7 43 80 22];

app.EditFieldLabel.Text = 'Путь к файлу';

% Create EditField\_bg\_path

app.EditField\_bg\_path = uieditfield(app.Panel\_2, 'text');

app.EditField\_bg\_path.Position = [102 43 100 22];

% Create Panel\_3

app.Panel\_3 = uipanel(app.UIFigure);

app.Panel\_3.Title = 'Загрузить маску';

app.Panel\_3.Position = [21 156 274 164];

% Create Button\_mask\_path

app.Button\_mask\_path = uibutton(app.Panel\_3, 'push');

app.Button\_mask\_path.ButtonPushedFcn = createCallbackFcn(app, @Button\_mask\_pathPushed, true);

app.Button\_mask\_path.Position = [209 111 56 22];

app.Button\_mask\_path.Text = 'Указать';

% Create EditField\_4Label

app.EditField\_4Label = uilabel(app.Panel\_3);

app.EditField\_4Label.HorizontalAlignment = 'right';

app.EditField\_4Label.Position = [6 111 80 22];

app.EditField\_4Label.Text = 'Путь к файлу';

% Create EditField\_mask\_path

app.EditField\_mask\_path = uieditfield(app.Panel\_3, 'text');

app.EditField\_mask\_path.Position = [101 111 100 22];

% Create Label\_4

app.Label\_4 = uilabel(app.Panel\_3);

app.Label\_4.HorizontalAlignment = 'right';

app.Label\_4.Position = [1 75 122 22];

app.Label\_4.Text = {'Цвет маски в файле:'; ''};

% Create EditField\_mask\_source\_color1

app.EditField\_mask\_source\_color1 = uieditfield(app.Panel\_3, 'numeric');

app.EditField\_mask\_source\_color1.Limits = [0 Inf];

app.EditField\_mask\_source\_color1.ValueDisplayFormat = '%.0f';

app.EditField\_mask\_source\_color1.Position = [39 54 47 22];

app.EditField\_mask\_source\_color1.Value = 250;

% Create EditField\_mask\_source\_color2

app.EditField\_mask\_source\_color2 = uieditfield(app.Panel\_3, 'numeric');

app.EditField\_mask\_source\_color2.Limits = [0 Inf];

app.EditField\_mask\_source\_color2.ValueDisplayFormat = '%.0f';

app.EditField\_mask\_source\_color2.Position = [94 54 47 22];

app.EditField\_mask\_source\_color2.Value = 255;

% Create Button\_mask\_load

app.Button\_mask\_load = uibutton(app.Panel\_3, 'push');

app.Button\_mask\_load.ButtonPushedFcn = createCallbackFcn(app, @Button\_mask\_loadPushed, true);

app.Button\_mask\_load.Position = [101.5 18 162 22];

app.Button\_mask\_load.Text = 'Загрузить маску из файла';

% Create EditField\_mask\_source\_color3

app.EditField\_mask\_source\_color3 = uieditfield(app.Panel\_3, 'numeric');

app.EditField\_mask\_source\_color3.Limits = [0 Inf];

app.EditField\_mask\_source\_color3.ValueDisplayFormat = '%.0f';

app.EditField\_mask\_source\_color3.Position = [147 54 47 22];

app.EditField\_mask\_source\_color3.Value = 255;

% Create Button\_mask\_source\_color\_pick

app.Button\_mask\_source\_color\_pick = uibutton(app.Panel\_3, 'push');

app.Button\_mask\_source\_color\_pick.ButtonPushedFcn = createCallbackFcn(app, @Button\_mask\_source\_color\_pickPushed, true);

app.Button\_mask\_source\_color\_pick.Position = [203 54 64 22];

app.Button\_mask\_source\_color\_pick.Text = {'Указать'; ''};

% Create Label\_5

app.Label\_5 = uilabel(app.UIFigure);

app.Label\_5.HorizontalAlignment = 'right';

app.Label\_5.Position = [20 331 229 22];

app.Label\_5.Text = 'Заменять замаскированные пиксели на';

% Create EditField\_mask\_pick\_values\_interval\_3

app.EditField\_mask\_pick\_values\_interval\_3 = uieditfield(app.UIFigure, 'numeric');

app.EditField\_mask\_pick\_values\_interval\_3.Limits = [0 Inf];

app.EditField\_mask\_pick\_values\_interval\_3.ValueDisplayFormat = '%.0f';

app.EditField\_mask\_pick\_values\_interval\_3.Position = [264 331 47 22];

end

end

methods (Access = public)

% Construct app

function app = PIV\_db\_ipp(varargin)

% Create and configure components

createComponents(app)

% Register the app with App Designer

registerApp(app, app.UIFigure)

% Execute the startup function

runStartupFcn(app, @(app)startupFcn(app, varargin{:}))

if nargout == 0

clear app

end

end

% Code that executes before app deletion

function delete(app)

% Delete UIFigure when app is deleted

delete(app.UIFigure)

end

end

end

## PIV\_db\_piv\_frames.mlapp

classdef PIV\_db\_piv\_frames < matlab.apps.AppBase

% Properties that correspond to app components

properties (Access = public)

UIFigure matlab.ui.Figure

pf\_apply matlab.ui.control.Button

pf\_cancel matlab.ui.control.Button

Label matlab.ui.control.Label

pf\_frame\_start matlab.ui.control.NumericEditField

Label\_2 matlab.ui.control.Label

pf\_frame\_end matlab.ui.control.NumericEditField

NLabel matlab.ui.control.Label

pf\_frame\_skip matlab.ui.control.Spinner

NLabel\_2 matlab.ui.control.Label

pf\_frame\_step matlab.ui.control.Spinner

pf\_select\_all matlab.ui.control.Button

pf\_UIAxes matlab.ui.control.UIAxes

pf\_Label\_info matlab.ui.control.Label

pf\_Reset matlab.ui.control.Button

end

properties (Access = private)

m PIV\_main

parent\_app

pf\_backup PIV\_frames

end

methods (Access = private)

function Update\_pf(app)

frame\_start = app.pf\_frame\_start.Value;

frame\_end = app.pf\_frame\_end.Value;

frame\_skip = app.pf\_frame\_skip.Value;

frame\_step = app.pf\_frame\_step.Value;

try

pf = PIV\_frames(frame\_start, frame\_end, frame\_skip, frame\_step);

catch

errordlg('Некорректные параметры: ни одного кадра не выбрано','Ошибка');

return;

end

if isempty(pf.first\_frames)

errordlg('Некорректные параметры: ни одного кадра не выбрано','Ошибка');

return;

end

app.parent\_app.m.Set\_pf(pf);

app.parent\_app.m.Redraw\_fields();

app.parent\_app.init\_gui(); %чтобы обновились пределы для спинера поле

%построения

cla(app.pf\_UIAxes);

app.pf\_UIAxes.YTick = [];

app.pf\_UIAxes.XLim = [1 app.parent\_app.m.im\_count];

app.pf\_UIAxes.YLim = [-1 2];

z = 0 \* pf.first\_frames;

quiver(app.pf\_UIAxes, ...

pf.first\_frames,...

z,...

z + pf.frame\_step, ...

z + 1,...

0, 'k',...

'ShowArrowHead','off');

app.pf\_Label\_info.Text = sprintf('Всего пар кадров: %d',pf.frame\_count);

end

function Load\_values(app)

pf = app.parent\_app.m.p.pf;

app.pf\_frame\_start.Value = pf.frame\_start;

app.pf\_frame\_end.Value = pf.frame\_end;

app.pf\_frame\_skip.Value = pf.frame\_skip;

app.pf\_frame\_step.Value = pf.frame\_step;

end

end

methods (Access = private)

% Code that executes after component creation

function startupFcn(app, parent\_app)

app.parent\_app = parent\_app;

app.pf\_backup = app.parent\_app.m.p.pf;

app.Load\_values();

app.Update\_pf();

end

% Button pushed function: pf\_apply

function pf\_applyPushed(app, event)

app.Update\_pf();

end

% Button pushed function: pf\_cancel

function pf\_cancelPushed(app, event)

app.parent\_app.m.Set\_pf(app.pf\_backup);

app.parent\_app.init\_gui(); %чтобы обновились пределы для спинера поле

delete(app);

end

% Button pushed function: pf\_select\_all

function pf\_select\_allPushed(app, event)

app.pf\_frame\_start.Value = 1;

app.pf\_frame\_end.Value = app.parent\_app.m.im\_count;

end

% Button pushed function: pf\_Reset

function pf\_ResetButtonPushed(app, event)

pf\_frame\_start = 1;

pf\_frame\_end = app.parent\_app.m.im\_count;

pf\_frame\_skip = 1;

pf\_frame\_step = 1;

pf = PIV\_frames(pf\_frame\_start, pf\_frame\_end, pf\_frame\_skip, pf\_frame\_step);

app.parent\_app.m.Set\_pf(pf);

app.parent\_app.init\_gui(); %чтобы обновились пределы для спинера поле

app.Load\_values();

app.Update\_pf();

end

end

% App initialization and construction

methods (Access = private)

% Create UIFigure and components

function createComponents(app)

% Create UIFigure

app.UIFigure = uifigure;

app.UIFigure.Position = [100 100 643 340];

app.UIFigure.Name = 'Параметры кадров';

% Create pf\_apply

app.pf\_apply = uibutton(app.UIFigure, 'push');

app.pf\_apply.ButtonPushedFcn = createCallbackFcn(app, @pf\_applyPushed, true);

app.pf\_apply.Position = [512 12 100 22];

app.pf\_apply.Text = 'Применить';

% Create pf\_cancel

app.pf\_cancel = uibutton(app.UIFigure, 'push');

app.pf\_cancel.ButtonPushedFcn = createCallbackFcn(app, @pf\_cancelPushed, true);

app.pf\_cancel.Position = [397 12 100 22];

app.pf\_cancel.Text = 'Отмена';

% Create Label

app.Label = uilabel(app.UIFigure);

app.Label.HorizontalAlignment = 'right';

app.Label.Position = [95 301 104 22];

app.Label.Text = 'Начинать с кадра';

% Create pf\_frame\_start

app.pf\_frame\_start = uieditfield(app.UIFigure, 'numeric');

app.pf\_frame\_start.Position = [214 301 100 22];

% Create Label\_2

app.Label\_2 = uilabel(app.UIFigure);

app.Label\_2.HorizontalAlignment = 'right';

app.Label\_2.Position = [71 266 128 22];

app.Label\_2.Text = 'Заканчивать на кадре';

% Create pf\_frame\_end

app.pf\_frame\_end = uieditfield(app.UIFigure, 'numeric');

app.pf\_frame\_end.Position = [214 266 100 22];

% Create NLabel

app.NLabel = uilabel(app.UIFigure);

app.NLabel.HorizontalAlignment = 'right';

app.NLabel.Position = [15 231 184 22];

app.NLabel.Text = 'Обрабатывать каждый N-й кадр';

% Create pf\_frame\_skip

app.pf\_frame\_skip = uispinner(app.UIFigure);

app.pf\_frame\_skip.Position = [214 231 100 22];

% Create NLabel\_2

app.NLabel\_2 = uilabel(app.UIFigure);

app.NLabel\_2.HorizontalAlignment = 'right';

app.NLabel\_2.Position = [38 197 161 22];

app.NLabel\_2.Text = 'Сравнивать через N кадров';

% Create pf\_frame\_step

app.pf\_frame\_step = uispinner(app.UIFigure);

app.pf\_frame\_step.Position = [214 197 100 22];

% Create pf\_select\_all

app.pf\_select\_all = uibutton(app.UIFigure, 'push');

app.pf\_select\_all.ButtonPushedFcn = createCallbackFcn(app, @pf\_select\_allPushed, true);

app.pf\_select\_all.Position = [338 266 86 57];

app.pf\_select\_all.Text = 'Вся запись';

% Create pf\_UIAxes

app.pf\_UIAxes = uiaxes(app.UIFigure);

title(app.pf\_UIAxes, 'Выбранные кадры')

xlabel(app.pf\_UIAxes, 'Номер кадра')

app.pf\_UIAxes.Position = [14 89 612 89];

% Create pf\_Label\_info

app.pf\_Label\_info = uilabel(app.UIFigure);

app.pf\_Label\_info.Position = [14 68 588 22];

app.pf\_Label\_info.Text = 'Инфо';

% Create pf\_Reset

app.pf\_Reset = uibutton(app.UIFigure, 'push');

app.pf\_Reset.ButtonPushedFcn = createCallbackFcn(app, @pf\_ResetButtonPushed, true);

app.pf\_Reset.Position = [284 12 100 22];

app.pf\_Reset.Text = 'Сбросить';

end

end

methods (Access = public)

% Construct app

function app = PIV\_db\_piv\_frames(varargin)

% Create and configure components

createComponents(app)

% Register the app with App Designer

registerApp(app, app.UIFigure)

% Execute the startup function

runStartupFcn(app, @(app)startupFcn(app, varargin{:}))

if nargout == 0

clear app

end

end

% Code that executes before app deletion

function delete(app)

% Delete UIFigure when app is deleted

delete(app.UIFigure)

end

end

end

## PIV\_db\_piv\_grid.mlapp

classdef PIV\_db\_piv\_grid < matlab.apps.AppBase

% Properties that correspond to app components

properties (Access = public)

UIFigure matlab.ui.Figure

XLabel\_4 matlab.ui.control.Label

YLabel\_4 matlab.ui.control.Label

Button\_lims\_auto matlab.ui.control.Button

Button\_cancel matlab.ui.control.Button

Button\_ginput\_start matlab.ui.control.Button

Spinner\_mask\_prc matlab.ui.control.Spinner

cCheckBox\_mask matlab.ui.control.CheckBox

Reset matlab.ui.control.Button

Button\_overlap matlab.ui.control.Button

Spinner matlab.ui.control.Spinner

Label\_2 matlab.ui.control.Label

Spinner\_step\_x matlab.ui.control.Spinner

Label\_5 matlab.ui.control.Label

Spinner\_step\_y matlab.ui.control.Spinner

Label matlab.ui.control.Label

Spinner\_start\_x matlab.ui.control.Spinner

Label\_4 matlab.ui.control.Label

Spinner\_start\_y matlab.ui.control.Spinner

Label\_3 matlab.ui.control.Label

Spinner\_end\_x matlab.ui.control.Spinner

Label\_6 matlab.ui.control.Label

Spinner\_end\_y matlab.ui.control.Spinner

Label\_overlap matlab.ui.control.Label

Button\_apply\_lims matlab.ui.control.Button

end

properties (Access = private)

parent\_app

pg\_backup PIV\_grid

end

methods (Access = private)

function Load\_values\_from\_parent\_app(app)

pg = app.parent\_app.m.p.pg;

app.Spinner\_start\_x.Value = pg.xArr(1);

app.Spinner\_start\_y.Value = pg.yArr(1);

app.Spinner\_end\_x.Value = pg.xArr(end);

app.Spinner\_end\_y.Value = pg.yArr(end);

if ~isnan(pg.xStep)

app.Spinner\_step\_x.Value = pg.xStep;

end

if ~isnan(pg.yStep)

app.Spinner\_step\_y.Value = pg.yStep;

end

overlap\_xy = PIV\_grid.Get\_overlap(...

app.parent\_app.m.p.pp.wSize,...

[pg.xStep, pg.yStep]);

app.Label\_overlap.Text = sprintf(...

'Текущее перекрытие: %.0f%% по x, %.0f%% по y',...

overlap\_xy(1), overlap\_xy(2));

end

function Update\_m(app)

s\_x = app.Spinner\_start\_x.Value;

s\_y = app.Spinner\_start\_y.Value;

st\_x = app.Spinner\_step\_x.Value;

st\_y = app.Spinner\_step\_y.Value;

e\_x = app.Spinner\_end\_x.Value;

e\_y = app.Spinner\_end\_y.Value;

x\_arr = s\_x:st\_x:e\_x;

y\_arr = s\_y:st\_y:e\_y;

if isempty(x\_arr) || isempty(y\_arr)

errordlg('Некорректные параметры сетки: нет узлов','Ошибка');

return;

end

pg = PIV\_grid(x\_arr,y\_arr);

if app.cCheckBox\_mask.Value

mask = app.parent\_app.m.p.ipp.mask;

if ~isempty(mask)

pg = pg.Disable\_from\_mask(...

mask,...

app.parent\_app.m.p.pp.wSize,...

app.Spinner\_mask\_prc.Value);

end

end

app.parent\_app.m.Set\_pg(pg);

app.parent\_app.m.Redraw\_fields();

app.Load\_values\_from\_parent\_app();

end

end

methods (Access = private)

% Code that executes after component creation

function startupFcn(app, parent\_app)

% addpath('D:\Кандауров\Программы\Matlab\\_path');

app.parent\_app = parent\_app;

app.pg\_backup = app.parent\_app.m.p.pg;

app.Load\_values\_from\_parent\_app();

end

% Callback function

function Button\_apply\_evalPushed(app, event)

t\_x = app.evalXEditField.Value;

t\_y = app.evalYEditField.Value;

im\_size\_x = app.parent\_app.m.im\_size(1);

im\_size\_y = app.parent\_app.m.im\_size(2);

w\_size\_x = app.parent\_app.m.p.pp.wSize(1);

w\_size\_y = app.parent\_app.m.p.pp.wSize(2);

eval (['x\_arr = ' t\_x]);

eval (['y\_arr = ' t\_y]);

app.parent\_app.m.Add\_to\_log(sprintf('x %d ',x\_arr));

app.parent\_app.m.Add\_to\_log(sprintf('y %d ',y\_arr));

pg = PIV\_grid(x\_arr,y\_arr);

app.parent\_app.m.Set\_pg(pg);

end

% Button pushed function: Button\_lims\_auto

function Button\_lims\_autoPushed(app, event)

im\_size = app.parent\_app.m.im\_size;

step\_x\_y = [app.Spinner\_step\_x.Value, app.Spinner\_step\_y.Value];

pg = PIV\_grid.Get\_max\_grid(im\_size, abs(step\_x\_y));

app.parent\_app.m.Set\_pg(pg);

app.Load\_values\_from\_parent\_app();

end

% Button pushed function: Button\_apply\_lims

function Button\_apply\_limsPushed(app, event)

app.Update\_m();

end

% Button pushed function: Button\_cancel

function Button\_cancelPushed(app, event)

app.parent\_app.m.Set\_pg(app.pg\_backup);

delete(app);

end

% Button pushed function: Button\_ginput\_start

function Button\_ginput\_startPushed(app, event)

p = app.parent\_app.m.selected\_points;

if size(p,2) < 2, return; end

p1 = p(end-1, :);

p2 = p(end , :);

sx = min( p1(1), p2(1) );

sy = min( p1(2), p2(2) );

ex = max( p1(1), p2(1) );

ey = max( p1(2), p2(2) );

app.Spinner\_start\_x.Value = round(sx);

app.Spinner\_start\_y.Value = round(sy);

app.Spinner\_end\_x.Value = round(ex);

app.Spinner\_end\_y.Value = round(ey);

app.Update\_m();

end

% Button pushed function: Reset

function ResetButtonPushed(app, event)

im\_size = app.parent\_app.m.im\_size;

step\_x\_y = round(app.parent\_app.m.p.pp.wSize/2);

pg = PIV\_grid.Get\_max\_grid(im\_size, abs(step\_x\_y));

app.parent\_app.m.Set\_pg(pg);

app.Load\_values\_from\_parent\_app();

end

% Callback function

function Reset\_2ButtonPushed(app, event)

app.evalXEditField.Value = 'ceil( w\_size\_x/2 + 1 ) : w\_size\_x/2 : floor( im\_size\_x - w\_size\_x/2 + 1 );';

app.evalYEditField.Value = 'ceil( w\_size\_y/2 + 1 ) : w\_size\_y/2 : floor( im\_size\_y - w\_size\_y/2 + 1 );';

end

% Button pushed function: Button\_overlap

function Button\_overlapPushed(app, event)

step\_xy = round ( PIV\_grid.Get\_step\_from\_overlap(...

app.parent\_app.m.p.pp.wSize,...

app.Spinner.Value) );

if ~isnan(step\_xy(1)) && step\_xy(1) > 0

app.Spinner\_step\_x.Value = step\_xy(1);

end

if ~isnan(step\_xy(2)) && step\_xy(2) > 0

app.Spinner\_step\_y.Value = step\_xy(2);

end

end

end

% App initialization and construction

methods (Access = private)

% Create UIFigure and components

function createComponents(app)

% Create UIFigure

app.UIFigure = uifigure;

app.UIFigure.Position = [100 100 467 418];

app.UIFigure.Name = 'Параметры сетки';

% Create XLabel\_4

app.XLabel\_4 = uilabel(app.UIFigure);

app.XLabel\_4.Position = [18 381 139 22];

app.XLabel\_4.Text = 'Координаты узлов по X:';

% Create YLabel\_4

app.YLabel\_4 = uilabel(app.UIFigure);

app.YLabel\_4.Position = [239 381 139 22];

app.YLabel\_4.Text = 'Координаты узлов по Y:';

% Create Button\_lims\_auto

app.Button\_lims\_auto = uibutton(app.UIFigure, 'push');

app.Button\_lims\_auto.ButtonPushedFcn = createCallbackFcn(app, @Button\_lims\_autoPushed, true);

app.Button\_lims\_auto.Position = [17 260 380 22];

app.Button\_lims\_auto.Text = 'Выбрать пределы автоматически';

% Create Button\_cancel

app.Button\_cancel = uibutton(app.UIFigure, 'push');

app.Button\_cancel.ButtonPushedFcn = createCallbackFcn(app, @Button\_cancelPushed, true);

app.Button\_cancel.Position = [229 38 100 22];

app.Button\_cancel.Text = 'Отмена';

% Create Button\_ginput\_start

app.Button\_ginput\_start = uibutton(app.UIFigure, 'push');

app.Button\_ginput\_start.ButtonPushedFcn = createCallbackFcn(app, @Button\_ginput\_startPushed, true);

app.Button\_ginput\_start.Position = [18 192 379 51];

app.Button\_ginput\_start.Text = {'Использовать две последние выбранные'; 'точки как углы области'};

% Create Spinner\_mask\_prc

app.Spinner\_mask\_prc = uispinner(app.UIFigure);

app.Spinner\_mask\_prc.Limits = [0 100];

app.Spinner\_mask\_prc.Position = [342 91 55 22];

app.Spinner\_mask\_prc.Value = 50;

% Create cCheckBox\_mask

app.cCheckBox\_mask = uicheckbox(app.UIFigure);

app.cCheckBox\_mask.Text = 'Убрать узлы сетки c % пикселей маски в окне больше';

app.cCheckBox\_mask.Position = [17 91 328 22];

% Create Reset

app.Reset = uibutton(app.UIFigure, 'push');

app.Reset.ButtonPushedFcn = createCallbackFcn(app, @ResetButtonPushed, true);

app.Reset.Position = [110 38 100 22];

app.Reset.Text = 'Сбросить';

% Create Button\_overlap

app.Button\_overlap = uibutton(app.UIFigure, 'push');

app.Button\_overlap.ButtonPushedFcn = createCallbackFcn(app, @Button\_overlapPushed, true);

app.Button\_overlap.Position = [18 129 187 22];

app.Button\_overlap.Text = 'Использовать перекрытие (%):';

% Create Spinner

app.Spinner = uispinner(app.UIFigure);

app.Spinner.Position = [297 129 100 22];

app.Spinner.Value = 50;

% Create Label\_2

app.Label\_2 = uilabel(app.UIFigure);

app.Label\_2.HorizontalAlignment = 'right';

app.Label\_2.Position = [16 323 26 22];

app.Label\_2.Text = 'шаг';

% Create Spinner\_step\_x

app.Spinner\_step\_x = uispinner(app.UIFigure);

app.Spinner\_step\_x.Position = [57 323 100 22];

app.Spinner\_step\_x.Value = 32;

% Create Label\_5

app.Label\_5 = uilabel(app.UIFigure);

app.Label\_5.HorizontalAlignment = 'right';

app.Label\_5.Position = [237 323 26 22];

app.Label\_5.Text = 'шаг';

% Create Spinner\_step\_y

app.Spinner\_step\_y = uispinner(app.UIFigure);

app.Spinner\_step\_y.Position = [278 323 100 22];

app.Spinner\_step\_y.Value = 32;

% Create Label

app.Label = uilabel(app.UIFigure);

app.Label.HorizontalAlignment = 'right';

app.Label.Position = [17 352 25 22];

app.Label.Text = 'от';

% Create Spinner\_start\_x

app.Spinner\_start\_x = uispinner(app.UIFigure);

app.Spinner\_start\_x.Position = [57 352 100 22];

% Create Label\_4

app.Label\_4 = uilabel(app.UIFigure);

app.Label\_4.HorizontalAlignment = 'right';

app.Label\_4.Position = [238 352 25 22];

app.Label\_4.Text = 'от';

% Create Spinner\_start\_y

app.Spinner\_start\_y = uispinner(app.UIFigure);

app.Spinner\_start\_y.Position = [278 352 100 22];

% Create Label\_3

app.Label\_3 = uilabel(app.UIFigure);

app.Label\_3.HorizontalAlignment = 'right';

app.Label\_3.Position = [17 294 25 22];

app.Label\_3.Text = 'до';

% Create Spinner\_end\_x

app.Spinner\_end\_x = uispinner(app.UIFigure);

app.Spinner\_end\_x.Position = [57 294 100 22];

app.Spinner\_end\_x.Value = 500;

% Create Label\_6

app.Label\_6 = uilabel(app.UIFigure);

app.Label\_6.HorizontalAlignment = 'right';

app.Label\_6.Position = [238 294 25 22];

app.Label\_6.Text = 'до';

% Create Spinner\_end\_y

app.Spinner\_end\_y = uispinner(app.UIFigure);

app.Spinner\_end\_y.Position = [278 294 100 22];

app.Spinner\_end\_y.Value = 500;

% Create Label\_overlap

app.Label\_overlap = uilabel(app.UIFigure);

app.Label\_overlap.Position = [22 161 611 22];

app.Label\_overlap.Text = 'Текущее перекрытие:';

% Create Button\_apply\_lims

app.Button\_apply\_lims = uibutton(app.UIFigure, 'push');

app.Button\_apply\_lims.ButtonPushedFcn = createCallbackFcn(app, @Button\_apply\_limsPushed, true);

app.Button\_apply\_lims.Position = [344 38 100 22];

app.Button\_apply\_lims.Text = 'Применить';

end

end

methods (Access = public)

% Construct app

function app = PIV\_db\_piv\_grid(varargin)

% Create and configure components

createComponents(app)

% Register the app with App Designer

registerApp(app, app.UIFigure)

% Execute the startup function

runStartupFcn(app, @(app)startupFcn(app, varargin{:}))

if nargout == 0

clear app

end

end

% Code that executes before app deletion

function delete(app)

% Delete UIFigure when app is deleted

delete(app.UIFigure)

end

end

end

## PIV\_db\_piv\_params.mlapp

classdef PIV\_db\_piv\_params < matlab.apps.AppBase

% Properties that correspond to app components

properties (Access = public)

PIVUIFigure matlab.ui.Figure

Button\_ok matlab.ui.control.Button

Label matlab.ui.control.Label

EditField\_wSize\_x matlab.ui.control.NumericEditField

EditField\_wSize\_y matlab.ui.control.NumericEditField

Button\_cancel matlab.ui.control.Button

Label\_2 matlab.ui.control.Label

EditField\_maxDispl\_x matlab.ui.control.NumericEditField

EditField\_maxDispl\_y matlab.ui.control.NumericEditField

Label\_3 matlab.ui.control.Label

EditField\_minS2NRatio matlab.ui.control.NumericEditField

CheckBox\_useSubpixel matlab.ui.control.CheckBox

CheckBox\_doMultiMax matlab.ui.control.CheckBox

CheckBox\_saveCCandIMC matlab.ui.control.CheckBox

Reset matlab.ui.control.Button

Label\_4 matlab.ui.control.Label

wSize1x matlab.ui.control.NumericEditField

wSize1y matlab.ui.control.NumericEditField

doFirstPass matlab.ui.control.CheckBox

end

properties (Access = private)

parent\_app

pp\_backup PIV\_params

end

methods (Access = private)

function update\_pp(app)

pp = app.parent\_app.m.p.pp;

pp.doFirstPass = app.doFirstPass.Value;

pp.wSize1(1) = app.wSize1x.Value;

pp.wSize1(2) = app.wSize1y.Value;

pp.wSize(1) = app.EditField\_wSize\_x.Value;

pp.wSize(2) = app.EditField\_wSize\_y.Value;

pp.maxDispl(1) = app.EditField\_maxDispl\_x.Value;

pp.maxDispl(2) = app.EditField\_maxDispl\_y.Value;

pp.minS2NRatio = app.EditField\_minS2NRatio.Value;

pp.useSubpixel = app.CheckBox\_useSubpixel.Value;

pp.doMultiMax = app.CheckBox\_doMultiMax.Value;

pp.saveCCandIMC = app.CheckBox\_saveCCandIMC.Value;

app.parent\_app.m.Set\_pp(pp);

end

function Load\_values(app)

pp = app.parent\_app.m.p.pp;

app.doFirstPass.Value = pp.doFirstPass;

app.wSize1x.Value = pp.wSize1(1);

app.wSize1y.Value = pp.wSize1(2);

app.EditField\_wSize\_x.Value = pp.wSize(1);

app.EditField\_wSize\_y.Value = pp.wSize(2);

app.EditField\_maxDispl\_x.Value = pp.maxDispl(1);

app.EditField\_maxDispl\_y.Value = pp.maxDispl(2);

app.EditField\_minS2NRatio.Value = pp.minS2NRatio;

app.CheckBox\_useSubpixel.Value = pp.useSubpixel;

app.CheckBox\_doMultiMax.Value = pp.doMultiMax;

app.CheckBox\_saveCCandIMC.Value = pp.saveCCandIMC;

end

end

methods (Access = private)

% Code that executes after component creation

function startupFcn(app, parent\_app)

app.parent\_app = parent\_app;

app.pp\_backup = app.parent\_app.m.p.pp;

app.Load\_values();

end

% Button pushed function: Button\_ok

function Button\_okPushed(app, event)

app.update\_pp();

end

% Button pushed function: Button\_cancel

function Button\_cancelPushed(app, event)

app.parent\_app.m.Set\_pp(app.pp\_backup);

delete(app);

end

% Button pushed function: Reset

function ResetButtonPushed(app, event)

pp = PIV\_params();

app.parent\_app.m.Set\_pp(pp);

app.Load\_values();

end

end

% App initialization and construction

methods (Access = private)

% Create UIFigure and components

function createComponents(app)

% Create PIVUIFigure

app.PIVUIFigure = uifigure;

app.PIVUIFigure.Position = [100 100 594 302];

app.PIVUIFigure.Name = 'Параметры PIV';

% Create Button\_ok

app.Button\_ok = uibutton(app.PIVUIFigure, 'push');

app.Button\_ok.ButtonPushedFcn = createCallbackFcn(app, @Button\_okPushed, true);

app.Button\_ok.Position = [427 30 100 22];

app.Button\_ok.Text = 'Применить';

% Create Label

app.Label = uilabel(app.PIVUIFigure);

app.Label.HorizontalAlignment = 'right';

app.Label.Position = [176 229 75 22];

app.Label.Text = 'Размер окна';

% Create EditField\_wSize\_x

app.EditField\_wSize\_x = uieditfield(app.PIVUIFigure, 'numeric');

app.EditField\_wSize\_x.Limits = [1 Inf];

app.EditField\_wSize\_x.Position = [266 229 38 22];

app.EditField\_wSize\_x.Value = 1;

% Create EditField\_wSize\_y

app.EditField\_wSize\_y = uieditfield(app.PIVUIFigure, 'numeric');

app.EditField\_wSize\_y.Limits = [1 Inf];

app.EditField\_wSize\_y.Position = [316 229 38 22];

app.EditField\_wSize\_y.Value = 1;

% Create Button\_cancel

app.Button\_cancel = uibutton(app.PIVUIFigure, 'push');

app.Button\_cancel.ButtonPushedFcn = createCallbackFcn(app, @Button\_cancelPushed, true);

app.Button\_cancel.Position = [313 30 100 22];

app.Button\_cancel.Text = 'Отмена';

% Create Label\_2

app.Label\_2 = uilabel(app.PIVUIFigure);

app.Label\_2.HorizontalAlignment = 'right';

app.Label\_2.Position = [102 199 149 22];

app.Label\_2.Text = 'Максимальное смещение';

% Create EditField\_maxDispl\_x

app.EditField\_maxDispl\_x = uieditfield(app.PIVUIFigure, 'numeric');

app.EditField\_maxDispl\_x.Position = [266 199 38 22];

% Create EditField\_maxDispl\_y

app.EditField\_maxDispl\_y = uieditfield(app.PIVUIFigure, 'numeric');

app.EditField\_maxDispl\_y.Position = [316 200 38 22];

% Create Label\_3

app.Label\_3 = uilabel(app.PIVUIFigure);

app.Label\_3.HorizontalAlignment = 'right';

app.Label\_3.Position = [20 168 231 22];

app.Label\_3.Text = 'Минимальное соотношение сигнал шум';

% Create EditField\_minS2NRatio

app.EditField\_minS2NRatio = uieditfield(app.PIVUIFigure, 'numeric');

app.EditField\_minS2NRatio.Position = [266 168 38 22];

app.EditField\_minS2NRatio.Value = 1;

% Create CheckBox\_useSubpixel

app.CheckBox\_useSubpixel = uicheckbox(app.PIVUIFigure);

app.CheckBox\_useSubpixel.Text = 'Субпиксельная точность';

app.CheckBox\_useSubpixel.Position = [266 135 160 22];

% Create CheckBox\_doMultiMax

app.CheckBox\_doMultiMax = uicheckbox(app.PIVUIFigure);

app.CheckBox\_doMultiMax.Text = 'Поиск нескольких максимумов';

app.CheckBox\_doMultiMax.Position = [266 105 194 22];

% Create CheckBox\_saveCCandIMC

app.CheckBox\_saveCCandIMC = uicheckbox(app.PIVUIFigure);

app.CheckBox\_saveCCandIMC.Text = 'Сохранять дополнительные данные';

app.CheckBox\_saveCCandIMC.Position = [266 76 226 22];

% Create Reset

app.Reset = uibutton(app.PIVUIFigure, 'push');

app.Reset.ButtonPushedFcn = createCallbackFcn(app, @ResetButtonPushed, true);

app.Reset.Position = [204 30 100 22];

app.Reset.Text = 'Сбросить';

% Create Label\_4

app.Label\_4 = uilabel(app.PIVUIFigure);

app.Label\_4.HorizontalAlignment = 'right';

app.Label\_4.Position = [66 260 185 22];

app.Label\_4.Text = 'Размер окна на первом проходе';

% Create wSize1x

app.wSize1x = uieditfield(app.PIVUIFigure, 'numeric');

app.wSize1x.Limits = [1 Inf];

app.wSize1x.Position = [266 260 38 22];

app.wSize1x.Value = 1;

% Create wSize1y

app.wSize1y = uieditfield(app.PIVUIFigure, 'numeric');

app.wSize1y.Limits = [1 Inf];

app.wSize1y.Position = [316 260 38 22];

app.wSize1y.Value = 1;

% Create doFirstPass

app.doFirstPass = uicheckbox(app.PIVUIFigure);

app.doFirstPass.Text = 'Использовать первый проход';

app.doFirstPass.Position = [370 260 186 22];

end

end

methods (Access = public)

% Construct app

function app = PIV\_db\_piv\_params(varargin)

% Create and configure components

createComponents(app)

% Register the app with App Designer

registerApp(app, app.PIVUIFigure)

% Execute the startup function

runStartupFcn(app, @(app)startupFcn(app, varargin{:}))

if nargout == 0

clear app

end

end

% Code that executes before app deletion

function delete(app)

% Delete UIFigure when app is deleted

delete(app.PIVUIFigure)

end

end

end

## PIV\_db\_plot.mlapp

classdef PIV\_db\_plot < matlab.apps.AppBase

% Properties that correspond to app components

properties (Access = public)

UIFigure matlab.ui.Figure

CheckBox\_imon matlab.ui.control.CheckBox

CheckBox\_gridon matlab.ui.control.CheckBox

CheckBox\_quiver\_on\_piv matlab.ui.control.CheckBox

CheckBox\_pcoloron matlab.ui.control.CheckBox

DropDown\_pcolor\_value matlab.ui.control.DropDown

ButtonGroup\_im\_select matlab.ui.container.ButtonGroup

Button\_\_im\_select\_initial matlab.ui.control.RadioButton

Button\_\_im\_select\_processed matlab.ui.control.RadioButton

CheckBox\_\_quiver\_dir\_only matlab.ui.control.CheckBox

CheckBox\_pcolor\_caxis\_auto matlab.ui.control.CheckBox

EditFieldLabel matlab.ui.control.Label

EditField\_grid\_s matlab.ui.control.EditField

Spinner\_3Label matlab.ui.control.Label

Spinner\_grid\_ssize matlab.ui.control.Spinner

ColormapEditFieldLabel matlab.ui.control.Label

ColormapEditField\_im\_cm matlab.ui.control.EditField

EditField\_2Label matlab.ui.control.Label

EditField\_q\_scale matlab.ui.control.NumericEditField

EditField\_4Label matlab.ui.control.Label

EditField\_p\_cl\_1 matlab.ui.control.NumericEditField

EditField\_5Label matlab.ui.control.Label

EditField\_p\_cl\_2 matlab.ui.control.NumericEditField

Button\_apply matlab.ui.control.Button

Button\_cancel matlab.ui.control.Button

EditField\_6Label matlab.ui.control.Label

EditField\_g\_c matlab.ui.control.EditField

CheckBox\_p\_cbon matlab.ui.control.CheckBox

EditField\_3Label\_2 matlab.ui.control.Label

EditField\_q\_c\_piv matlab.ui.control.EditField

EditField\_3Label\_3 matlab.ui.control.Label

EditField\_q\_c\_fi matlab.ui.control.EditField

ShadingDropDownLabel matlab.ui.control.Label

ShadingDropDown\_pcolor\_shading matlab.ui.control.DropDown

ColormapLabel matlab.ui.control.Label

ShadingDropDown\_pcolor\_colormap matlab.ui.control.DropDown

Label matlab.ui.control.Label

Spinner\_quiver\_step matlab.ui.control.Spinner

Label\_2 matlab.ui.control.Label

ShadingDropDown\_pcolor\_source matlab.ui.control.DropDown

CheckBox\_quiver\_on\_fi matlab.ui.control.CheckBox

grid\_show\_status matlab.ui.control.CheckBox

grid\_show\_status\_filtering matlab.ui.control.CheckBox

Button matlab.ui.control.Button

quiver\_on\_mean matlab.ui.control.CheckBox

EditField\_3Label\_4 matlab.ui.control.Label

quiver\_color\_mean matlab.ui.control.EditField

plot\_in\_scale matlab.ui.control.CheckBox

show\_grid matlab.ui.control.CheckBox

show\_ticks matlab.ui.control.CheckBox

show\_title matlab.ui.control.CheckBox

Label\_3 matlab.ui.control.Label

Label\_4 matlab.ui.control.Label

cb\_grid\_show\_windows matlab.ui.control.CheckBox

end

properties (Access = private)

parent\_app

plot\_params\_backup Plot\_params

end

methods (Access = private)

function Load\_values(app)

plot\_params = app.parent\_app.m.p.plot\_params;

app.CheckBox\_imon.Value = plot\_params.image\_on;

app.Button\_\_im\_select\_processed.Value = plot\_params.image\_processed;

app.ColormapEditField\_im\_cm.Value = plot\_params.image\_colormap;

app.CheckBox\_gridon.Value = plot\_params.grid\_on;

app.EditField\_grid\_s.Value = plot\_params.grid\_symbol;

app.Spinner\_grid\_ssize.Value = plot\_params.grid\_symbol\_size;

app.EditField\_g\_c.Value = plot\_params.grid\_symbol\_color;

app.grid\_show\_status.Value = plot\_params.grid\_show\_status;

app.grid\_show\_status\_filtering.Value = plot\_params.grid\_show\_status\_filtering;

app.cb\_grid\_show\_windows.Value = plot\_params.grid\_show\_windows;

app.CheckBox\_pcoloron.Value = plot\_params.pcolor\_on;

app.ShadingDropDown\_pcolor\_source.Value = plot\_params.pcolor\_source;

app.ShadingDropDown\_pcolor\_colormap.Value = plot\_params.pcolor\_colormap;

app.DropDown\_pcolor\_value.Value = plot\_params.pcolor\_value;

app.CheckBox\_pcolor\_caxis\_auto.Value = plot\_params.pcolor\_caxis\_auto;

app.EditField\_p\_cl\_1.Value = plot\_params.pcolor\_caxis(1);

app.EditField\_p\_cl\_2.Value = plot\_params.pcolor\_caxis(2);

app.CheckBox\_p\_cbon.Value = plot\_params.pcolor\_colorbar\_on;

app.ShadingDropDown\_pcolor\_shading.Value= plot\_params.pcolor\_shading;

app.CheckBox\_quiver\_on\_piv.Value = plot\_params.quiver\_on\_piv;

app.CheckBox\_quiver\_on\_fi.Value = plot\_params.quiver\_on\_fi;

app.EditField\_q\_c\_piv.Value = plot\_params.quiver\_color\_piv;

app.EditField\_q\_c\_fi.Value = plot\_params.quiver\_color\_fi;

app.EditField\_q\_scale.Value = plot\_params.quiver\_scale;

app.Spinner\_quiver\_step.Value = plot\_params.quiver\_step;

app.CheckBox\_\_quiver\_dir\_only.Value = plot\_params.quiver\_directions\_only;

app.quiver\_on\_mean.Value = plot\_params.quiver\_on\_mean;

app.quiver\_color\_mean.Value = plot\_params.quiver\_color\_mean;

app.plot\_in\_scale.Value = plot\_params.plot\_in\_scale;

app.show\_grid.Value = plot\_params.show\_grid;

app.show\_ticks.Value = plot\_params.show\_ticks;

app.show\_title.Value = plot\_params.show\_title;

end

end

methods (Access = private)

% Code that executes after component creation

function startupFcn(app, parent\_app)

app.parent\_app = parent\_app;

plot\_params = app.parent\_app.m.p.plot\_params;

app.plot\_params\_backup = plot\_params;

app.ShadingDropDown\_pcolor\_colormap.Items = {

'parula',...

'jet',...

'hsv',...

'hot',...

'cool',...

'spring',...

'summer',...

'autumn',...

'winter',...

'gray',...

'bone',...

'copper',...

'pink',...

'lines',...

'colorcube',...

'prism',...

'flag',...

'white'};

app.Load\_values();

end

% Button pushed function: Button\_cancel

function Button\_cancelPushed(app, event)

app.parent\_app.m.Set\_plot\_params( app.plot\_params\_backup );

delete(app);

end

% Button pushed function: Button\_apply

function Button\_applyPushed(app, event)

plot\_params = app.parent\_app.m.p.plot\_params;

plot\_params.image\_on= app.CheckBox\_imon.Value;

plot\_params.image\_processed= app.Button\_\_im\_select\_processed.Value;

plot\_params.image\_colormap=app.ColormapEditField\_im\_cm.Value;

plot\_params.grid\_on=app.CheckBox\_gridon.Value;

plot\_params.grid\_symbol=app.EditField\_grid\_s.Value;

plot\_params.grid\_symbol\_size=app.Spinner\_grid\_ssize.Value;

plot\_params.grid\_symbol\_color=app.EditField\_g\_c.Value;

plot\_params.grid\_show\_status = app.grid\_show\_status.Value;

plot\_params.grid\_show\_status\_filtering = app.grid\_show\_status\_filtering.Value;

plot\_params.grid\_show\_windows = app.cb\_grid\_show\_windows.Value;

plot\_params.pcolor\_on=app.CheckBox\_pcoloron.Value;

plot\_params.pcolor\_source = app.ShadingDropDown\_pcolor\_source.Value;

plot\_params.pcolor\_colormap=app.ShadingDropDown\_pcolor\_colormap.Value;

plot\_params.pcolor\_value=app.DropDown\_pcolor\_value.Value;

plot\_params.pcolor\_caxis\_auto=app.CheckBox\_pcolor\_caxis\_auto.Value;

plot\_params.pcolor\_caxis(1)=app.EditField\_p\_cl\_1.Value;

plot\_params.pcolor\_caxis(2)=app.EditField\_p\_cl\_2.Value;

plot\_params.pcolor\_colorbar\_on=app.CheckBox\_p\_cbon.Value;

plot\_params.pcolor\_shading = app.ShadingDropDown\_pcolor\_shading.Value;

plot\_params.quiver\_on\_piv = app.CheckBox\_quiver\_on\_piv.Value;

plot\_params.quiver\_on\_fi = app.CheckBox\_quiver\_on\_fi.Value;

plot\_params.quiver\_color\_piv = app.EditField\_q\_c\_piv.Value;

plot\_params.quiver\_color\_fi = app.EditField\_q\_c\_fi.Value;

plot\_params.quiver\_scale = app.EditField\_q\_scale.Value;

plot\_params.quiver\_step = app.Spinner\_quiver\_step.Value;

plot\_params.quiver\_directions\_only = app.CheckBox\_\_quiver\_dir\_only.Value;

plot\_params.quiver\_on\_mean = app.quiver\_on\_mean.Value;

plot\_params.quiver\_color\_mean = app.quiver\_color\_mean.Value;

plot\_params.plot\_in\_scale = app.plot\_in\_scale.Value;

plot\_params.show\_grid = app.show\_grid.Value;

plot\_params.show\_ticks = app.show\_ticks.Value;

plot\_params.show\_title = app.show\_title.Value;

app.parent\_app.m.Set\_plot\_params( plot\_params );

app.parent\_app.m.Redraw\_fields();

end

% Button pushed function: Button

function ButtonPushed(app, event)

plot\_params = Plot\_params();

app.parent\_app.m.Set\_plot\_params( plot\_params );

app.Load\_values();

end

end

% App initialization and construction

methods (Access = private)

% Create UIFigure and components

function createComponents(app)

% Create UIFigure

app.UIFigure = uifigure;

app.UIFigure.Position = [100 100 542 617];

app.UIFigure.Name = 'Параметры отображения';

% Create CheckBox\_imon

app.CheckBox\_imon = uicheckbox(app.UIFigure);

app.CheckBox\_imon.Text = 'Изображения';

app.CheckBox\_imon.Position = [23 580 97 22];

app.CheckBox\_imon.Value = true;

% Create CheckBox\_gridon

app.CheckBox\_gridon = uicheckbox(app.UIFigure);

app.CheckBox\_gridon.Text = 'PIV сетка';

app.CheckBox\_gridon.Position = [23 498 75 22];

% Create CheckBox\_quiver\_on\_piv

app.CheckBox\_quiver\_on\_piv = uicheckbox(app.UIFigure);

app.CheckBox\_quiver\_on\_piv.Text = 'Векторные поля исходные';

app.CheckBox\_quiver\_on\_piv.Position = [19 236 170 22];

% Create CheckBox\_pcoloron

app.CheckBox\_pcoloron = uicheckbox(app.UIFigure);

app.CheckBox\_pcoloron.Text = 'Скалярные значения скорости';

app.CheckBox\_pcoloron.Position = [23 391 194 22];

% Create DropDown\_pcolor\_value

app.DropDown\_pcolor\_value = uidropdown(app.UIFigure);

app.DropDown\_pcolor\_value.Items = {'u', 'v', 'V'};

app.DropDown\_pcolor\_value.Position = [146 360 51 22];

app.DropDown\_pcolor\_value.Value = 'u';

% Create ButtonGroup\_im\_select

app.ButtonGroup\_im\_select = uibuttongroup(app.UIFigure);

app.ButtonGroup\_im\_select.BorderType = 'none';

app.ButtonGroup\_im\_select.Position = [37 530 123 51];

% Create Button\_\_im\_select\_initial

app.Button\_\_im\_select\_initial = uiradiobutton(app.ButtonGroup\_im\_select);

app.Button\_\_im\_select\_initial.Text = 'Исходные';

app.Button\_\_im\_select\_initial.Position = [11 25 78 22];

% Create Button\_\_im\_select\_processed

app.Button\_\_im\_select\_processed = uiradiobutton(app.ButtonGroup\_im\_select);

app.Button\_\_im\_select\_processed.Text = 'После препроцессинга';

app.Button\_\_im\_select\_processed.Position = [11 3 150 22];

app.Button\_\_im\_select\_processed.Value = true;

% Create CheckBox\_\_quiver\_dir\_only

app.CheckBox\_\_quiver\_dir\_only = uicheckbox(app.UIFigure);

app.CheckBox\_\_quiver\_dir\_only.Text = 'Только направление';

app.CheckBox\_\_quiver\_dir\_only.Position = [168 162 136 22];

% Create CheckBox\_pcolor\_caxis\_auto

app.CheckBox\_pcolor\_caxis\_auto = uicheckbox(app.UIFigure);

app.CheckBox\_pcolor\_caxis\_auto.Text = 'Пределы автоматически';

app.CheckBox\_pcolor\_caxis\_auto.Position = [58 328 159 22];

% Create EditFieldLabel

app.EditFieldLabel = uilabel(app.UIFigure);

app.EditFieldLabel.HorizontalAlignment = 'right';

app.EditFieldLabel.Position = [47 467 49 22];

app.EditFieldLabel.Text = 'Символ';

% Create EditField\_grid\_s

app.EditField\_grid\_s = uieditfield(app.UIFigure, 'text');

app.EditField\_grid\_s.Position = [111 467 24 22];

app.EditField\_grid\_s.Value = '+';

% Create Spinner\_3Label

app.Spinner\_3Label = uilabel(app.UIFigure);

app.Spinner\_3Label.HorizontalAlignment = 'right';

app.Spinner\_3Label.Position = [146 467 97 22];

app.Spinner\_3Label.Text = 'Размер символа';

% Create Spinner\_grid\_ssize

app.Spinner\_grid\_ssize = uispinner(app.UIFigure);

app.Spinner\_grid\_ssize.Limits = [0 Inf];

app.Spinner\_grid\_ssize.Position = [258 467 100 22];

app.Spinner\_grid\_ssize.Value = 5;

% Create ColormapEditFieldLabel

app.ColormapEditFieldLabel = uilabel(app.UIFigure);

app.ColormapEditFieldLabel.HorizontalAlignment = 'right';

app.ColormapEditFieldLabel.Position = [209 554 58 22];

app.ColormapEditFieldLabel.Text = 'Colormap';

% Create ColormapEditField\_im\_cm

app.ColormapEditField\_im\_cm = uieditfield(app.UIFigure, 'text');

app.ColormapEditField\_im\_cm.Position = [282 554 38 22];

app.ColormapEditField\_im\_cm.Value = 'jet';

% Create EditField\_2Label

app.EditField\_2Label = uilabel(app.UIFigure);

app.EditField\_2Label.HorizontalAlignment = 'right';

app.EditField\_2Label.Position = [23 162 57 22];

app.EditField\_2Label.Text = 'Масштаб';

% Create EditField\_q\_scale

app.EditField\_q\_scale = uieditfield(app.UIFigure, 'numeric');

app.EditField\_q\_scale.Position = [95 162 48 22];

app.EditField\_q\_scale.Value = 10;

% Create EditField\_4Label

app.EditField\_4Label = uilabel(app.UIFigure);

app.EditField\_4Label.HorizontalAlignment = 'right';

app.EditField\_4Label.Position = [293 328 29 22];

app.EditField\_4Label.Text = 'Мин';

% Create EditField\_p\_cl\_1

app.EditField\_p\_cl\_1 = uieditfield(app.UIFigure, 'numeric');

app.EditField\_p\_cl\_1.Position = [337 328 30 22];

% Create EditField\_5Label

app.EditField\_5Label = uilabel(app.UIFigure);

app.EditField\_5Label.HorizontalAlignment = 'right';

app.EditField\_5Label.Position = [384 328 34 22];

app.EditField\_5Label.Text = 'Макс';

% Create EditField\_p\_cl\_2

app.EditField\_p\_cl\_2 = uieditfield(app.UIFigure, 'numeric');

app.EditField\_p\_cl\_2.Position = [433 328 31 22];

% Create Button\_apply

app.Button\_apply = uibutton(app.UIFigure, 'push');

app.Button\_apply.ButtonPushedFcn = createCallbackFcn(app, @Button\_applyPushed, true);

app.Button\_apply.Position = [422 19 100 22];

app.Button\_apply.Text = 'Применить';

% Create Button\_cancel

app.Button\_cancel = uibutton(app.UIFigure, 'push');

app.Button\_cancel.ButtonPushedFcn = createCallbackFcn(app, @Button\_cancelPushed, true);

app.Button\_cancel.Position = [313 19 100 22];

app.Button\_cancel.Text = 'Отмена';

% Create EditField\_6Label

app.EditField\_6Label = uilabel(app.UIFigure);

app.EditField\_6Label.HorizontalAlignment = 'right';

app.EditField\_6Label.Position = [372 467 32 22];

app.EditField\_6Label.Text = 'Цвет';

% Create EditField\_g\_c

app.EditField\_g\_c = uieditfield(app.UIFigure, 'text');

app.EditField\_g\_c.Position = [419 467 85 22];

app.EditField\_g\_c.Value = 'r';

% Create CheckBox\_p\_cbon

app.CheckBox\_p\_cbon = uicheckbox(app.UIFigure);

app.CheckBox\_p\_cbon.Text = 'Показывать сolorbar';

app.CheckBox\_p\_cbon.Position = [58 267 135 22];

% Create EditField\_3Label\_2

app.EditField\_3Label\_2 = uilabel(app.UIFigure);

app.EditField\_3Label\_2.HorizontalAlignment = 'right';

app.EditField\_3Label\_2.Position = [342 236 32 22];

app.EditField\_3Label\_2.Text = 'Цвет';

% Create EditField\_q\_c\_piv

app.EditField\_q\_c\_piv = uieditfield(app.UIFigure, 'text');

app.EditField\_q\_c\_piv.Position = [389 236 85 22];

app.EditField\_q\_c\_piv.Value = 'r';

% Create EditField\_3Label\_3

app.EditField\_3Label\_3 = uilabel(app.UIFigure);

app.EditField\_3Label\_3.HorizontalAlignment = 'right';

app.EditField\_3Label\_3.Position = [342 215 32 22];

app.EditField\_3Label\_3.Text = 'Цвет';

% Create EditField\_q\_c\_fi

app.EditField\_q\_c\_fi = uieditfield(app.UIFigure, 'text');

app.EditField\_q\_c\_fi.Position = [389 215 85 22];

app.EditField\_q\_c\_fi.Value = 'r';

% Create ShadingDropDownLabel

app.ShadingDropDownLabel = uilabel(app.UIFigure);

app.ShadingDropDownLabel.HorizontalAlignment = 'right';

app.ShadingDropDownLabel.Position = [227 297 50 22];

app.ShadingDropDownLabel.Text = 'Shading';

% Create ShadingDropDown\_pcolor\_shading

app.ShadingDropDown\_pcolor\_shading = uidropdown(app.UIFigure);

app.ShadingDropDown\_pcolor\_shading.Items = {'faceted', 'flat', 'interp'};

app.ShadingDropDown\_pcolor\_shading.Position = [292 297 100 22];

app.ShadingDropDown\_pcolor\_shading.Value = 'interp';

% Create ColormapLabel

app.ColormapLabel = uilabel(app.UIFigure);

app.ColormapLabel.HorizontalAlignment = 'right';

app.ColormapLabel.Position = [47 297 58 22];

app.ColormapLabel.Text = 'Colormap';

% Create ShadingDropDown\_pcolor\_colormap

app.ShadingDropDown\_pcolor\_colormap = uidropdown(app.UIFigure);

app.ShadingDropDown\_pcolor\_colormap.Items = {};

app.ShadingDropDown\_pcolor\_colormap.Position = [120 297 100 22];

app.ShadingDropDown\_pcolor\_colormap.Value = {};

% Create Label

app.Label = uilabel(app.UIFigure);

app.Label.HorizontalAlignment = 'right';

app.Label.Position = [320 162 88 22];

app.Label.Text = 'Прореживание';

% Create Spinner\_quiver\_step

app.Spinner\_quiver\_step = uispinner(app.UIFigure);

app.Spinner\_quiver\_step.Limits = [1 Inf];

app.Spinner\_quiver\_step.Position = [423 162 48 22];

app.Spinner\_quiver\_step.Value = 1;

% Create Label\_2

app.Label\_2 = uilabel(app.UIFigure);

app.Label\_2.HorizontalAlignment = 'right';

app.Label\_2.Position = [398 360 25 22];

app.Label\_2.Text = '';

% Create ShadingDropDown\_pcolor\_source

app.ShadingDropDown\_pcolor\_source = uidropdown(app.UIFigure);

app.ShadingDropDown\_pcolor\_source.Items = {'piv', 'fi', 'mean'};

app.ShadingDropDown\_pcolor\_source.Position = [372 360 100 22];

app.ShadingDropDown\_pcolor\_source.Value = 'piv';

% Create CheckBox\_quiver\_on\_fi

app.CheckBox\_quiver\_on\_fi = uicheckbox(app.UIFigure);

app.CheckBox\_quiver\_on\_fi.Text = 'Векторные поля после фильтрации и интерполяции';

app.CheckBox\_quiver\_on\_fi.Position = [19 215 316 22];

% Create grid\_show\_status

app.grid\_show\_status = uicheckbox(app.UIFigure);

app.grid\_show\_status.Text = 'Показывать статус';

app.grid\_show\_status.Position = [47 432 126 22];

% Create grid\_show\_status\_filtering

app.grid\_show\_status\_filtering = uicheckbox(app.UIFigure);

app.grid\_show\_status\_filtering.Text = 'Показывать статус фильтрации';

app.grid\_show\_status\_filtering.Position = [188 432 199 22];

% Create Button

app.Button = uibutton(app.UIFigure, 'push');

app.Button.ButtonPushedFcn = createCallbackFcn(app, @ButtonPushed, true);

app.Button.Position = [201 19 100 22];

app.Button.Text = 'Сбросить';

% Create quiver\_on\_mean

app.quiver\_on\_mean = uicheckbox(app.UIFigure);

app.quiver\_on\_mean.Text = 'Векторные поля: среднее';

app.quiver\_on\_mean.Position = [19 194 165 22];

% Create EditField\_3Label\_4

app.EditField\_3Label\_4 = uilabel(app.UIFigure);

app.EditField\_3Label\_4.HorizontalAlignment = 'right';

app.EditField\_3Label\_4.Position = [342 194 32 22];

app.EditField\_3Label\_4.Text = 'Цвет';

% Create quiver\_color\_mean

app.quiver\_color\_mean = uieditfield(app.UIFigure, 'text');

app.quiver\_color\_mean.Position = [389 194 85 22];

app.quiver\_color\_mean.Value = 'r';

% Create plot\_in\_scale

app.plot\_in\_scale = uicheckbox(app.UIFigure);

app.plot\_in\_scale.Text = 'Строить в размерных величинах';

app.plot\_in\_scale.Position = [19 112 205 22];

% Create show\_grid

app.show\_grid = uicheckbox(app.UIFigure);

app.show\_grid.Text = 'Показывать сетку';

app.show\_grid.Position = [19 92 120 22];

% Create show\_ticks

app.show\_ticks = uicheckbox(app.UIFigure);

app.show\_ticks.Text = 'Показывать метки на осях';

app.show\_ticks.Position = [19 72 168 22];

% Create show\_title

app.show\_title = uicheckbox(app.UIFigure);

app.show\_title.Text = 'Показывать подпись';

app.show\_title.Position = [19 52 137 22];

% Create Label\_3

app.Label\_3 = uilabel(app.UIFigure);

app.Label\_3.Position = [59 360 59 22];

app.Label\_3.Text = 'Значение';

% Create Label\_4

app.Label\_4 = uilabel(app.UIFigure);

app.Label\_4.Position = [282 360 57 22];

app.Label\_4.Text = 'Источник';

% Create cb\_grid\_show\_windows

app.cb\_grid\_show\_windows = uicheckbox(app.UIFigure);

app.cb\_grid\_show\_windows.Text = 'Окна';

app.cb\_grid\_show\_windows.Position = [407 432 50 22];

end

end

methods (Access = public)

% Construct app

function app = PIV\_db\_plot(varargin)

% Create and configure components

createComponents(app)

% Register the app with App Designer

registerApp(app, app.UIFigure)

% Execute the startup function

runStartupFcn(app, @(app)startupFcn(app, varargin{:}))

if nargout == 0

clear app

end

end

% Code that executes before app deletion

function delete(app)

% Delete UIFigure when app is deleted

delete(app.UIFigure)

end

end

end

## PIV\_db\_process.mlapp

classdef PIV\_db\_process < matlab.apps.AppBase

% Properties that correspond to app components

properties (Access = public)

UIFigure matlab.ui.Figure

Button\_start\_one matlab.ui.control.Button

Button\_start\_all matlab.ui.control.Button

Button\_stop matlab.ui.control.Button

UIAxes matlab.ui.control.UIAxes

Label\_info\_1 matlab.ui.control.Label

CheckBox\_plot matlab.ui.control.CheckBox

Label\_info\_2 matlab.ui.control.Label

CheckBox\_drawnow matlab.ui.control.CheckBox

Label matlab.ui.control.Label

Spinner\_drawnow\_framestep matlab.ui.control.Spinner

Label\_2 matlab.ui.control.Label

Spinner\_plot\_framestep matlab.ui.control.Spinner

Button\_start\_unprocessed matlab.ui.control.Button

Button\_start\_one\_2 matlab.ui.control.Button

Button\_fi matlab.ui.control.Button

Do\_fi\_after\_piv matlab.ui.control.CheckBox

autosave matlab.ui.control.CheckBox

Label\_3 matlab.ui.control.Label

autosave\_period matlab.ui.control.Spinner

end

properties (Access = private)

parent\_app

flag\_run\_PIV = false;

iteration\_time\_full = nan;

time\_left\_min = 0;

end

methods (Access = private)

function Update\_info(app)

app.parent\_app.m.ax\_process = app.UIAxes;

app.parent\_app.m.Redraw\_process('piv\_processor');

app.Label\_info\_1.Text = app.parent\_app.m.Process\_piv\_\_Get\_info\_progress();

app.Label\_info\_2.Text = app.parent\_app.m.Process\_piv\_\_Get\_info();

drawnow();

end

end

methods (Access = private)

% Code that executes after component creation

function startupFcn(app, parent\_app)

app.parent\_app = parent\_app;

app.Update\_info();

end

% Button pushed function: Button\_start\_one

function Button\_start\_onePushed(app, event)

app.parent\_app.m.Process\_piv\_current();

app.Update\_info();

app.parent\_app.m.Redraw\_fields();

end

% Button pushed function: Button\_start\_all

function Button\_start\_allPushed(app, event)

if app.flag\_run\_PIV, return; end %если уже выполняется не запускать еще раз

app.flag\_run\_PIV = true;

app.parent\_app.m.Process\_piv\_reset();

app.parent\_app.m.Add\_to\_log('Вычисление PIV начато');

for ti = 1 : app.parent\_app.m.ti\_count

app.parent\_app.m.Go\_to\_ti(ti);

app.parent\_app.m.Process\_piv\_current();

% перестроить поля

if app.CheckBox\_plot.Value &&...

rem(ti, app.Spinner\_plot\_framestep.Value) == 0

app.parent\_app.m.Redraw\_fields();

app.parent\_app.init\_gui();

end

if app.CheckBox\_drawnow.Value &&...

rem(ti, app.Spinner\_drawnow\_framestep.Value) == 0

app.Update\_info();

drawnow(); % прерваться на перестроение ГУЯ

end

% выйти если нажата кнопка

if ~app.flag\_run\_PIV

app.parent\_app.m.Add\_to\_log('Вычисление PIV прервано');

app.Label\_info\_2.Text = 'Вычисление PIV прервано';

return;

end

%автосохранение

if app.autosave.Value &&...

rem(ti, app.autosave\_period.Value) == 0

app.parent\_app.m.Proj\_save();

end

end

app.flag\_run\_PIV = false;

app.Update\_info();

app.parent\_app.m.Add\_to\_log('Вычисление PIV закончено');

app.Label\_info\_2.Text = 'Вычисление PIV закончено';

drawnow();

if app.Do\_fi\_after\_piv.Value

app.parent\_app.m.Process\_fi\_all();

app.parent\_app.m.Redraw\_fields();

end

end

% Button pushed function: Button\_stop

function Button\_stopPushed(app, event)

app.flag\_run\_PIV = false; % остановить вычисления

end

% Button pushed function: Button\_start\_unprocessed

function Button\_start\_unprocessedPushed(app, event)

if app.flag\_run\_PIV, return; end %если уже выполняется не запускать еще раз

app.flag\_run\_PIV = true;

app.parent\_app.m.Add\_to\_log('Вычисление PIV начато');

ti\_arr = find(~app.parent\_app.m.Process\_piv\_\_is\_processed\_arr);

for ti\_arr\_i = 1:numel(ti\_arr)

ti = ti\_arr(ti\_arr\_i);

app.parent\_app.m.Go\_to\_ti(ti);

app.parent\_app.m.Process\_piv\_current();

% перестроить поля

if app.CheckBox\_plot.Value &&...

rem(ti, app.Spinner\_plot\_framestep.Value) == 0

app.parent\_app.m.Redraw\_fields();

app.parent\_app.init\_gui();

end

if app.CheckBox\_drawnow.Value &&...

rem(ti, app.Spinner\_drawnow\_framestep.Value) == 0

app.Update\_info();

drawnow(); % прерваться на перестроение ГУЯ

end

% выйти если нажата кнопка

if ~app.flag\_run\_PIV

app.parent\_app.m.Add\_to\_log('Вычисление PIV прервано');

app.Label\_info\_2.Text = 'Вычисление PIV прервано';

return;

end

%автосохранение

if app.autosave.Value &&...

rem(ti, app.autosave\_period.Value) == 0

app.parent\_app.m.Proj\_save();

end

end

app.flag\_run\_PIV = false;

app.Update\_info();

app.parent\_app.m.Add\_to\_log('Вычисление PIV закончено');

app.Label\_info\_2.Text = 'Вычисление PIV закончено';

drawnow();

if app.Do\_fi\_after\_piv.Value

app.parent\_app.m.Process\_fi\_all();

app.parent\_app.m.Redraw\_fields();

end

end

% Button pushed function: Button\_start\_one\_2

function Button\_start\_one\_2Pushed(app, event)

if app.flag\_run\_PIV, return; end %если уже выполняется не запускать еще раз

app.flag\_run\_PIV = true;

ti = app.parent\_app.m.current\_ti;

while app.flag\_run\_PIV

app.parent\_app.m.Process\_piv\_current();

% перестроить поля

if app.CheckBox\_plot.Value &&...

rem(ti, app.Spinner\_plot\_framestep.Value) == 0

app.parent\_app.m.Redraw\_fields();

app.parent\_app.init\_gui();

end

if app.CheckBox\_drawnow.Value &&...

rem(ti, app.Spinner\_drawnow\_framestep.Value) == 0

app.Update\_info();

drawnow(); % прерваться на перестроение ГУЯ

end

% выйти если нажата кнопка

if ~app.flag\_run\_PIV

app.parent\_app.m.Add\_to\_log('Вычисление PIV прервано');

app.Label\_info\_2.Text = 'Вычисление PIV прервано';

break;

end

%автосохранение

if app.autosave.Value &&...

rem(ti, app.autosave\_period.Value) == 0

app.parent\_app.m.Proj\_save();

end

end

app.flag\_run\_PIV = false;

app.Update\_info();

app.parent\_app.m.Add\_to\_log('Вычисление PIV закончено');

app.Label\_info\_2.Text = 'Вычисление PIV закончено';

drawnow();

end

% Button pushed function: Button\_fi

function Button\_fiPushed(app, event)

app.parent\_app.m.Process\_fi\_all();

app.parent\_app.m.Redraw\_fields();

end

end

% App initialization and construction

methods (Access = private)

% Create UIFigure and components

function createComponents(app)

% Create UIFigure

app.UIFigure = uifigure;

app.UIFigure.Position = [100 100 645 523];

app.UIFigure.Name = 'Вычисление полей';

% Create Button\_start\_one

app.Button\_start\_one = uibutton(app.UIFigure, 'push');

app.Button\_start\_one.ButtonPushedFcn = createCallbackFcn(app, @Button\_start\_onePushed, true);

app.Button\_start\_one.Position = [16 487 191 22];

app.Button\_start\_one.Text = 'Вычислить для текущего кадра';

% Create Button\_start\_all

app.Button\_start\_all = uibutton(app.UIFigure, 'push');

app.Button\_start\_all.ButtonPushedFcn = createCallbackFcn(app, @Button\_start\_allPushed, true);

app.Button\_start\_all.Position = [228 487 206 22];

app.Button\_start\_all.Text = 'Запустить с начала';

% Create Button\_stop

app.Button\_stop = uibutton(app.UIFigure, 'push');

app.Button\_stop.ButtonPushedFcn = createCallbackFcn(app, @Button\_stopPushed, true);

app.Button\_stop.Position = [459 458 100 51];

app.Button\_stop.Text = 'Остановить';

% Create UIAxes

app.UIAxes = uiaxes(app.UIFigure);

app.UIAxes.Position = [15 135 625 202];

% Create Label\_info\_1

app.Label\_info\_1 = uilabel(app.UIFigure);

app.Label\_info\_1.Position = [16 104 610 22];

app.Label\_info\_1.Text = 'Info';

% Create CheckBox\_plot

app.CheckBox\_plot = uicheckbox(app.UIFigure);

app.CheckBox\_plot.Text = 'Строить поля во время вычисления';

app.CheckBox\_plot.Position = [15 424 225 22];

app.CheckBox\_plot.Value = true;

% Create Label\_info\_2

app.Label\_info\_2 = uilabel(app.UIFigure);

app.Label\_info\_2.Position = [16 83 610 22];

app.Label\_info\_2.Text = 'Info';

% Create CheckBox\_drawnow

app.CheckBox\_drawnow = uicheckbox(app.UIFigure);

app.CheckBox\_drawnow.Text = 'Обновлять интерфейс во время вычисления';

app.CheckBox\_drawnow.Position = [15 403 275 22];

app.CheckBox\_drawnow.Value = true;

% Create Label

app.Label = uilabel(app.UIFigure);

app.Label.HorizontalAlignment = 'right';

app.Label.Position = [307 403 117 22];

app.Label.Text = 'Период обновления';

% Create Spinner\_drawnow\_framestep

app.Spinner\_drawnow\_framestep = uispinner(app.UIFigure);

app.Spinner\_drawnow\_framestep.Limits = [1 Inf];

app.Spinner\_drawnow\_framestep.Position = [439 403 68 22];

app.Spinner\_drawnow\_framestep.Value = 1;

% Create Label\_2

app.Label\_2 = uilabel(app.UIFigure);

app.Label\_2.HorizontalAlignment = 'right';

app.Label\_2.Position = [307 424 117 22];

app.Label\_2.Text = 'Период обновления';

% Create Spinner\_plot\_framestep

app.Spinner\_plot\_framestep = uispinner(app.UIFigure);

app.Spinner\_plot\_framestep.Limits = [1 Inf];

app.Spinner\_plot\_framestep.Position = [439 424 68 22];

app.Spinner\_plot\_framestep.Value = 1;

% Create Button\_start\_unprocessed

app.Button\_start\_unprocessed = uibutton(app.UIFigure, 'push');

app.Button\_start\_unprocessed.ButtonPushedFcn = createCallbackFcn(app, @Button\_start\_unprocessedPushed, true);

app.Button\_start\_unprocessed.Position = [228 458 206 22];

app.Button\_start\_unprocessed.Text = 'Запустить для необработанных';

% Create Button\_start\_one\_2

app.Button\_start\_one\_2 = uibutton(app.UIFigure, 'push');

app.Button\_start\_one\_2.ButtonPushedFcn = createCallbackFcn(app, @Button\_start\_one\_2Pushed, true);

app.Button\_start\_one\_2.Position = [15 458 191 22];

app.Button\_start\_one\_2.Text = 'Вычислять для текущего кадра';

% Create Button\_fi

app.Button\_fi = uibutton(app.UIFigure, 'push');

app.Button\_fi.ButtonPushedFcn = createCallbackFcn(app, @Button\_fiPushed, true);

app.Button\_fi.Position = [16 42 239 22];

app.Button\_fi.Text = 'Провести фильтрацию и интерполяцию';

% Create Do\_fi\_after\_piv

app.Do\_fi\_after\_piv = uicheckbox(app.UIFigure);

app.Do\_fi\_after\_piv.Text = 'Произвести фильтрацию и интерполяцию после PIV';

app.Do\_fi\_after\_piv.Position = [16 360 318 22];

app.Do\_fi\_after\_piv.Value = true;

% Create autosave

app.autosave = uicheckbox(app.UIFigure);

app.autosave.Text = 'Автоматически сохранять проект';

app.autosave.Position = [15 382 208 22];

% Create Label\_3

app.Label\_3 = uilabel(app.UIFigure);

app.Label\_3.HorizontalAlignment = 'right';

app.Label\_3.Position = [308 382 116 22];

app.Label\_3.Text = 'Период сохранения';

% Create autosave\_period

app.autosave\_period = uispinner(app.UIFigure);

app.autosave\_period.Limits = [1 Inf];

app.autosave\_period.Position = [439 382 68 22];

app.autosave\_period.Value = 1000;

end

end

methods (Access = public)

% Construct app

function app = PIV\_db\_process(varargin)

% Create and configure components

createComponents(app)

% Register the app with App Designer

registerApp(app, app.UIFigure)

% Execute the startup function

runStartupFcn(app, @(app)startupFcn(app, varargin{:}))

if nargout == 0

clear app

end

end

% Code that executes before app deletion

function delete(app)

% Delete UIFigure when app is deleted

delete(app.UIFigure)

end

end

end

## PIV\_db\_proj\_edit.mlapp

classdef PIV\_db\_proj\_edit < matlab.apps.AppBase

% Properties that correspond to app components

properties (Access = public)

new\_project matlab.ui.Figure

im\_path\_b matlab.ui.control.Button

Button\_ok matlab.ui.control.Button

Button\_cancel matlab.ui.control.Button

zy matlab.ui.control.NumericEditField

Label matlab.ui.control.Label

im\_path matlab.ui.control.EditField

Label\_2 matlab.ui.control.Label

comment matlab.ui.control.TextArea

EditFieldLabel matlab.ui.control.Label

EditField matlab.ui.control.NumericEditField

pxEditFieldLabel\_2 matlab.ui.control.Label

pxEditField matlab.ui.control.NumericEditField

pxEditFieldLabel matlab.ui.control.Label

zx matlab.ui.control.NumericEditField

Label\_3 matlab.ui.control.Label

im\_ext matlab.ui.control.EditField

end

properties (Access = private)

parent\_app % Description

end

methods (Access = private)

% Code that executes after component creation

function startupFcn(app, parent\_app)

app.parent\_app = parent\_app;

m = app.parent\_app.m;

app.im\_path.Value = m.p.il.images\_folder;

app.im\_ext.Value = m.p.il.images\_ext;

rp = m.p.rp;

app.comment.Value = rp.comment;

app.EditField.Value = rp.fps;

app.pxEditField.Value = rp.scale;

app.zx.Value = rp.zero\_pos\_px(1);

app.zy.Value = rp.zero\_pos\_px(2);

end

% Close request function: new\_project

function new\_projectCloseRequest(app, event)

delete(app)

end

% Button pushed function: Button\_ok

function Button\_okPushed(app, event)

m = app.parent\_app.m;

il = Image\_loader\_params;

il.images\_folder = app.im\_path.Value;

il.images\_ext = app.im\_ext.Value;

m.Set\_il\_params(il);

rp = Record\_params();

rp.comment = app.comment.Value;

rp.fps = app.EditField.Value;

rp.scale = app.pxEditField.Value;

rp.zero\_pos\_px = [ app.zx.Value app.zy.Value ];

m.Set\_rp\_params(rp);

app.parent\_app.init\_gui();

m.Redraw\_fields();

delete(app);

end

% Button pushed function: Button\_cancel

function Button\_cancelPushed(app, event)

delete(app);

end

% Button pushed function: im\_path\_b

function im\_path\_bButtonPushed(app, event)

selpath = uigetdir(app.im\_path.Value);

if selpath ~= 0

app.im\_path.Value = selpath;

end

end

end

% App initialization and construction

methods (Access = private)

% Create UIFigure and components

function createComponents(app)

% Create new\_project

app.new\_project = uifigure;

app.new\_project.Position = [100 100 641 286];

app.new\_project.Name = 'Параметры проекта';

app.new\_project.CloseRequestFcn = createCallbackFcn(app, @new\_projectCloseRequest, true);

% Create im\_path\_b

app.im\_path\_b = uibutton(app.new\_project, 'push');

app.im\_path\_b.ButtonPushedFcn = createCallbackFcn(app, @im\_path\_bButtonPushed, true);

app.im\_path\_b.Position = [544 256 83 22];

app.im\_path\_b.Text = 'Выбрать';

% Create Button\_ok

app.Button\_ok = uibutton(app.new\_project, 'push');

app.Button\_ok.ButtonPushedFcn = createCallbackFcn(app, @Button\_okPushed, true);

app.Button\_ok.Position = [526 16 100 22];

app.Button\_ok.Text = 'Применить';

% Create Button\_cancel

app.Button\_cancel = uibutton(app.new\_project, 'push');

app.Button\_cancel.ButtonPushedFcn = createCallbackFcn(app, @Button\_cancelPushed, true);

app.Button\_cancel.Position = [410 16 100 22];

app.Button\_cancel.Text = 'Отмена';

% Create zy

app.zy = uieditfield(app.new\_project, 'numeric');

app.zy.Position = [303 58 100 22];

% Create Label

app.Label = uilabel(app.new\_project);

app.Label.HorizontalAlignment = 'right';

app.Label.Position = [4 256 177 22];

app.Label.Text = 'Путь к папке с изображениями';

% Create im\_path

app.im\_path = uieditfield(app.new\_project, 'text');

app.im\_path.Position = [196 256 335 22];

% Create Label\_2

app.Label\_2 = uilabel(app.new\_project);

app.Label\_2.HorizontalAlignment = 'right';

app.Label\_2.Position = [119 189 61 22];

app.Label\_2.Text = 'Описание';

% Create comment

app.comment = uitextarea(app.new\_project);

app.comment.Position = [195 153 432 60];

% Create EditFieldLabel

app.EditFieldLabel = uilabel(app.new\_project);

app.EditFieldLabel.HorizontalAlignment = 'right';

app.EditFieldLabel.Position = [61 119 118 22];

app.EditFieldLabel.Text = 'Частота кадров (к/с)';

% Create EditField

app.EditField = uieditfield(app.new\_project, 'numeric');

app.EditField.Position = [194 119 100 22];

app.EditField.Value = 1;

% Create pxEditFieldLabel\_2

app.pxEditFieldLabel\_2 = uilabel(app.new\_project);

app.pxEditFieldLabel\_2.HorizontalAlignment = 'right';

app.pxEditFieldLabel\_2.Position = [73 88 106 22];

app.pxEditFieldLabel\_2.Text = 'Масштаб (мкм/px)';

% Create pxEditField

app.pxEditField = uieditfield(app.new\_project, 'numeric');

app.pxEditField.Position = [194 88 100 22];

app.pxEditField.Value = 1;

% Create pxEditFieldLabel

app.pxEditFieldLabel = uilabel(app.new\_project);

app.pxEditFieldLabel.HorizontalAlignment = 'right';

app.pxEditFieldLabel.Position = [47 58 132 22];

app.pxEditFieldLabel.Text = 'Начало координат (px)';

% Create zx

app.zx = uieditfield(app.new\_project, 'numeric');

app.zx.Position = [194 58 100 22];

% Create Label\_3

app.Label\_3 = uilabel(app.new\_project);

app.Label\_3.HorizontalAlignment = 'right';

app.Label\_3.Position = [29 223 152 22];

app.Label\_3.Text = 'Расширение изображений';

% Create im\_ext

app.im\_ext = uieditfield(app.new\_project, 'text');

app.im\_ext.Position = [196 223 31 22];

app.im\_ext.Value = 'jpg';

end

end

methods (Access = public)

% Construct app

function app = PIV\_db\_proj\_edit(varargin)

% Create and configure components

createComponents(app)

% Register the app with App Designer

registerApp(app, app.new\_project)

% Execute the startup function

runStartupFcn(app, @(app)startupFcn(app, varargin{:}))

if nargout == 0

clear app

end

end

% Code that executes before app deletion

function delete(app)

% Delete UIFigure when app is deleted

delete(app.new\_project)

end

end

end

## PIV\_db\_proj\_new.mlapp

classdef PIV\_db\_proj\_new < matlab.apps.AppBase

% Properties that correspond to app components

properties (Access = public)

new\_project matlab.ui.Figure

Label matlab.ui.control.Label

im\_path matlab.ui.control.EditField

Label\_2 matlab.ui.control.Label

comment matlab.ui.control.TextArea

im\_path\_b matlab.ui.control.Button

Button\_ok matlab.ui.control.Button

Button\_cancel matlab.ui.control.Button

fpsEditFieldLabel matlab.ui.control.Label

fpsEditField matlab.ui.control.NumericEditField

scalepxEditFieldLabel matlab.ui.control.Label

scalepxEditField matlab.ui.control.NumericEditField

Label\_3 matlab.ui.control.Label

proj\_path matlab.ui.control.EditField

proj\_path\_b matlab.ui.control.Button

pxEditFieldLabel matlab.ui.control.Label

zx matlab.ui.control.NumericEditField

zy matlab.ui.control.NumericEditField

Label\_4 matlab.ui.control.Label

im\_ext matlab.ui.control.EditField

end

properties (Access = private)

parent\_app % Description

end

methods (Access = private)

% Code that executes after component creation

function startupFcn(app, parent\_app)

app.parent\_app = parent\_app;

m = app.parent\_app.m;

if ~isempty(app.parent\_app.m.core)

app.proj\_path.Value = m.proj\_path;

app.im\_path.Value = m.p.il.images\_folder;

app.im\_ext.Value = m.p.il.images\_ext;

rp = m.p.rp;

app.comment.Value = rp.comment;

app.fpsEditField.Value = rp.fps;

app.scalepxEditField.Value = rp.scale;

app.zx.Value = rp.zero\_pos\_px(1);

app.zy.Value = rp.zero\_pos\_px(2);

end

end

% Button pushed function: im\_path\_b

function im\_path\_bPushed(app, event)

selpath = uigetdir(app.im\_path.Value);

if selpath ~= 0

app.im\_path.Value = selpath;

end

end

% Button pushed function: Button\_cancel

function Button\_cancelPushed(app, event)

delete(app);

end

% Button pushed function: Button\_ok

function Button\_okPushed(app, event)

m = app.parent\_app.m;

m.Proj\_new(...

app.im\_path.Value, ...

app.im\_ext.Value, ...

app.proj\_path.Value);

rp = Record\_params();

rp.comment = app.comment.Value;

rp.fps = app.fpsEditField.Value;

rp.scale = app.scalepxEditField.Value;

rp.zero\_pos\_px = [ app.zx.Value app.zy.Value ];

m.Set\_rp\_params(rp);

app.parent\_app.init\_gui();

delete(app);

end

% Button pushed function: proj\_path\_b

function proj\_path\_bPushed(app, event)

[file,path] = uiputfile('\*.proj');

if file == 0, return; end

selpath = fullfile(path,file);

app.proj\_path.Value = selpath;

end

end

% App initialization and construction

methods (Access = private)

% Create UIFigure and components

function createComponents(app)

% Create new\_project

app.new\_project = uifigure;

app.new\_project.Position = [100 100 636 364];

app.new\_project.Name = 'Параметры проекта';

% Create Label

app.Label = uilabel(app.new\_project);

app.Label.HorizontalAlignment = 'right';

app.Label.Position = [4 278 177 22];

app.Label.Text = 'Путь к папке с изображениями';

% Create im\_path

app.im\_path = uieditfield(app.new\_project, 'text');

app.im\_path.Position = [196 278 335 22];

% Create Label\_2

app.Label\_2 = uilabel(app.new\_project);

app.Label\_2.HorizontalAlignment = 'right';

app.Label\_2.Position = [4 211 61 22];

app.Label\_2.Text = 'Описание';

% Create comment

app.comment = uitextarea(app.new\_project);

app.comment.Position = [80 175 547 60];

% Create im\_path\_b

app.im\_path\_b = uibutton(app.new\_project, 'push');

app.im\_path\_b.ButtonPushedFcn = createCallbackFcn(app, @im\_path\_bPushed, true);

app.im\_path\_b.Position = [544 278 83 22];

app.im\_path\_b.Text = 'Выбрать';

% Create Button\_ok

app.Button\_ok = uibutton(app.new\_project, 'push');

app.Button\_ok.ButtonPushedFcn = createCallbackFcn(app, @Button\_okPushed, true);

app.Button\_ok.Position = [525 38 102 22];

app.Button\_ok.Text = 'Создать проект';

% Create Button\_cancel

app.Button\_cancel = uibutton(app.new\_project, 'push');

app.Button\_cancel.ButtonPushedFcn = createCallbackFcn(app, @Button\_cancelPushed, true);

app.Button\_cancel.Position = [410 38 100 22];

app.Button\_cancel.Text = 'Отмена';

% Create fpsEditFieldLabel

app.fpsEditFieldLabel = uilabel(app.new\_project);

app.fpsEditFieldLabel.HorizontalAlignment = 'right';

app.fpsEditFieldLabel.Position = [110 140 47 22];

app.fpsEditFieldLabel.Text = 'fps (к/с)';

% Create fpsEditField

app.fpsEditField = uieditfield(app.new\_project, 'numeric');

app.fpsEditField.Position = [172 140 100 22];

app.fpsEditField.Value = 1;

% Create scalepxEditFieldLabel

app.scalepxEditFieldLabel = uilabel(app.new\_project);

app.scalepxEditFieldLabel.HorizontalAlignment = 'right';

app.scalepxEditFieldLabel.Position = [74 111 83 22];

app.scalepxEditFieldLabel.Text = 'scale (мкм/px)';

% Create scalepxEditField

app.scalepxEditField = uieditfield(app.new\_project, 'numeric');

app.scalepxEditField.Position = [172 111 100 22];

app.scalepxEditField.Value = 1;

% Create Label\_3

app.Label\_3 = uilabel(app.new\_project);

app.Label\_3.HorizontalAlignment = 'right';

app.Label\_3.Position = [54 314 127 22];

app.Label\_3.Text = 'Путь к файлу проекта';

% Create proj\_path

app.proj\_path = uieditfield(app.new\_project, 'text');

app.proj\_path.Position = [196 314 335 22];

% Create proj\_path\_b

app.proj\_path\_b = uibutton(app.new\_project, 'push');

app.proj\_path\_b.ButtonPushedFcn = createCallbackFcn(app, @proj\_path\_bPushed, true);

app.proj\_path\_b.Position = [544 314 83 22];

app.proj\_path\_b.Text = 'Выбрать';

% Create pxEditFieldLabel

app.pxEditFieldLabel = uilabel(app.new\_project);

app.pxEditFieldLabel.HorizontalAlignment = 'right';

app.pxEditFieldLabel.Position = [25 79 132 22];

app.pxEditFieldLabel.Text = 'Начало координат (px)';

% Create zx

app.zx = uieditfield(app.new\_project, 'numeric');

app.zx.Position = [172 79 100 22];

% Create zy

app.zy = uieditfield(app.new\_project, 'numeric');

app.zy.Position = [281 79 100 22];

% Create Label\_4

app.Label\_4 = uilabel(app.new\_project);

app.Label\_4.HorizontalAlignment = 'right';

app.Label\_4.Position = [29 245 152 22];

app.Label\_4.Text = 'Расширение изображений';

% Create im\_ext

app.im\_ext = uieditfield(app.new\_project, 'text');

app.im\_ext.Position = [196 245 31 22];

app.im\_ext.Value = 'jpg';

end

end

methods (Access = public)

% Construct app

function app = PIV\_db\_proj\_new(varargin)

% Create and configure components

createComponents(app)

% Register the app with App Designer

registerApp(app, app.new\_project)

% Execute the startup function

runStartupFcn(app, @(app)startupFcn(app, varargin{:}))

if nargout == 0

clear app

end

end

% Code that executes before app deletion

function delete(app)

% Delete UIFigure when app is deleted

delete(app.new\_project)

end

end

end

## PIV\_main\_App.mlapp

classdef PIV\_main\_App < matlab.apps.AppBase

% Properties that correspond to app components

properties (Access = public)

UIFigure matlab.ui.Figure

Menu matlab.ui.container.Menu

Menu\_proj\_new matlab.ui.container.Menu

Menu\_proj\_load matlab.ui.container.Menu

Menu\_proj\_save matlab.ui.container.Menu

Menu\_proj\_saveas matlab.ui.container.Menu

Menu\_proj\_edit matlab.ui.container.Menu

Menu\_7 matlab.ui.container.Menu

Menu\_ipp matlab.ui.container.Menu

Menu\_pg matlab.ui.container.Menu

Menu\_pf matlab.ui.container.Menu

Menu\_pp matlab.ui.container.Menu

Menu\_pfi matlab.ui.container.Menu

Menu\_plot matlab.ui.container.Menu

Run matlab.ui.container.Menu

RunPIV matlab.ui.container.Menu

Menu\_24 matlab.ui.container.Menu

Menu\_25 matlab.ui.container.Menu

FrameSlider\_frame\_number matlab.ui.control.Slider

Label matlab.ui.control.Label

Spinner\_frame\_number matlab.ui.control.Spinner

Label\_2 matlab.ui.control.Label

Spinner\_ti matlab.ui.control.Spinner

Play matlab.ui.control.Button

Stop matlab.ui.control.Button

Play\_2 matlab.ui.control.Button

play\_loop matlab.ui.control.CheckBox

Readraw matlab.ui.control.Button

Panel\_sp matlab.ui.container.Panel

DropDown\_sp\_add\_pxm matlab.ui.control.DropDown

Button\_sp\_add matlab.ui.control.Button

xyEditField\_y matlab.ui.control.NumericEditField

xyLabel matlab.ui.control.Label

xyEditField\_x matlab.ui.control.NumericEditField

Button\_sp\_delete\_all matlab.ui.control.Button

end

properties (Access = private)

end

properties (Access = public)

m PIV\_main

fig\_fields

flag\_play = false;

%Диалогбоксы

db\_proj\_new PIV\_db\_proj\_new

db\_proj\_edit PIV\_db\_proj\_edit

db\_piv\_grid PIV\_db\_piv\_grid

db\_piv\_params PIV\_db\_piv\_params

db\_ipp PIV\_db\_ipp

db\_piv\_frames PIV\_db\_piv\_frames

db\_plot PIV\_db\_plot

db\_process PIV\_db\_process

db\_fi\_params PIV\_db\_fi\_params

db\_export PIV\_db\_export

db\_analyze PIV\_db\_analyze

end

methods (Access = public)

function init\_gui(app) % привести гуй в соответствие с настройками. весь бэкенд должен быть инициализирован на этом этапе.

app.UIFigure.Name = app.m.proj\_path;

if isempty(app.m.core), return; end

app.FrameSlider\_frame\_number.Limits = [1 app.m.im\_count];

app.Spinner\_frame\_number.Limits = [1 app.m.im\_count];

if (app.m.ti\_count > 1)

app.Spinner\_ti.Enable = true;

app.Spinner\_ti.Limits = [1 app.m.ti\_count];

else

app.Spinner\_ti.Enable = false;

end

app.Spinner\_frame\_number.Value = app.m.current\_frame;

app.FrameSlider\_frame\_number.Value = app.m.current\_frame;

if ~isnan(app.m.current\_ti)

app.Spinner\_ti.Value = app.m.current\_ti;

end

end

function proj\_load(app, proj\_path)

app.add\_to\_log(sprintf('Загружается проект из %s', proj\_path));

app.P\_path = proj\_path;

f = load(app.P\_path,'-mat');

app.P = f.Proj;

app.init();

app.add\_to\_log(sprintf('Загружен проект из %s', proj\_path));

end

end

methods (Access = private)

% Code that executes after component creation

function startupFcn(app, piv\_main)

app.m = piv\_main;

% piv\_main.gui\_app = app;

app.m.Redraw\_fields();

app.init\_gui();

end

% Value changed function: FrameSlider\_frame\_number

function FrameSlider\_frame\_numberValueChanged(app, event)

value = round(app.FrameSlider\_frame\_number.Value);

app.m.Go\_to\_frame(value);

app.m.Redraw\_fields();

app.init\_gui();

end

% Menu selected function: Menu\_pg

function Menu\_pgSelected(app, event)

app.db\_piv\_grid = PIV\_db\_piv\_grid(app);

end

% Menu selected function: Menu\_proj\_new

function Menu\_proj\_newSelected(app, event)

app.db\_proj\_new = PIV\_db\_proj\_new(app);

end

% Menu selected function: Menu\_proj\_saveas

function Menu\_proj\_saveasSelected(app, event)

[file,path] = uiputfile('\*.proj');

if file == 0, return; end

selpath = fullfile(path,file);

app.m.proj\_path = selpath;

app.m.Proj\_save(app.m.proj\_path);

app.m.Add\_to\_log(sprintf('Проект сохренен в %s', app.m.proj\_path));

app.init\_gui();

end

% Menu selected function: Menu\_proj\_load

function Menu\_proj\_loadSelected(app, event)

[file,path] = uigetfile('\*.proj');

if file == 0, return; end

file\_path = fullfile(path,file);

app.m.Proj\_load(file\_path);

app.init\_gui();

end

% Menu selected function: Menu\_proj\_save

function Menu\_proj\_saveSelected(app, event)

if isempty(app.m.proj\_path)

[file,path] = uiputfile('\*.proj');

if file == 0, return; end

selpath = fullfile(path,file);

app.m.proj\_path = selpath;

end

app.m.Proj\_save(app.m.proj\_path);

app.m.Add\_to\_log(sprintf('Проект сохренен в %s', app.m.proj\_path));

end

% Close request function: UIFigure

function UIFigureCloseRequest(app, event)

% %save

% msg = 'Сохранить проект перед закрытием программы?';

% title = 'Выход из программы';

% selection = uiconfirm(app.UIFigure,msg,title,...

% 'Options',{'Сохранить','Не сохранять','Отмена'},...

% 'DefaultOption',1,'CancelOption',3);

% switch (selection)

% case 'Сохранить'

% if isempty(app.m.proj\_path)

% file\_path = uiputfile('\*.piv\_aura\_proj');

% if file\_path == 0, return; end

% app.m.proj\_path = file\_path;

% end

% app.m.proj\_save(app.m.proj\_path);

% app.m.add\_to\_log(sprintf('Проект сохренен в %s', app.m.proj\_path));

% case 'Отмена'

% return;

% end

delete(app.db\_proj\_new);

delete(app.db\_proj\_edit);

delete(app.db\_piv\_grid);

delete(app.db\_piv\_params);

% TODO провеить все ли на месте?

if ishandle(app.m) && ishandle(app.m.fig\_fields)

close(app.m.fig\_fields);

end

delete(app);

end

% Menu selected function: Menu\_proj\_edit

function Menu\_proj\_editSelected(app, event)

app.db\_proj\_edit = PIV\_db\_proj\_edit(app);

end

% Menu selected function: Menu\_pp

function Menu\_ppSelected(app, event)

app.db\_piv\_params = PIV\_db\_piv\_params(app);

end

% Callback function

function Menu\_9Selected(app, event)

app.run\_current\_frame();

end

% Value changed function: Spinner\_frame\_number

function Spinner\_frame\_numberValueChanged(app, event)

value = round(app.Spinner\_frame\_number.Value);

app.m.Go\_to\_frame(value);

app.m.Redraw\_fields();

app.init\_gui();

end

% Menu selected function: Menu\_ipp

function Menu\_ippSelected(app, event)

app.db\_ipp = PIV\_db\_ipp(app);

end

% Menu selected function: Menu\_pf

function Menu\_pfSelected(app, event)

app.db\_piv\_frames = PIV\_db\_piv\_frames(app);

end

% Value changed function: Spinner\_ti

function Spinner\_tiValueChanged(app, event)

value = app.Spinner\_ti.Value;

app.m.Go\_to\_ti(value);

app.m.Redraw\_fields();

app.init\_gui();

end

% Menu selected function: Menu\_plot

function Menu\_plotSelected(app, event)

app.db\_plot = PIV\_db\_plot(app);

end

% Menu selected function: RunPIV

function RunPIVMenuSelected(app, event)

app.db\_process = PIV\_db\_process(app);

end

% Menu selected function: Menu\_pfi

function Menu\_pfiSelected(app, event)

app.db\_fi\_params = PIV\_db\_fi\_params(app);

end

% Button pushed function: Play

function PlayButtonPushed(app, event)

if app.flag\_play, return; end

app.flag\_play = true;

while(app.flag\_play)

frame\_i = app.m.current\_frame;

if frame\_i >= app.m.im\_count

if app.play\_loop.Value

frame\_i = 1;

else

break;

end

end

app.m.Go\_to\_frame( frame\_i + 1 );

app.m.Redraw\_fields();

app.init\_gui();

drawnow();

end

app.flag\_play = false;

end

% Button pushed function: Stop

function StopButtonPushed(app, event)

app.flag\_play = false;

end

% Button pushed function: Play\_2

function Play\_2ButtonPushed(app, event)

if app.flag\_play, return; end

app.flag\_play = true;

while(app.flag\_play)

ti = app.m.current\_ti;

if ti >= app.m.ti\_count

if app.play\_loop.Value

ti = 1;

else

break;

end

end

app.m.Go\_to\_ti( ti + 1 );

app.m.Redraw\_fields();

app.init\_gui();

drawnow();

end

app.flag\_play = false;

end

% Callback function

function Stop\_2ButtonPushed(app, event)

app.flag\_play = false;

end

% Menu selected function: Menu\_24

function Menu\_24Selected(app, event)

app.db\_analyze = PIV\_db\_analyze(app);

end

% Menu selected function: Menu\_25

function Menu\_25Selected(app, event)

app.db\_export = PIV\_db\_export(app);

end

% Button pushed function: Readraw

function ReadrawButtonPushed(app, event)

app.m.Redraw\_fields();

end

% Button pushed function: Button\_sp\_add

function Button\_sp\_addPushed(app, event)

x = app.xyEditField\_x.Value;

y = app.xyEditField\_y.Value;

if strcmp(app.DropDown\_sp\_add\_pxm.Value, 'px')

app.m.Add\_selected\_point\_px(x,y);

else

app.m.Add\_selected\_point\_m(x,y);

end

app.m.Redraw\_fields();

end

% Button pushed function: Button\_sp\_delete\_all

function Button\_sp\_delete\_allPushed(app, event)

app.m.Clear\_selected\_point\_m();

app.m.Redraw\_fields();

end

end

% App initialization and construction

methods (Access = private)

% Create UIFigure and components

function createComponents(app)

% Create UIFigure

app.UIFigure = uifigure;

app.UIFigure.Position = [100 100 614 383];

app.UIFigure.Name = 'Основное окно';

app.UIFigure.CloseRequestFcn = createCallbackFcn(app, @UIFigureCloseRequest, true);

% Create Menu

app.Menu = uimenu(app.UIFigure);

app.Menu.Text = 'Проект';

% Create Menu\_proj\_new

app.Menu\_proj\_new = uimenu(app.Menu);

app.Menu\_proj\_new.MenuSelectedFcn = createCallbackFcn(app, @Menu\_proj\_newSelected, true);

app.Menu\_proj\_new.Text = 'Новый';

% Create Menu\_proj\_load

app.Menu\_proj\_load = uimenu(app.Menu);

app.Menu\_proj\_load.MenuSelectedFcn = createCallbackFcn(app, @Menu\_proj\_loadSelected, true);

app.Menu\_proj\_load.Text = 'Открыть';

% Create Menu\_proj\_save

app.Menu\_proj\_save = uimenu(app.Menu);

app.Menu\_proj\_save.MenuSelectedFcn = createCallbackFcn(app, @Menu\_proj\_saveSelected, true);

app.Menu\_proj\_save.Text = 'Сохранить';

% Create Menu\_proj\_saveas

app.Menu\_proj\_saveas = uimenu(app.Menu);

app.Menu\_proj\_saveas.MenuSelectedFcn = createCallbackFcn(app, @Menu\_proj\_saveasSelected, true);

app.Menu\_proj\_saveas.Text = 'Сохранить как';

% Create Menu\_proj\_edit

app.Menu\_proj\_edit = uimenu(app.Menu);

app.Menu\_proj\_edit.MenuSelectedFcn = createCallbackFcn(app, @Menu\_proj\_editSelected, true);

app.Menu\_proj\_edit.Text = 'Параметры проекта';

% Create Menu\_7

app.Menu\_7 = uimenu(app.UIFigure);

app.Menu\_7.Text = 'Параметры';

% Create Menu\_ipp

app.Menu\_ipp = uimenu(app.Menu\_7);

app.Menu\_ipp.MenuSelectedFcn = createCallbackFcn(app, @Menu\_ippSelected, true);

app.Menu\_ipp.Text = 'Параметры изображений';

% Create Menu\_pg

app.Menu\_pg = uimenu(app.Menu\_7);

app.Menu\_pg.MenuSelectedFcn = createCallbackFcn(app, @Menu\_pgSelected, true);

app.Menu\_pg.Text = 'Параметры сетки';

% Create Menu\_pf

app.Menu\_pf = uimenu(app.Menu\_7);

app.Menu\_pf.MenuSelectedFcn = createCallbackFcn(app, @Menu\_pfSelected, true);

app.Menu\_pf.Text = 'Параметры кадров';

% Create Menu\_pp

app.Menu\_pp = uimenu(app.Menu\_7);

app.Menu\_pp.MenuSelectedFcn = createCallbackFcn(app, @Menu\_ppSelected, true);

app.Menu\_pp.Text = 'Параметры PIV';

% Create Menu\_pfi

app.Menu\_pfi = uimenu(app.Menu\_7);

app.Menu\_pfi.MenuSelectedFcn = createCallbackFcn(app, @Menu\_pfiSelected, true);

app.Menu\_pfi.Text = 'Фильтрация и интерполяция';

% Create Menu\_plot

app.Menu\_plot = uimenu(app.Menu\_7);

app.Menu\_plot.MenuSelectedFcn = createCallbackFcn(app, @Menu\_plotSelected, true);

app.Menu\_plot.Text = 'Отображение';

% Create Run

app.Run = uimenu(app.UIFigure);

app.Run.Text = 'Инструменты';

% Create RunPIV

app.RunPIV = uimenu(app.Run);

app.RunPIV.MenuSelectedFcn = createCallbackFcn(app, @RunPIVMenuSelected, true);

app.RunPIV.Text = 'Вычисление полей';

% Create Menu\_24

app.Menu\_24 = uimenu(app.Run);

app.Menu\_24.MenuSelectedFcn = createCallbackFcn(app, @Menu\_24Selected, true);

app.Menu\_24.Text = 'Анализ полей';

% Create Menu\_25

app.Menu\_25 = uimenu(app.Run);

app.Menu\_25.MenuSelectedFcn = createCallbackFcn(app, @Menu\_25Selected, true);

app.Menu\_25.Text = 'Экспорт';

% Create FrameSlider\_frame\_number

app.FrameSlider\_frame\_number = uislider(app.UIFigure);

app.FrameSlider\_frame\_number.Limits = [0 7];

app.FrameSlider\_frame\_number.ValueChangedFcn = createCallbackFcn(app, @FrameSlider\_frame\_numberValueChanged, true);

app.FrameSlider\_frame\_number.Position = [16 362 585 3];

% Create Label

app.Label = uilabel(app.UIFigure);

app.Label.HorizontalAlignment = 'right';

app.Label.Position = [16 276 36 22];

app.Label.Text = 'Кадр:';

% Create Spinner\_frame\_number

app.Spinner\_frame\_number = uispinner(app.UIFigure);

app.Spinner\_frame\_number.ValueChangedFcn = createCallbackFcn(app, @Spinner\_frame\_numberValueChanged, true);

app.Spinner\_frame\_number.Position = [67 276 100 22];

% Create Label\_2

app.Label\_2 = uilabel(app.UIFigure);

app.Label\_2.HorizontalAlignment = 'right';

app.Label\_2.Position = [18 244 34 22];

app.Label\_2.Text = 'Поле';

% Create Spinner\_ti

app.Spinner\_ti = uispinner(app.UIFigure);

app.Spinner\_ti.ValueChangedFcn = createCallbackFcn(app, @Spinner\_tiValueChanged, true);

app.Spinner\_ti.Position = [67 244 100 22];

% Create Play

app.Play = uibutton(app.UIFigure, 'push');

app.Play.ButtonPushedFcn = createCallbackFcn(app, @PlayButtonPushed, true);

app.Play.Position = [182 276 100 22];

app.Play.Text = 'Старт';

% Create Stop

app.Stop = uibutton(app.UIFigure, 'push');

app.Stop.ButtonPushedFcn = createCallbackFcn(app, @StopButtonPushed, true);

app.Stop.Position = [297 244 84 54];

app.Stop.Text = 'Стоп';

% Create Play\_2

app.Play\_2 = uibutton(app.UIFigure, 'push');

app.Play\_2.ButtonPushedFcn = createCallbackFcn(app, @Play\_2ButtonPushed, true);

app.Play\_2.Position = [182 244 100 22];

app.Play\_2.Text = 'Старт';

% Create play\_loop

app.play\_loop = uicheckbox(app.UIFigure);

app.play\_loop.Text = 'Циклическое воспроизведение';

app.play\_loop.Position = [395 244 196 22];

% Create Readraw

app.Readraw = uibutton(app.UIFigure, 'push');

app.Readraw.ButtonPushedFcn = createCallbackFcn(app, @ReadrawButtonPushed, true);

app.Readraw.Position = [395 276 137 22];

app.Readraw.Text = 'Перестроить текущий';

% Create Panel\_sp

app.Panel\_sp = uipanel(app.UIFigure);

app.Panel\_sp.Title = 'Выбор точек';

app.Panel\_sp.Position = [23 107 397 113];

% Create DropDown\_sp\_add\_pxm

app.DropDown\_sp\_add\_pxm = uidropdown(app.Panel\_sp);

app.DropDown\_sp\_add\_pxm.Items = {'px', 'м', ''};

app.DropDown\_sp\_add\_pxm.Position = [317 50 43 22];

app.DropDown\_sp\_add\_pxm.Value = 'px';

% Create Button\_sp\_add

app.Button\_sp\_add = uibutton(app.Panel\_sp, 'push');

app.Button\_sp\_add.ButtonPushedFcn = createCallbackFcn(app, @Button\_sp\_addPushed, true);

app.Button\_sp\_add.Position = [9 50 100 22];

app.Button\_sp\_add.Text = 'Добавить';

% Create xyEditField\_y

app.xyEditField\_y = uieditfield(app.Panel\_sp, 'numeric');

app.xyEditField\_y.Position = [247 50 53 22];

% Create xyLabel

app.xyLabel = uilabel(app.Panel\_sp);

app.xyLabel.HorizontalAlignment = 'right';

app.xyLabel.Position = [141 50 25 22];

app.xyLabel.Text = 'x, y';

% Create xyEditField\_x

app.xyEditField\_x = uieditfield(app.Panel\_sp, 'numeric');

app.xyEditField\_x.Position = [181 50 54 22];

% Create Button\_sp\_delete\_all

app.Button\_sp\_delete\_all = uibutton(app.Panel\_sp, 'push');

app.Button\_sp\_delete\_all.ButtonPushedFcn = createCallbackFcn(app, @Button\_sp\_delete\_allPushed, true);

app.Button\_sp\_delete\_all.Position = [9 17 100 22];

app.Button\_sp\_delete\_all.Text = {'Удалить все'; ''};

end

end

methods (Access = public)

% Construct app

function app = PIV\_main\_App(varargin)

% Create and configure components

createComponents(app)

% Register the app with App Designer

registerApp(app, app.UIFigure)

% Execute the startup function

runStartupFcn(app, @(app)startupFcn(app, varargin{:}))

if nargout == 0

clear app

end

end

% Code that executes before app deletion

function delete(app)

% Delete UIFigure when app is deleted

delete(app.UIFigure)

end

end

end