

```

1  classdef class_videoinput < handle
2      %CLASS_WINVIDEO Communication with a camera via videoinput
3      %
4      properties (SetAccess = immutable)
5          name = 'NAME';      % an human readable identifier
6          type = '';          % videoinput type
7          device_name = '';   % like in imaqhwinfo(type,
8                               id).DeviceName
9          format = '';        % format for videoinput function
10         color_space = '';    % ReturnedColorSpace
11         id = false;          % ID-Number
12     end
13
14     properties (Access = private)
15         handle = false;      % raw handle of the camera connection
16         prev_timer = false;% timer for async snapshot preview
17     end
18
19     methods
20         % constructor
21         function obj = class_videoinput(name, type, format,
22                                         color_space, device_name)
23             obj.name = name;
24             obj.type = type;
25             obj.format = format;
26             obj.color_space = color_space;
27             obj.device_name = device_name;
28
29             % find camera id
30             cam_id = -1;
31             info = imaqhwinfo(obj.type);
32             for i = info.DeviceIDs
33                 dev_info = imaqhwinfo(obj.type, i{1});
34                 if strcmp(dev_info.DeviceName, device_name)
35                     cam_id = i{1};
36                 end
37             end
38
39             % return if device_name is not found
40             if cam_id == -1
41                 error('Can''t find camera %s by DeviceName
42                        "%s"', name, device_name);
43             end
44
45             % initialize camera
46             obj.id = cam_id;
47             fprintf('Found videoinput camera %s at ID %d.\n',
48                     name, cam_id);
49         end
50     end
51 end

```

```

46
47 % connect to the camera
48 function success = connect(obj)
49     success = false;
50     if obj.handle ~= false
51         success = true;
52         return;
53     end
54
55     if obj.id == false
56         warning('Can''t connect to %s, don''t know
57             ID.', obj.name);
58         return;
59     end
60
61     try
62         obj.handle = videoinput(obj.type, obj.id,
63             obj.format);
64         obj.handle.ReturnedColorSpace = obj.color_space;
65         fprintf('Connected to %s\n', obj.name);
66         success = true;
67     catch e
68         warning('Connecting to %s failed:
69             videoinput(''%s'', %d, '">%s''). %s', obj.type,
70             obj.name, obj.id, obj.format, getReport(e));
71     end
72 end
73
74 % close camera connection
75 function close(obj)
76     if obj.handle ~= false
77         delete(obj.handle);
78         obj.handle = false;
79     end
80     fprintf('Closed connection to %s\n', obj.name);
81 end
82
83 % Call config_function with handle if connected
84 function [success, varargout] = config(obj,
85     config_function)
86     success = false;
87     varargout = cell(1, nargin - 1);
88
89     % warn and return if connecting failes
90     if obj.connect() == false
91         warning('Config error: Can''t connect to camera
92             %s.', obj.name);
93         return;
94     end

```

```

89
90     try
91         if isa(config_function, 'function_handle')
92             [obj.handle, vararginout{1:nargout - 1}] =
                config_function(obj.handle);
93         elseif iscell(config_function) &&
            isa(config_function{1}, 'function_handle')
94             [obj.handle, vararginout{1:nargout - 1}] =
                config_function{1}(obj.handle,
                    config_function{2:end});
95         else
96             error('videoinput config callback is not
                callable');
97         end
98
99         success = true;
100     catch e
101         warning('Exception while config %s: %s',
            obj.name, getReport(e));
102     end
103 end
104
105 % preview live image
106 function success = preview(obj, adjust_function, axes,
    rotate)
107     success = false;
108
109     % warn and return if connecting failes
110     if obj.connect() == false
111         warning('Preview error: Can''t connect to
            camera %s.', obj.name);
112         return;
113     end
114
115     try
116         size = obj.handle.ROIPosition;
117         bands = obj.handle.NumberOfBands;
118
119         if nargin > 3 && rotate == true
120             size([3, 4]) = size([4, 3]);
121         end
122
123         if nargin < 3
124             % preview as figure
125             hImage = image(zeros(size(4), size(3),
                bands));
126         else
127             % preview on axes in GUI
128             hImage = image(zeros(size(4), size(3),

```

```

        bands), 'Parent', axes);
129     axes.DataAspectRatio = [1, 1, 1];
130     end
131
132     if nargin > 1 && isa(adjust_function,
        'function_handle')
133         setappdata(hImage,
            'UpdatePreviewWindowFcn', adjust_function);
134     end
135
136     preview(obj.handle, hImage);
137
138     success = true;
139     catch e
140         warning('Exception while preview %s: %s',
            obj.name, getReport(e));
141     end
142 end
143
144 % stop preview live image
145 function success = stoppreview(obj)
146     success = false;
147     if obj.handle == false
148         return;
149     end
150
151     try
152         if obj.prev_timer ~= false
153             stop(obj.prev_timer);
154             delete(obj.prev_timer);
155             obj.prev_timer = false;
156         end
157
158         closepreview(obj.handle);
159         success = true;
160     catch e
161         warning('Exception while preview %s: %s',
            obj.name, getReport(e));
162     end
163 end
164
165 function [success, img] = snapshot(obj,
    adjust_function, axes)
166     success = false;
167
168     try
169         if nargin < 2
170             adjust_function = false;
171         end

```

```

172
173         img = false;
174
175         obj.stoppreview();
176
177         % warn and return if connecting failes
178         if obj.connect() == false
179             warning('Snapshot error: Can''t connect to
180                 camera %s.', obj.name);
181             return;
182         end
183
184         img = getsnapshot(obj.handle);
185
186         if nargin > 2
187             if isa(adjust_function, 'function_handle')
188                 img = adjust_function(img);
189             end
190
191             imshow(img, 'Parent', axes);
192         else
193             imshow(img);
194         end
195
196         success = true;
197         catch e
198             warning('Exception while snapshot %s: %s',
199                 obj.name, getReport(e));
200         end
201
202         function delete(obj)
203             obj.close();
204         end
205     end
206

```