

---

# Grundlagen der Bildverarbeitung

## Übung 5 – Matlab GUIDE

---

Gurbandurdy Dovletov, M.Sc.

Raum: BC 414

Tel.: 0203-379-3583

Email: [gurbandurdy.dovletov@uni-due.de](mailto:gurbandurdy.dovletov@uni-due.de)

19. Mai 2022

# Matlab GUIDE

---

- *GUI* development environment
- Drag-and-drop Entwicklungsumgebung
- Für statische Benutzeroberflächen
- Code für Callbackfunktionen wird generiert
- Entsprechende Stellen müssen lediglich modifiziert werden

# Matlab GUIDE

---

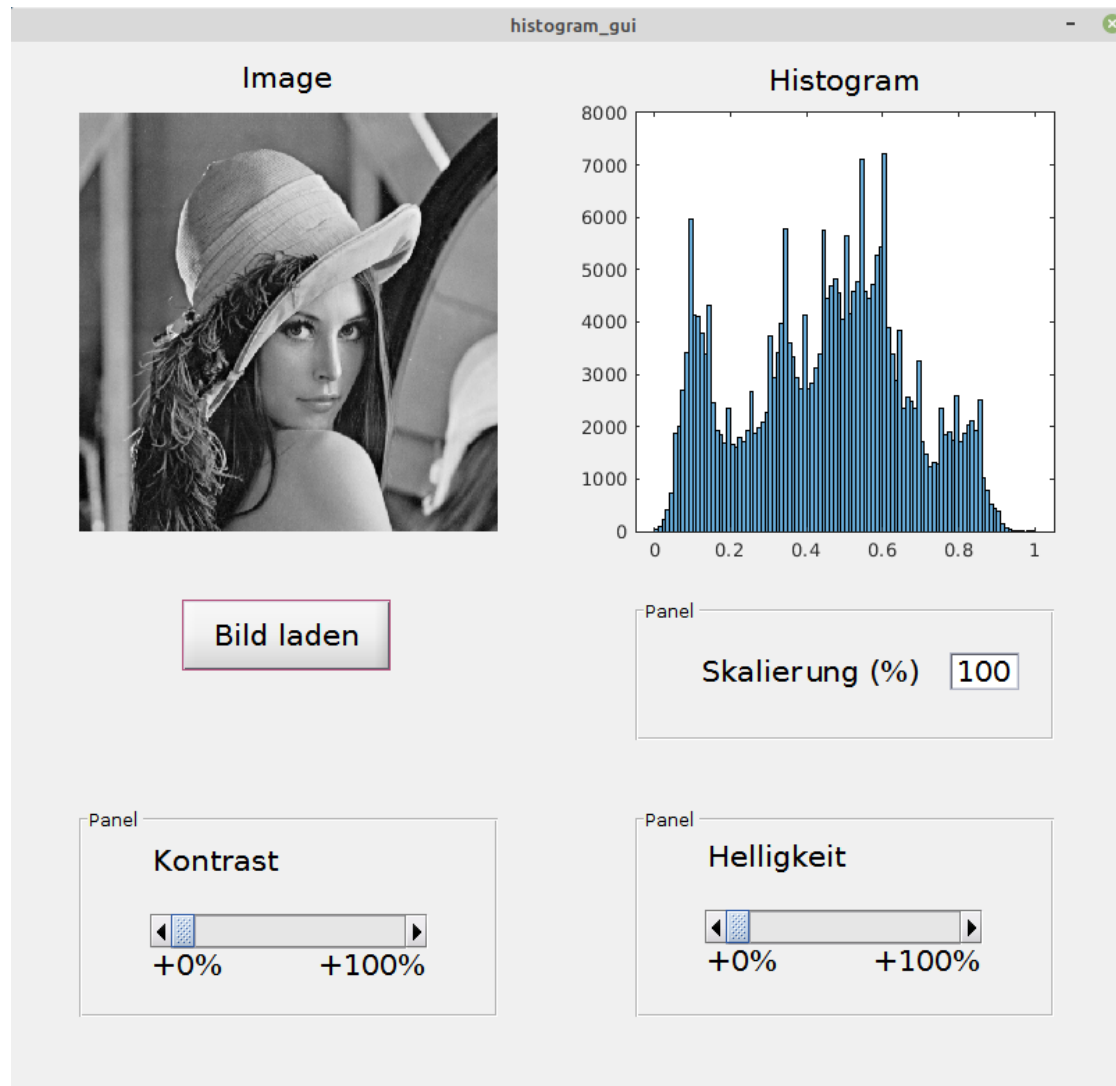
- Rufen Sie GUIDE auf, indem Sie in der Konsole `guide` ausführen
  - Blank GUI (Default)
- Erstellen Sie eine Benutzeroberfläche mit folgenden Komponenten:
  - 2 Axes (Koordinatensysteme)
  - 2 horizontalen Slider
  - 1 EditBox
  - 1 Button
- Speichern Sie diese *figure* ab.

# Matlab GUIDE

---

- Mit Rechtsklick auf GUI Komponenten können Sie im zugehörigen .m-file in den Codeabschnitt gelangen, der das Verhalten vom entsprechenden Callback bestimmt
  - Rechtsklick → View Callbacks → Callback
- Beim Button werden 3 Parameter übergeben:
  - `hObject` – Steuerungskomponente der GUI
  - `EventData` – reserviert für zukünftiges Matlab Release
  - **handles** – struct, das GUI relevante Daten speichert
  - Um `handles` zu aktualisieren, kann `guidata(hObject,handles)` ; verwendet werden

# Aufgabe 5 GUI



# Matlab GUIDE

---

- Datei können über die Funktion `uigetfile()` geladen werden
  - `[filename, pathname] = uigetfile({'*.jpg'; '*.png'; '*.bmp'; '*.tif'});`
  - `path = strcat(pathname, filename);`
  - `img = imread(path);`

# Matlab GUIDE

---

- Zeigen Sie geladenes Bild im Komponent axis1
  - `axes(handles.axes1);`
  - `imshow(img);`
- Da das Bild und entsprechendes Histogramm automatisch aktualisiert sein muss, muss man das Bild im `handles` Struktur speichern und danach `handles` aktualisieren.
  - `handles.img = img;`
  - `guidata(hObject, handles);`

# Matlab GUIDE

```
% --- Executes on button press in pushbutton1.  
function pushbutton1_Callback(hObject, eventdata, handles)  
% hObject    handle to pushbutton1 (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
  
[filename, pathname] = uigetfile({'*.jpg'; '*.png'; '*.bmp'; '*.tif'});  
path = strcat(pathname, filename);  
  
% read image  
img = imread(path);  
  
% Show image  
axes(handles.axes1);  
imshow(...);  
% Show histogram  
axes(handles.axes2);  
...;  
  
% save image in 'handles' and update them  
handles.img = img;  
guidata(hObject, handles);
```



# Matlab GUIDE

---

- Im Callback von Slider muss man zuerst das Bild von handles auslesen
  - `img = handles.img;`
- Jetzt kann das Bildes ändern, z.B.:
  - `img = img*10;`
- Danach muss man das Bilder und entsprechendes Histogramm aktualisieren (zeigen)
  - `axes(handles.axes1);`
  - `imshow(img);`

# Matlab GUIDE

```
% --- Executes on slider movement.
function slider1_Callback(hObject, eventdata, handles)
% hObject      handle to slider1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%        get(hObject,'Min') and get(hObject,'Max') to determine range of slider
percent = get(hObject,'Value');

% read image from 'handles'
img = handles.img;

% change contrast
img = ...;

% show image and histogram
...
```

# Matlab GUIDE

---

- Text in EditBox im Double Format auslesen
  - `str2double(get(hObject,'String'));`

# Matlab GUIDE

```
function edit1_Callback(hObject, eventdata, handles)
% hObject      handle to edit1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as a double

    percent = str2double(get(hObject,'String'))/100;

    % resize image
    img = ...

    % show image and histogram
    ...
```