**Завдання**

**Завдання**:

1. Ознайомитися з теоретичними відомостями реалізації алгоритмів сегментації медичних зображень, що засновані на пошуку границь.
2. Розробити програмний додаток для завантаження двох зображень комплементарної природи в форматі DICOM та виконання мультимодальної їх візуалізації.
3. Створити події від клавіатури, при обробці яких можна окремо виконати мультимодальну візуалізацію двох зображень комплементарної природи та почергово відновити оригінальне відображення завантажених томографічних зрізів.
4. Виконати мультимодальну візуалізацію двох зображень комплементарної природи (методом мультимодального рендерінгу використати відповідно до заданого в варіанті).
5. Скласти і захистити звіт по роботі.

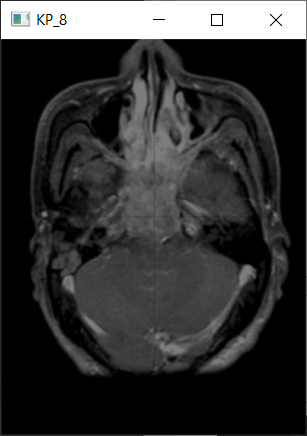
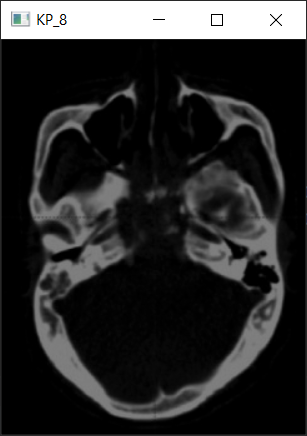
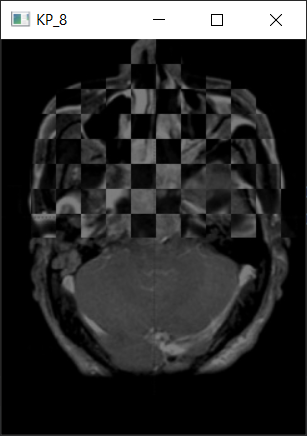
| **Номер варіанту** | **Метод мультимодального рендерінгу для проведення візуалізації** | **Зона перетворення** |
| --- | --- | --- |
| 8 | Метод подвоєння зображення | Верхня половина площі зображення |

**Лістинг программи**

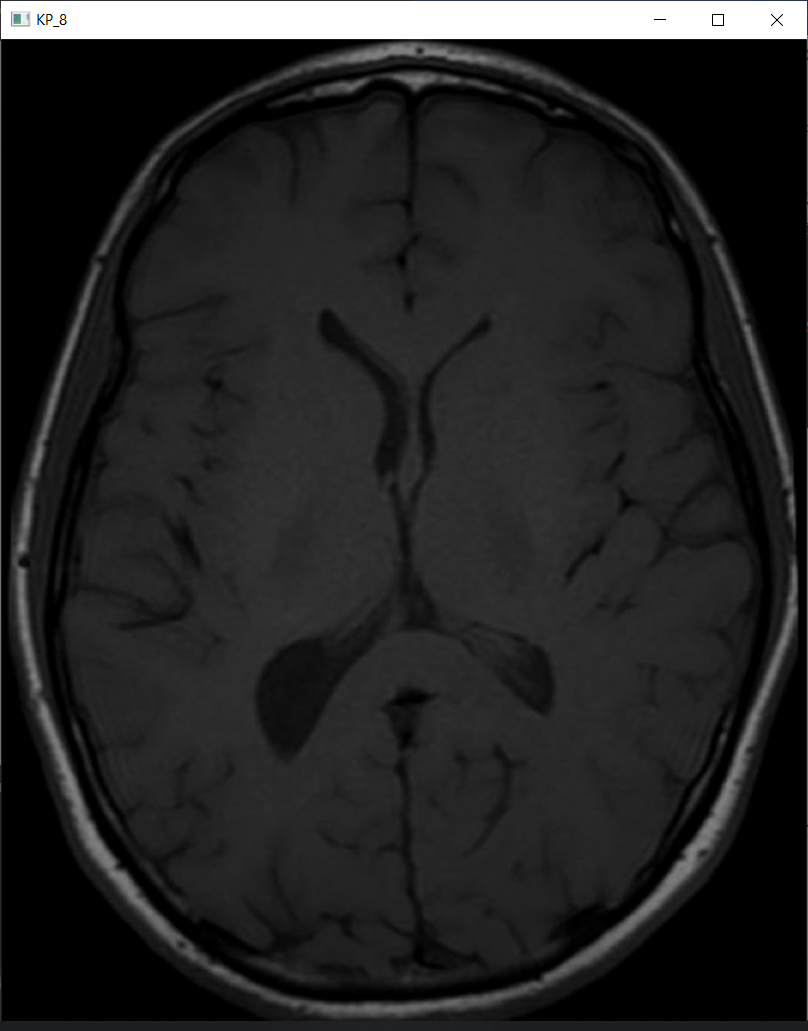
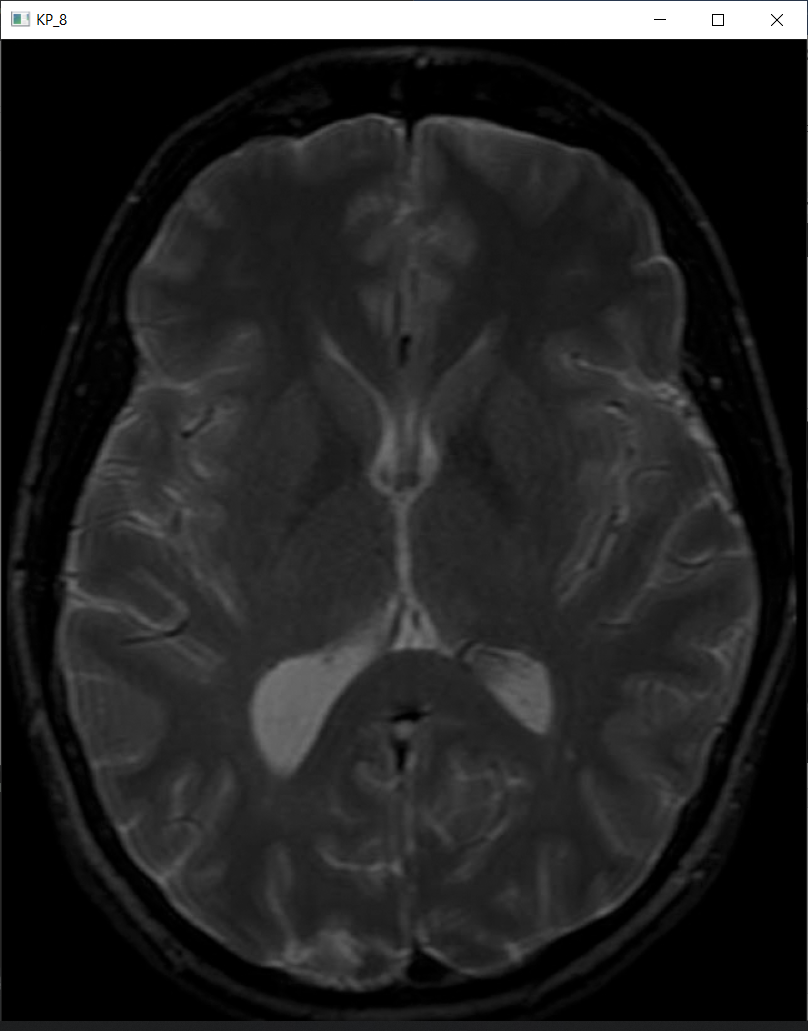
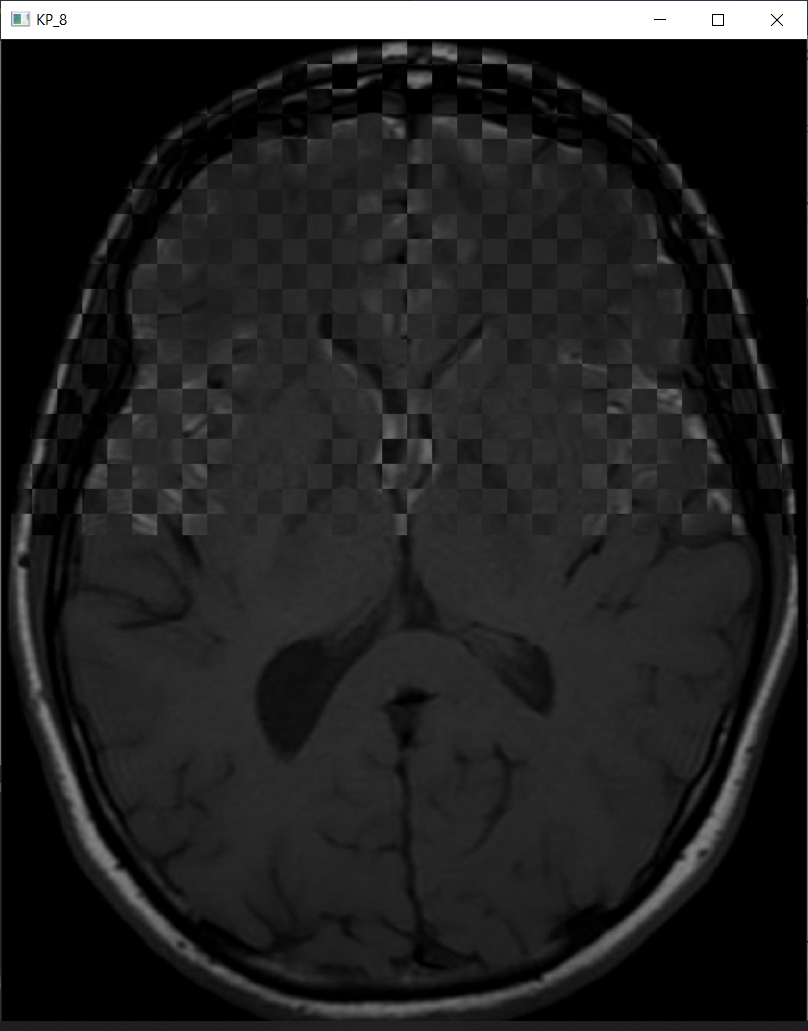
from OpenGL.GL import \*  
from OpenGL.GLU import \*  
from OpenGL.GLUT import \*  
import pydicom  
import numpy as np  
  
path\_file1 = "3-t1.dcm"  
path\_file2 = "3-t2.dcm"  
  
  
class Image:  
 def \_\_init\_\_(self):  
 self.image1 = pydicom.read\_file(path\_file1)  
 self.image2 = pydicom.read\_file(path\_file2)  
  
 self.width1, self.height1 = self.image1[0x280011].value, self.image1[0x280010].value  
 self.width2, self.height2 = self.image2[0x280011].value, self.image2[0x280010].value  
  
 self.pixels = []  
  
 @staticmethod  
 def max\_brightness(image):  
 image\_type = np.dtype('int' + str(image[0x280100].value))  
 return np.iinfo(image\_type).max  
  
 def colored\_image(self, image):  
 output = []  
 image\_pixels = image.pixel\_array  
  
 p\_max = np.amax(image\_pixels)  
 p\_min = np.amin(image\_pixels)  
 delta = p\_max - p\_min  
 max\_brightness = self.max\_brightness(image)  
  
 for row in image\_pixels:  
 new\_row = []  
 for pixel in row:  
 new\_pixel = int((float(pixel) / delta) \* max\_brightness)  
 if new\_pixel < 0:  
 new\_pixel = 0  
 elif new\_pixel > max\_brightness:  
 new\_pixel = max\_brightness  
 new\_row.append([new\_pixel, new\_pixel, new\_pixel, max\_brightness])  
 output.append(new\_row)  
  
 return output  
  
 def init(self):  
 glEnable(GL\_TEXTURE\_2D)  
 glBindTexture(GL\_TEXTURE\_2D, glGenTextures(1))  
  
 glTexImage2D(GL\_TEXTURE\_2D, 0, GL\_RGBA, self.width1, self.height1, 0, GL\_RGBA, GL\_UNSIGNED\_BYTE, self.pixels)  
  
 glTexParameterf(GL\_TEXTURE\_2D, GL\_TEXTURE\_MAG\_FILTER, GL\_NEAREST)  
 glTexParameterf(GL\_TEXTURE\_2D, GL\_TEXTURE\_MIN\_FILTER, GL\_NEAREST)  
 glTexParameter(GL\_TEXTURE\_2D, GL\_TEXTURE\_WRAP\_S, GL\_CLAMP\_TO\_BORDER)  
 glTexParameter(GL\_TEXTURE\_2D, GL\_TEXTURE\_WRAP\_T, GL\_CLAMP\_TO\_BORDER)  
  
 glClearColor(0, 0, 0, 0)  
 glMatrixMode(GL\_PROJECTION)  
 glLoadIdentity()  
  
 gluOrtho2D(-self.width2 / 2, self.width2 / 2, -self.height2 / 2, self.height2 / 2)  
  
 def display(self):  
 glClearColor(0.0, 0.0, 0.0, 0.0)  
 glClear(GL\_COLOR\_BUFFER\_BIT)  
 self.draw\_texture()  
  
 def draw\_texture(self):  
 glBegin(GL\_QUADS)  
 glTexCoord2d(0.0, 0.0)  
 glVertex2d(self.width2 / 2, self.height2 / 2)  
 glTexCoord2d(0.0, 1.0)  
 glVertex2d(self.width2 / 2, -self.height2 / 2)  
 glTexCoord2d(1.0, 1.0)  
 glVertex2d(-self.width2 / 2, -self.height2 / 2)  
 glTexCoord2d(1.0, 0.0)  
 glVertex2d(-self.width2 / 2, self.height2 / 2)  
 glEnd()  
 glFlush()  
  
 def multimodal\_image(self):  
 output = []  
 size = 40  
  
 image1 = self.colored\_image(self.image1)  
 image2 = self.colored\_image(self.image2)  
  
 num\_i = 0  
 for i in range(len(image1)):  
 new\_row = []  
 if num\_i == size:  
 num\_i = 0  
 num\_j = 0  
 for j in range(len(image1[0])):  
 new\_pixel = image1[i][j]  
 if num\_j == size:  
 num\_j = 0  
 if i < len(image1) / 2:  
 if ((0 <= num\_i < (size / 2)) and ((size / 2) <= num\_j < size)) or (((size / 2) <= num\_i < size) and (0 <= num\_j < (size / 2))):  
 new\_pixel = image2[i][j]  
 new\_row.append([new\_pixel[0], new\_pixel[1], new\_pixel[2], new\_pixel[3]])  
 num\_j += 1  
 output.append(new\_row)  
 num\_i += 1  
  
 return output  
  
 def keyboard\_func(self, my\_key, x, y):  
 key = unicode(my\_key, errors='ignore')  
 if key == '1':  
 self.pixels = self.colored\_image(self.image1)  
 if key == '2':  
 self.pixels = self.colored\_image(self.image2)  
 if key == 'm':  
 self.pixels = self.multimodal\_image()  
  
 glTexImage2D(GL\_TEXTURE\_2D, 0, GL\_RGBA, self.width1, self.height1, 0, GL\_RGBA, GL\_UNSIGNED\_BYTE, self.pixels)  
 self.display()  
  
  
def init\_window(width, height):  
 glutInitWindowSize(width, height)  
 glutInitWindowPosition((glutGet(GLUT\_SCREEN\_WIDTH) - width) // 2, (glutGet(GLUT\_SCREEN\_HEIGHT) - height) // 2)  
 glutCreateWindow('KP\_8')  
  
  
def main():  
 glutInit()  
 glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGBA)  
 file = Image()  
 init\_window(file.width2, file.height2)  
 file.init()  
 glutDisplayFunc(file.display)  
 glutKeyboardFunc(file.keyboard\_func)  
 glutMainLoop()  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()

Результат

*Для Var1*

*Для Var2*

**Контрольні запитання**

1. *Визначити два основні напрямки комбінації (об’єднання інформації) зображень різних модальностей.*

Використання мультимодальних систем (об’єднання різних діагностичних пристроїв в один)

Створення мультимодальних зображень (суміщення різних зрізів від окремих приладів при окремому їх проведенні)

1. *Які три проблеми необхідно вирішити для отримання мультимодальних зображень за допомогою суміщення наборів даних від різних (окремо проведених) досліджень?*

* Проблеми, які пов’язані з різнорідністю файлових структур (вирішено за допомогою використання стандарту DICOM)
* Проблеми суміщення одних і тих самих анатомічних зон (рішення – реєстрація зображення)
* Проблеми візуалізації даних (рішення – мультимодальний рендерінг)

1. *Перелічити переваги використання мультимодальних зображень.*

* Виділення осередків, що не визначаються одним методом дослідження
* Встановлення фізіологічної ролі сумнівних на анатомічних зображеннях об’єктів
* Виявлення запальних процесів
* Оцінку резидуальної активності після хірургічного втручання

1. *Які критерії використовують для класифікації процесу реєстрації мультимодальних зображень?*

* Розмірність зображення (2D-до-2D, 3D-до-3D, 2D-до-3D)
* Ступені свободи геометричних перетворень (жорсткі, афінні, нелінійні)
* Принцип реєстрації (заснований на орієнтації, або сегментації, або подібності вокселів)
* Зона перетворення для реєстрації (глобальна, місцева)
* Рівень автоматизації (автоматичний, напівавтоматичний)

1. *Яка розмірність для реєстрації зображення, що отримані за допомогою:*
2. *комп'ютерної (рентгенівської) томографії та магнітно-резонансної томографії;*

2D-до-3D

1. *магнітно-резонансної томографії та рентгенівської ангіографії;*

2D-до-2D

1. *тривимірного зображення за даними комп’ютерної (рентгенівської) томографії та рентгенівської ангіографії?*

3D-до-3D

1. *Як вирішується головна проблема мультимодального рендерінгу?*

За допомогою поєднання декількох можливих методів рендерінгу.

1. *Які колірні простори слід використовувати під час проведення сумарного розподілення кольорів для мультимодального рендерінгу даних анатомічних та функціональних зображень?*

Для анатомічних зображень в більшості випадків використовують сірошкальну палітру, а для функціональних – контрастну кольорову палітру (інтенсивність кольору асоціюється з інтенсивністю змін).

1. *Основна ідея мультимодального рендерінгу з використанням методу подвоєння зображення.*

Основна ідея це збереження кольорової гами зображення кожної модальності (нове зображення створюється з вдвічі більшими, ніж початковими, розмірами матриці, кожен другий піксель містить інформацію про анатомічні модальності, а інші – функціональні).

1. *Які недоліки використання мультимодального рендерінгу методом шахової дошки?*

При використанні кольорових компонентів можуть бути обмеження, оскільки при невеликих розмірах квадратів суміжні квадрати можуть значно погіршити візуальний ефект кольорового подання.

1. *Як зберігається зображення з альфа-каналом відповідно до стандарту* ***DICOM****?*

За допомогою двох архітектур альфа каналу це:

* Little-endian – формат “ARGB”
* Big-endian – формат “RGBA”