МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(МОСКОВСКИЙ ПОЛИТЕХ)**

Транспортный факультет

Направление подготовки: 01.03.02 «Прикладная математика и информатика»

Отчет по Лабораторной работе №3

“Аффинные преобразования и гомография изображений”

Выполнил:

Студент:

Группа: 211-172

Тимошенко Ф.Р.

Проверил:

Кожухов Д.А.

Дата: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва 2024

## 1. Введение

Цель задания состояла в разработке программы для обработки изображений с использованием графического интерфейса пользователя (GUI). Программа должна была позволять пользователю выполнять различные операции над изображениями, такие как масштабирование, поворот, сдвиг, отражение и проекция на произвольную плоскость.

## 2. Реализация

Проект состоит из двух файлов. «window.py» и «main.py».

В качестве импортируемой библиотеки в первом файле используется PyQt6, октуда импортированы модули: QComboBox, QSlider, QApplication, QMainWindow, QLabel, QVBoxLayout, QWidget, QPushButton, QHBoxLayout, QLineEdit, Qt, QPixmap. В этом файле реализован класс ImageWindow, который наследуется от QMainWindow. Этот класс создает макет окна, с метками, слайдерами, изображениями, кнопками и полями ввода. Этот класс содержит следующие методы:

1. update\_images1(self, image\_path1) и update\_images2(self, image\_path2)

Эти методы позволяют отображать и обновлять изображения 1 и 2

1. update\_button(self)

Метод, в котором создаются все элементы, помимо изображений, которые отображаются на окне

1. show\_scale(self,i) и остальные методы начинающиеся на show\_

Управляют отображением элементов на окне

1. self.img\_hide()

Управляют отображением изображений на окне

Файл «main.py» является входной точкой в приложение. В этом файле реализован класс QMainWindow, который наследуется от ImageWindow.

Атрибутами класса являются:

1. self.initial\_path='' – путь выбранного изображения
2. self.scaleX=200 – величина маштабирования по оси Х
3. self.scaleY=200 – величина маштабирования по оси У
4. self.shiftX=0 – координата перемещения по оси Х
5. self.shiftY=0 – координата перемещения по оси У
6. self.selected\_oper=0 – индекс выбранной операции
7. self.centerx=50 –координаты по Х
8. self.centery=50 - координаты по У
9. self.angle=-90 – угол поворота

Методы класса:

1. on\_button1\_clicked(self) метод, который при нажатии на кнопку загружал изображение
2. on\_combo\_box\_changed(self, index) метод, который передает индекс выбранной операции пользователем и отображает нужные элементы на окне
3. on\_combo\_boxr\_changed(self, index) метод, который вызывает метод self.flip(index) и передает индекс операции выбранный пользователем.
4. onChanged\_x(self, value) и остальные методы начинающиеся на onChanged отвечают за присвоение конкретному атрибуту класса выбранного пользователем значение с помощью ползуенка
5. download\_img(self, i) загружает изображение и присвает атрибуту self.initial\_path значение пути выбранного в диалоговом окне файла
6. loadcv2(self, ini) создает переменную img, которая считывает в cv2 ранее загруженное изображение, и возвращает ее значение
7. saved\_and\_print\_process(self, img) сохраняет переданное изображение и отображает и отображает в макете изображения 2
8. scale(self) масштабирование изображения с параметрами, вводимыми пользователем. Применяется билинейная фильтрация
9. shift(self) осуществлять сдвиг изображения на заданное пользователем значение
10. input\_center(self,img:cv2) получает данные из полей для ввода и преобразует их в координаты для выбора точки на изображении
11. rotate(self) осуществляет поворот изображения относительно выбранной пользователем точки на заданный пользователем угол. Применяется билинейная фильтрация
12. flip(self, index) осуществляет отражение изображения одним из четырёх способов, не изменяя размеров изображения

## 3. Листинг кода:

«window.py»

|  |
| --- |
| 1 **from** PyQt6.QtWidgets **import** QComboBox, QSlider, QApplication, QMainWindow, QLabel, QVBoxLayout, QWidget, QPushButton, QHBoxLayout, QLineEdit  2 **from** PyQt6.QtGui **import** QPixmap  3 **from** PyQt6.QtCore **import** Qt  4  5 **class** **ImageWindow**(**QMainWindow**):  6 **def** **\_\_init\_\_**(self):  7 super().\_\_init\_\_()  8  9 self.setWindowTitle("Изображения")  10 self.setGeometry(50, 50, 400, 400)  11  12 self.central\_widget = QWidget()  13 self.setCentralWidget(self.central\_widget)  14  15 self.layout = QHBoxLayout()  16 self.central\_widget.setLayout(self.layout)  17  18  19 self.image\_layout = QVBoxLayout()  20 self.layout.addLayout(self.image\_layout)  21  22  23 self.image\_label1 = QLabel()  24 self.label1\_title = QLabel('До обработки')  25 self.image\_label2 = QLabel()  26 self.label2\_title = QLabel('После обработки')  27  28  29 #self.update\_images1("stuff/images/white.jpg")  30 #self.update\_images2("stuff/images/white.jpg")  31  32  33 self.image\_layout.addWidget(self.label1\_title, alignment=Qt.AlignmentFlag.AlignCenter)  34 self.image\_layout.addWidget(self.image\_label1, alignment=Qt.AlignmentFlag.AlignCenter)  35 self.label1\_title.hide()  36 self.image\_layout.addWidget(self.label2\_title, alignment=Qt.AlignmentFlag.AlignCenter)  37 self.image\_layout.addWidget(self.image\_label2, alignment=Qt.AlignmentFlag.AlignCenter)  38  39  40 self.button\_layout = QVBoxLayout()  41 self.layout.addLayout(self.button\_layout)  42 self.update\_button()  43  44 self.label1\_title.hide()  45 self.image\_label1.hide()  46 self.label2\_title.hide()  47 self.image\_label2.hide()  48 self.show()  49  50 **def** **update\_images1**(self, image\_path1):  51 self.label1\_title.show()  52 pixmap1 = QPixmap(image\_path1)  53  54  55 scaled\_pixmap1 = pixmap1.scaled(200, 200, Qt.AspectRatioMode.KeepAspectRatio)  56  57  58 self.image\_label1.setPixmap(scaled\_pixmap1)  59 self.image\_label1.show()  60  61 self.update()  62 **def** **img\_hide**(self):  63 self.label1\_title.hide()  64 self.image\_label1.hide()  65  66 **def** **update\_images2**(self, image\_path2):  67 self.label2\_title.show()  68  69 pixmap2 = QPixmap(image\_path2)  70  71  72 scaled\_pixmap2 = pixmap2.scaled(200,200, Qt.AspectRatioMode.KeepAspectRatio)  73  74  75 self.image\_label2.setPixmap(scaled\_pixmap2)  76 self.image\_label2.show()  77  78 self.update()  79  80  81 **def** **update\_button**(self):  82 self.button1 = QPushButton("Выбрать изображение")  83 self.button1.clicked.connect(self.on\_button1\_clicked)  84 self.button\_layout.addWidget(self.button1)  85  86 self.combo\_box = QComboBox()  87 self.combo\_box.addItem("Маштабирование")  88 self.combo\_box.addItem("Сдвиг")  89 self.combo\_box.addItem("Поворот")  90 self.combo\_box.addItem("Отражение")  91 self.combo\_box.addItem("Проекция")  92 self.combo\_box.currentIndexChanged.connect(self.on\_combo\_box\_changed)  93 self.button\_layout.addWidget(self.combo\_box)  94  95 #Ползунок на х  96 self.labelx\_title = QLabel('X')  97 self.button\_layout.addWidget(self.labelx\_title)  98 self.labelx\_title.hide()  99  100 self.labelx = QLabel('0', self)  101 self.labelx.setAlignment(Qt.AlignmentFlag.AlignCenter)  102 self.button\_layout.addWidget(self.labelx)  103 self.labelx.hide()  104  105 self.slider\_x = QSlider(Qt.Orientation.Horizontal, self)  106 self.slider\_x.setMinimum(1)  107 self.slider\_x.setMaximum(200)  108 self.slider\_x.setValue(200)  109 self.slider\_x.setTickPosition(QSlider.TickPosition.TicksBelow)  110 self.slider\_x.setTickInterval(1)  111 self.slider\_x.valueChanged.connect(self.onChanged\_x)  112 self.button\_layout.addWidget(self.slider\_x)  113 self.slider\_x.hide()  114  115 #Ползунок на y  116 self.labely\_title = QLabel('Y')  117 self.button\_layout.addWidget(self.labely\_title)  118 self.labely\_title.hide()  119  120 self.labely = QLabel('0', self)  121 self.labely.setAlignment(Qt.AlignmentFlag.AlignCenter)  122 self.button\_layout.addWidget(self.labely)  123 self.labely.hide()  124  125 self.slider\_y = QSlider(Qt.Orientation.Horizontal, self)  126 self.slider\_y.setMinimum(1)  127 self.slider\_y.setMaximum(200)  128 self.slider\_y.setValue(200)  129 self.slider\_y.setTickPosition(QSlider.TickPosition.TicksBelow)  130 self.slider\_y.setTickInterval(1)  131 self.slider\_y.valueChanged.connect(self.onChanged\_y)  132 self.button\_layout.addWidget(self.slider\_y)  133 self.slider\_y.hide()  134  135 #Ползунок на х shift  136 self.labelxs\_title = QLabel('X')  137 self.button\_layout.addWidget(self.labelxs\_title)  138 self.labelxs\_title.hide()  139  140 self.labelxs = QLabel('0', self)  141 self.labelxs.setAlignment(Qt.AlignmentFlag.AlignCenter)  142 self.button\_layout.addWidget(self.labelxs)  143 self.labelxs.hide()  144  145 self.slider\_xs = QSlider(Qt.Orientation.Horizontal, self)  146 self.slider\_xs.setMinimum(-1300)  147 self.slider\_xs.setMaximum(1300)  148 self.slider\_xs.setValue(0)  149 self.slider\_xs.setTickPosition(QSlider.TickPosition.TicksBelow)  150 self.slider\_xs.setTickInterval(1)  151 self.slider\_xs.valueChanged.connect(self.onChanged\_xs)  152 self.button\_layout.addWidget(self.slider\_xs)  153 self.slider\_xs.hide()  154  155 #Ползунок на y shift  156 self.labelys\_title = QLabel('Y')  157 self.button\_layout.addWidget(self.labelys\_title)  158 self.labelys\_title.hide()  159  160 self.labelys = QLabel('0', self)  161 self.labelys.setAlignment(Qt.AlignmentFlag.AlignCenter)  162 self.button\_layout.addWidget(self.labelys)  163 self.labelys.hide()  164  165 self.slider\_ys = QSlider(Qt.Orientation.Horizontal, self)  166 self.slider\_ys.setMinimum(-1300)  167 self.slider\_ys.setMaximum(1300)  168 self.slider\_ys.setValue(0)  169 self.slider\_ys.setTickPosition(QSlider.TickPosition.TicksBelow)  170 self.slider\_ys.setTickInterval(1)  171 self.slider\_ys.valueChanged.connect(self.onChanged\_ys)  172 self.button\_layout.addWidget(self.slider\_ys)  173 self.slider\_ys.hide()  174  175 #Поле ввода по x  176 self.labelxi\_title = QLabel('X%')  177 self.button\_layout.addWidget(self.labelxi\_title)  178 self.labelxi\_title.hide()  179  180 self.input\_fieldx = QLineEdit(self)  181 self.button\_layout.addWidget(self.input\_fieldx)  182 self.input\_fieldx.hide()  183 #Поле ввода по y  184 self.labelyi\_title = QLabel('Y%')  185 self.button\_layout.addWidget(self.labelyi\_title)  186 self.labelyi\_title.hide()  187  188 self.input\_fieldy = QLineEdit(self)  189 self.button\_layout.addWidget(self.input\_fieldy)  190 self.input\_fieldy.hide()  191  192 #Ползунок на angle  193 self.labela\_title = QLabel('Угол')  194 self.button\_layout.addWidget(self.labela\_title)  195 self.labela\_title.hide()  196  197 self.labela = QLabel('0', self)  198 self.labela.setAlignment(Qt.AlignmentFlag.AlignCenter)  199 self.button\_layout.addWidget(self.labela)  200 self.labela.hide()  201  202 self.slider\_a = QSlider(Qt.Orientation.Horizontal, self)  203 self.slider\_a.setMinimum(-180)  204 self.slider\_a.setMaximum(180)  205 self.slider\_a.setValue(0)  206 self.slider\_a.setTickPosition(QSlider.TickPosition.TicksBelow)  207 self.slider\_a.setTickInterval(1)  208 self.slider\_a.valueChanged.connect(self.onChanged\_a)  209 self.button\_layout.addWidget(self.slider\_a)  210 self.slider\_a.hide()  211  212 self.button\_apply = QPushButton("Применить")  213 self.button\_apply.clicked.connect(self.on\_button\_apply\_clicked)  214 self.button\_layout.addWidget(self.button\_apply)  215 self.button\_apply.hide()  216  217 #Комбобокс для отражения:  218 self.combo\_boxr = QComboBox()  219 self.combo\_boxr.addItem("По вертикали")  220 self.combo\_boxr.addItem("По горизонт  али")  221 self.combo\_boxr.addItem("По вертикале и по горизонтале")  222 self.combo\_boxr.addItem("Не отражать")  223 self.combo\_boxr.currentIndexChanged.connect(self.on\_combo\_boxr\_changed)  224 self.button\_layout.addWidget(self.combo\_boxr)  225 self.combo\_boxr.hide()  226  227 **def** **show\_scale**(self,i):  228 **if** i:  229 self.labelx\_title.show()  230 self.labelx.show()  231 self.slider\_x.show()  232 self.labely\_title.show()  233 self.labely.show()  234 self.slider\_y.show()  235 #self.button\_apply.hide()  236 **else**:  237 self.labelx\_title.hide()  238 self.labelx.hide()  239 self.slider\_x.hide()  240 self.labely\_title.hide()  241 self.labely.hide()  242 self.slider\_y.hide()  243 #self.button\_apply.show()  244 **def** **show\_shift**(self, i):  245 **if** i:  246 self.labelxs\_title.show()  247 self.labelxs.show()  248 self.slider\_xs.show()  249 self.labelys\_title.show()  250 self.labelys.show()  251 self.slider\_ys.show()  252 self.button\_apply.hide()  253 **else**:  254 self.labelxs\_title.hide()  255 self.labelxs.hide()  256 self.slider\_xs.hide()  257 self.labelys\_title.hide()  258 self.labelys.hide()  259 self.slider\_ys.hide()  260 #self.button\_apply.show()  261 **def** **show\_rotate**(self,i):  262 **if** i:  263 self.input\_fieldx.show()  264 self.input\_fieldy.show()  265 self.labelyi\_title.show()  266 self.labelxi\_title.show()  267 self.labela\_title.show()  268 self.labela.show()  269 self.slider\_a.show()  270 **else**:  271  272 self.input\_fieldx.hide()  273 self.input\_fieldy.hide()  274 self.labelxi\_title.hide()  275 self.labelyi\_title.hide()  276 self.labela\_title.hide()  277 self.labela.hide()  278 self.slider\_a.hide()  279 **def** **show\_flip**(self, i):  280 **if** i:  281 self.combo\_boxr.show()  282 **else**:  283 self.combo\_boxr.hide() |

«main.py»

|  |
| --- |
| 1. **import** cv2 2. **import** sys 3. **from** window **import** ImageWindow 4. **import** numpy **as** np 5. **from** PyQt6.QtWidgets **import** QApplication, QFileDialog, QVBoxLayout, QLineEdit, QLabel, QHBoxLayout, QWidget 6. save\_process\_path='stuff/saved/save\_proc.jpg' 7. **class** **Mywindow**(**ImageWindow**): 8. **def** **\_\_init\_\_**(self): 9. super().\_\_init\_\_() 10. self.initial\_path='' 11. self.scaleX=200 12. self.scaleY=200 13. self.shiftX=0 14. self.shiftY=0 15. self.selected\_oper=0 16. self.centerx=50 17. self.centery=50 18. self.angle=-90 19. #Кнопки 20. **def** **on\_button1\_clicked**(self): 21. self.img\_hide() 22. self.download\_img(1) 23. # def on\_button\_apply\_clicked(self): 24. # if self.selected\_oper==2: 25. # self.rotate() 26. #Комбобоксы 27. **def** **on\_combo\_box\_changed**(self, index): 28. self.selected\_oper=index 29. **if** index == 0: 30. self.show\_scale(1) 31. self.show\_shift(0) 32. self.show\_rotate(0) 33. self.show\_flip(0) 34. **elif** index == 1: 35. self.show\_scale(0) 36. self.show\_shift(1) 37. self.show\_rotate(0) 38. self.show\_flip(0) 39. **elif** index == 2: 40. self.show\_scale(0) 41. self.show\_shift(0) 42. self.show\_rotate(1) 43. self.show\_flip(0) 44. **elif** index == 3: 45. self.show\_scale(0) 46. self.show\_shift(0) 47. self.show\_rotate(0) 48. self.show\_flip(1) 49. **def** **on\_combo\_boxr\_changed**(self, index): 50. self.flip(index) 51. #Ползунки 52. **def** **onChanged\_x**(self, value): 53. self.scaleX=value 54. self.labelx.setText(str(value)) 55. **if** self.selected\_oper==0: 56. self.scale() 57. **def** **onChanged\_y**(self, value): 58. self.scaleY=value 59. self.labely.setText(str(value)) 60. **if** self.selected\_oper==0: 61. self.scale() 62. **def** **onChanged\_xs**(self, value): 63. self.shiftX=value 64. self.labelxs.setText(str(value)) 65. self.shift() 66. **def** **onChanged\_ys**(self, value): 67. self.shiftY=value 68. self.labelys.setText(str(value)) 69. self.shift() 70. **def** **onChanged\_a**(self, value): 71. self.angle=value 72. self.labela.setText(str(value)) 73. self.rotate() 74. **def** **download\_img**(self, i): 75. **try**: 76. self.initial\_path, \_ = QFileDialog.getOpenFileName(self, "Выберите изображение", "", "Изображения (\*.png \*.jpg \*.jpeg)") 77. **if** **not** self.initial\_path: 78. **raise** FileNotFoundError("Путь к изображению не был выбран.") 79. **if** i==1: 80. self.update\_images1(self.initial\_path) 81. **else**: 82. **raise** FileNotFoundError("Куда ты хочешь картинку?") 83. **except** Exception **as** e: 84. print("Ошибка при загрузке изображения", e) 85. **return** None 87. **def** **loadcv2**(self, ini): 88. **try**: 89. **if** **not** ini: 90. **raise** FileNotFoundError("Путь к изображению не был выбран.") 91. img = cv2.imread(ini) 92. **return** img 93. **except** Exception **as** e: 94. print("Ошибка при выполнении операции: ", e) 95. **return** None 96. **def** **saved\_and\_print\_process**(self, img): 97. cv2.imwrite(save\_process\_path, img) 98. self.update\_images2(save\_process\_path) 99. **def** **scale**(self): 100. img = self.loadcv2(self.initial\_path) 101. img = cv2.resize(img, (self.scaleY, self.scaleX), interpolation=cv2.INTER\_LINEAR) 102. self.saved\_and\_print\_process(img) 103. **def** **shift**(self): 104. img = self.loadcv2(self.initial\_path) 105. rows, cols = img.shape[:2] 106. M = np.float32([[1, 0, self.shiftX], [0, 1, self.shiftY]]) 107. img = cv2.warpAffine(img, M, (cols, rows)) 108. self.saved\_and\_print\_process(img) 109. **def** **input\_center**(self,img:cv2): 110. # if not img: 111. # raise ValueError("Не выбранно изображение") 112. **if** **not** (self.input\_fieldy.text() **and** self.input\_fieldx.text()): 113. **raise** ValueError("Не выбранна координата точки") 114. self.centery,self.centerx=int(self.input\_fieldy.text())/100,int(self.input\_fieldx.text())/100 115. rows, cols=img.shape[:2] 116. self.centery=int(rows\*self.centery) 117. self.centerx=int(cols\*self.centerx) 118. **def** **rotate**(self): 119. **try**: 120. img = self.loadcv2(self.initial\_path) 121. self.input\_center(img) 122. rows, cols = img.shape[:2] 123. #M = cv2.getRotationMatrix2D((self.centery,self.centerx), self.angle, 1) 124. M = cv2.getRotationMatrix2D((self.centery,self.centerx), self.angle, 1) 125. img = cv2.warpAffine(img, M, (cols, rows), flags=cv2.INTER\_LINEAR) 126. self.saved\_and\_print\_process(img) 127. **except** Exception **as** e: 128. print("Ошибка при rotate: ", e) 129. **return** None 130. **def** **flip**(self, index): 131. **try**: 132. img = self.loadcv2(self.initial\_path) 133. **if** index==0: 134. img=cv2.flip(img, 1) 135. **elif** index==1: 136. img=cv2.flip(img, 0) 137. **elif** index==2: 138. img=cv2.flip(img,-1) 139. **elif** index==3: 140. self.saved\_and\_print\_process(img) 141. **else**: 142. **raise**("Индекс выбран не правильно") 143. self.saved\_and\_print\_process(img) 144. **except** Exception **as** e: 145. print("Ошибка при reflection: ", e) 146. **return** None 147. **if** \_\_name\_\_ == "\_\_main\_\_": 148. app = QApplication(sys.argv) 149. window = Mywindow() 150. sys.exit(app.exec()) |