

```

1  classdef class_gigecam < handle
2      %CLASS_GIGECAM Communication with a camera via gigecam
3      %
4      properties (SetAccess = immutable)
5          name = 'NAME';      % an human readable identifier
6          format = '';        % format for gigecam function
7          model = '';          % manufacturer like in
                                gigecamlist().Model[i]
8          manufacturer = ''; % manufacturer like in
                                gigecamlist().Manufacturer[i]
9          ip = false;          % IP-Address
10     end
11
12     properties (Access = private)
13         handle = false;      % raw handle of the camera connection
14     end
15
16     methods
17         % constructor
18         function obj = class_gigecam(name, format, model,
19             manufacturer, list)
20             obj.name = name;
21             obj.model = model;
22             obj.manufacturer = manufacturer;
23             obj.format = format;
24
25             if nargin < 5
26                 list = gigecamlist();
27             end
28
29             % find camera index
30             index = -1;
31             [model_i, ~] = find(strcmp(list.Model, model));
32             [manuf_i, ~] = find(strcmp(list.Manufacturer,
33                 manufacturer));
34             i = intersect(manuf_i, model_i);
35
36             if size(i, 1) == 1
37                 index = i;
38             end;
39
40             % return if camera is not found
41             if index == -1
42                 error('Can''t find gigecam %s. (Manufacturer:
43                     %s, Model: %s)', name, manufacturer, model);
44             end
45
46             % initialize camera
47             obj.ip = list.IPAddress{index, 1};

```

```

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

```

```

88         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('gigecam 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)
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         res = obj.handle;
117
118         if nargin < 2
119             % preview as figure
120             hImage = image(zeros(res.Height, res.Width,
                1));
121         else
122             % preview on axes in GUI
123             hImage = image(zeros(res.Height, res.Width,
                1), 'Parent', axes);
124             axes.DataAspectRatio = [1, 1, 1];
125         end
126
127         if isa(adjust_function, 'function_handle')

```

```

128         setappdata(hImage,
129             'UpdatePreviewWindowFcn', adjust_function);
130     end
131     preview(obj.handle, hImage);
132
133     success = true;
134     catch e
135         warning('Exception while preview %s: %s',
136             obj.name, getReport(e));
137     end
138
139     % stop preview live image
140     function success = stoppreview(obj)
141         success = false;
142         if obj.handle == false
143             return;
144         end
145
146         try
147             closePreview(obj.handle);
148             success = true;
149         catch e
150             warning('Exception while preview %s: %s',
151                 obj.name, getReport(e));
152         end
153
154     function [success, img] = snapshot(obj,
155         adjust_function, axes)
156         success = false;
157
158         try
159             if nargin < 2
160                 adjust_function = false;
161             end
162
163             img = false;
164
165             obj.stoppreview();
166
167             % warn and return if connecting failes
168             if obj.connect() == false
169                 warning('Snapshot error: Can't connect to
170                     camera %s.', obj.name);
171                 return;
172             end

```

```
172         img = snapshot(obj.handle);
173
174         if isa(adjust_function, 'function_handle')
175             img = adjust_function(img);
176         end
177
178         if nargin > 2
179             imshow(img, 'Parent', axes);
180         end
181
182         success = true;
183     catch e
184         warning('Exception while snapshot %s: %s',
185             obj.name, getReport(e));
186     end
187
188     function delete(obj)
189         obj.close();
190     end
191 end
192
193 end
194
```