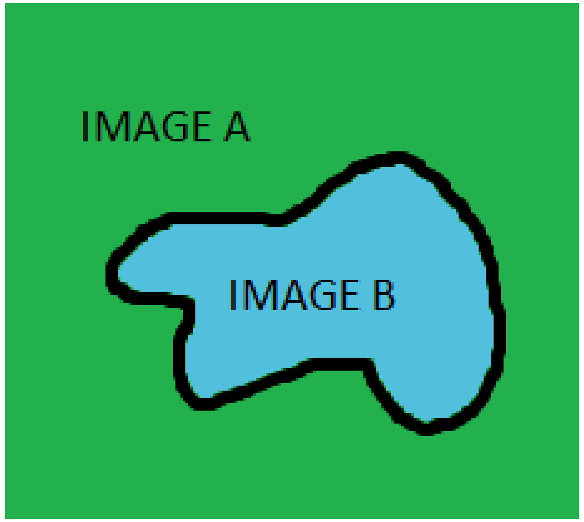
Применение вариационного метода для блендинга изображений (*Using the Variation Method to Image Blending*)

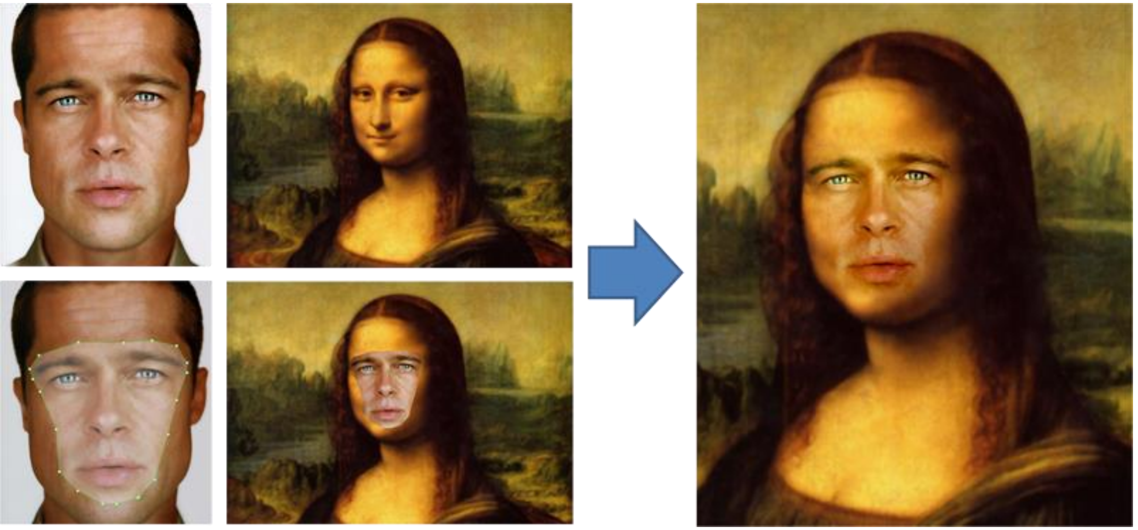
Владислав Хайдуров

Постановка задачи

Пусть у нас имеются изображения ***A***, на которое *в определенном месте* накладывается изображение ***B***.

Вставку изображения ***B*** нужно выполнить так, чтобы градиент изображения ***A*** в области изображения ***B*** равнялся градиенту начального изображения ***B***.





Актуальные работы по направлению

1. J. Matías Di Martino, Gabriele Facciolo, and Enric Meinhardt-Llopis, Poisson Image Editing, Image Processing On Line, 6 (2016), pp. 300–325. https://doi.org/10.5201/ipol.2016.163
2. Nordin Saad, A'qilah Ahmad Dahalan, Azali Saudi. Modified Poisson compositing technique on skewed grid. AIMS Mathematics, 2022, Volume 7, Issue 2: 2176-2194. doi: 10.3934/math.2022124.
3. Fouad Mohammad Salama, Nur Nadiah Abd Hamid, Umair Ali, Norhashidah Hj. Mohd Ali. Fast hybrid explicit group methods for solving 2D fractional advection-diffusion equation. AIMS Mathematics, 2022, Volume 7, Issue 9: 15854-15880. doi: 10.3934/math.2022868

Подобные задачи

- *Задача восстановления изображения по векторному полю градиентов [1–3].*
- **<u>Задача бесшовной вставки (блендинг) изображения [1].</u>**
- *Задача восстановления изображения со смешиванием градиентов изображений [1–2].*
- *Задача редактирования области изображения [1–3].*
- *Задача восстановления части изображения [1–3].*
- *Задача замены части изображения на похожую [1–3].*

Подход к решению задачи

Проблемы:

- при численном решении уравнения Пуассона образовываются большие СЛАУ с разреженной структурой;
- нужно много вычислительных ресурсов.

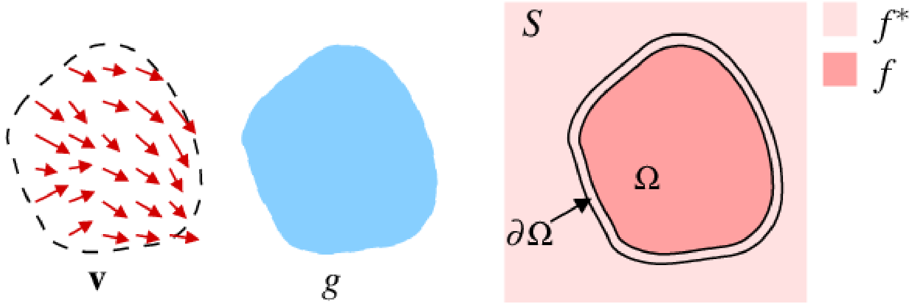
Возможные решения:

- использование многосеточных методов решения СЛАУ;
- распараллеливание процессов в ходе решения СЛАУ;
- использование вспомогательных готовых библиотек (желательно с открытым кодом);
- комбинация существующих решений.

Математический инструмент

Найти функцию $f(x,y)$ для которой функціонал имеет глобальний минимум:

$$J(f) = \iint_{\Omega} F(\nabla f, v)\,dxdy \to min,$$



$$F(\nabla f, v) = \|\nabla f, v\| = \left(\frac{\partial f}{\partial x} - v_x\right)^2 + \left(\frac{\partial f}{\partial y} - v_y\right)^2.$$

$$\frac{\partial F}{\partial f} - \frac{d}{dx}\frac{\partial F}{\partial f'_x} - \frac{d}{dy}\frac{\partial F}{\partial f'_y} = 0$$ – уравнение Эйлера – Лагранжа.

$$2\left(\frac{\partial^2 f}{\partial x^2} - \frac{\partial v_x}{\partial x}\right) + 2\left(\frac{\partial^2 f}{\partial y^2} - \frac{\partial v_y}{\partial y}\right) = 0, \qquad \boldsymbol{\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y}}.$$

Математический инструмент

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y}.$$

Граничные условия вида:

а) Дирихле (первого рода):

$$f\Big|_{\partial\Omega} = f^*\Big|_{\partial\Omega};$$

б) Неймана (второго рода):

$$f'\Big|_{\partial\Omega} = g^*\Big|_{\partial\Omega};$$

в) Робіна (третьего рода):

$$\alpha f\Big|_{\partial\Omega} + \beta f'\Big|_{\partial\Omega} = \varphi^*\Big|_{\partial\Omega}.$$

Бесшовная вставка. Математический пример

Найти $\boldsymbol{\varphi}(\boldsymbol{x})$ для которой

$$J(\varphi', v) = \int_{\frac{1}{3}}^{\frac{2}{3}} F(\varphi', v)dx = \int_{\frac{1}{3}}^{\frac{2}{3}} [(\varphi' - v)^2]dx \to min,$$

где $v(x) = \pi \cos \pi x$, $\varphi(1/3) = f(1/3)$, $\varphi(2/3) = f(2/3)$,

$$f(x) = 2 - \cos 2\pi x^2, \qquad x \in [0; 1].$$

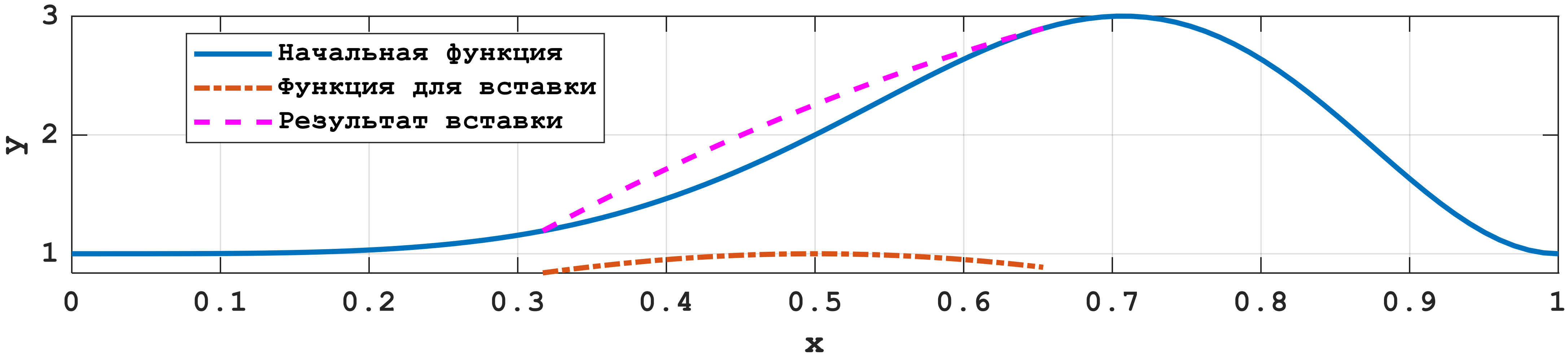
Решение:

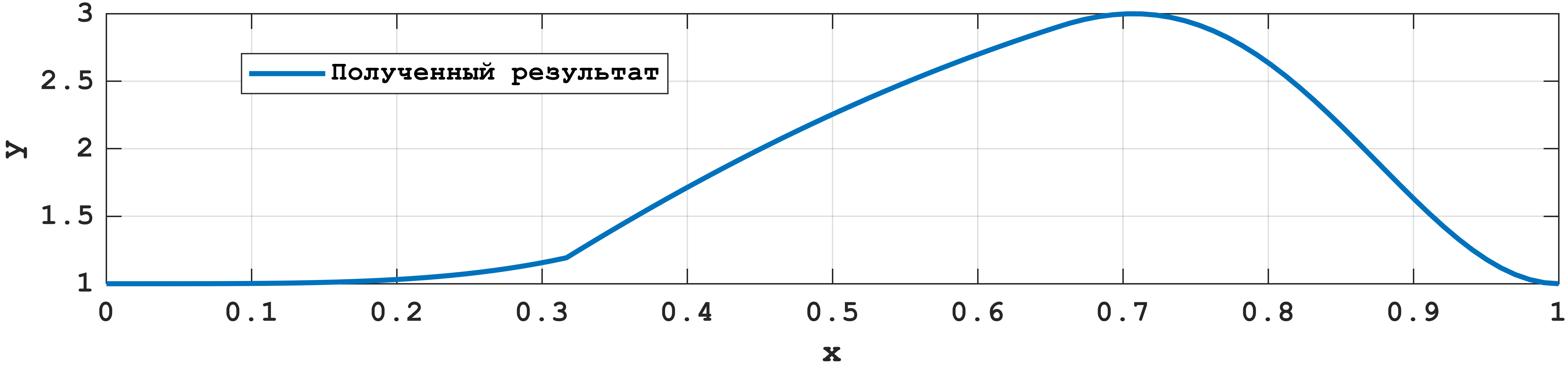
$$2\left(\frac{d^2\varphi}{dx^2} - \frac{dv}{dx}\right) = 0 \Rightarrow \frac{d^2\varphi}{dx^2} = \frac{dv}{dx}, \qquad \frac{dv}{dx} = (\pi \cos \pi x)' = -\pi^2 \sin \pi x.$$

Окончательно имеем:

$$\frac{d^2\varphi}{dx^2} = -\pi^2 \sin \pi x, \qquad \varphi\left(\frac{1}{3}\right) = f\left(\frac{1}{3}\right), \varphi\left(\frac{2}{3}\right) = f\left(\frac{2}{3}\right).$$

Бесшовная вставка. Математический пример





Бесшовная вставка. Математический пример 2

Знайти $\boldsymbol{\varphi}(\boldsymbol{x},\boldsymbol{y})$ для якої

$$J(\nabla\varphi,v)=\int_{\frac{1}{3}}^{\frac{2}{3}}\int_{\frac{1}{3}}^{\frac{2}{3}}F(\nabla\varphi,v)dxdy=\int_{\frac{1}{3}}^{\frac{2}{3}}\int_{\frac{1}{3}}^{\frac{2}{3}}\left[(\varphi_x'-v_x)^2+\left(\varphi_y'-v_y\right)^2\right]dxdy\to min,$$

де $v_x=g_x'$, $v_y=g_y'$, $g(x,y)=\sin\big(\pi(x-0.5)(y-0.5)\big)$

з граничними умовами:

$$\varphi(1/3,y)=f(1/3,y),\qquad \varphi(2/3,y)=f(2/3,y),$$
$$\varphi(x,1/3)=f(x,1/3),\qquad \varphi(x,2/3)=f(x,2/3).$$
$$f(x,y)=sin\big(\pi(x-0.5)(y-0.5)\big).$$

Бесшовная вставка. Математический пример 2

Решение:

$$v_x(x,y) = \frac{\partial g}{\partial x} = \pi(y-0.5)\,cos\big(\pi(x-0.5)(y-0.5)\big), \qquad v_y(x,y) = \frac{\partial g}{\partial y} = \pi(x-0.5)\,cos\big(\pi(x-0.5)(y-0.5)\big).$$

Используя уравнение Эйлера-Лагранжа, получим

$$\frac{\partial^2\varphi}{\partial x^2} + \frac{\partial^2\varphi}{\partial y^2} = \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y}.$$

$$\frac{\partial v_x}{\partial x} = -\pi^2(y-0.5)^2\,sin\big(\pi(x-0.5)(y-0.5)\big), \qquad \frac{\partial v_y}{\partial y} = -\pi^2(y-0.5)^2\,sin\big(\pi(x-0.5)(y-0.5)\big).$$

Окончательно имеем:

$$\frac{\partial^2\varphi}{\partial x^2} + \frac{\partial^2\varphi}{\partial y^2} = -\pi^2\big((x-0.5)^2 + (y-0.5)^2\big)\,sin\big(\pi(x-0.5)(y-0.5)\big),$$

$$(x,y) \in \left[\frac{1}{3};\frac{2}{3}\right]^2, \qquad \varphi\left(\frac{1}{3},y\right) = f\left(\frac{1}{3},y\right), \qquad \varphi\left(\frac{2}{3},y\right) = f\left(\frac{2}{3},y\right),$$

$$\varphi\left(x,\frac{1}{3}\right) = f\left(x,\frac{1}{3}\right), \qquad \varphi\left(x,\frac{2}{3}\right) = f\left(x,\frac{2}{3}\right).$$

Бесшовная вставка. Математический пример 2

$$\frac{\partial v_x}{\partial x} = -\pi^2 (y-0.5)^2 \, sin\big(\pi(x-0.5)(y-0.5)\big),$$
$$\frac{\partial v_y}{\partial y} = -\pi^2 (y-0.5)^2 \, sin\big(\pi(x-0.5)(y-0.5)\big).$$

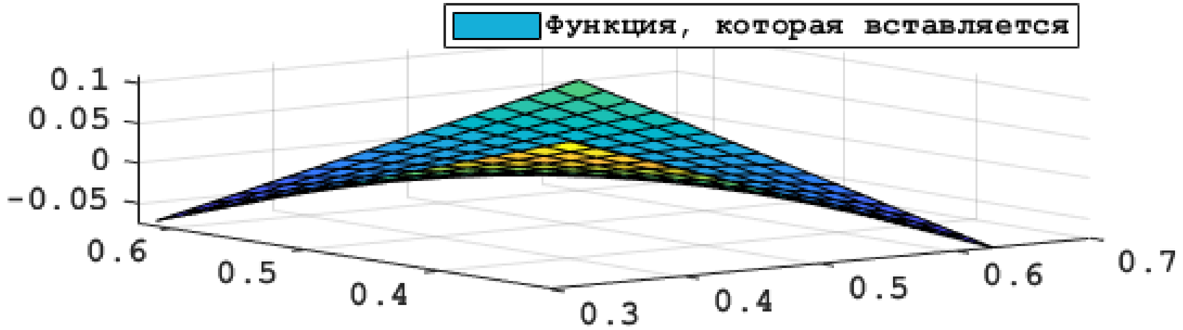
Остаточно маємо:

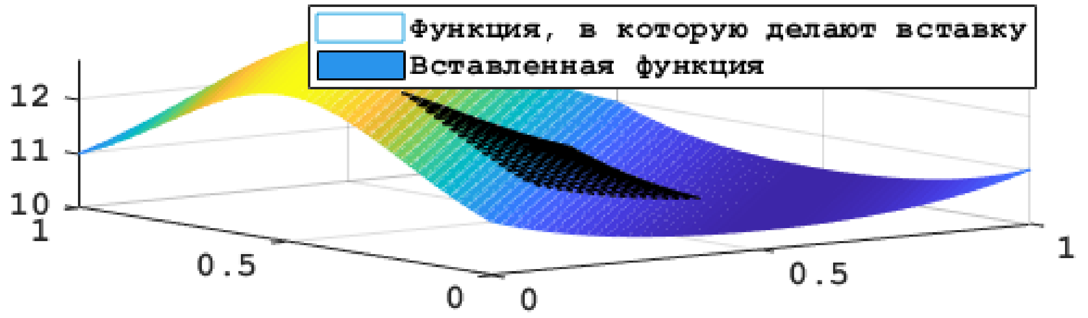
$$\frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2} = -\pi^2\big((x-0.5)^2 + (y-0.5)^2\big) \, sin\big(\pi(x-0.5)(y-0.5)\big),$$

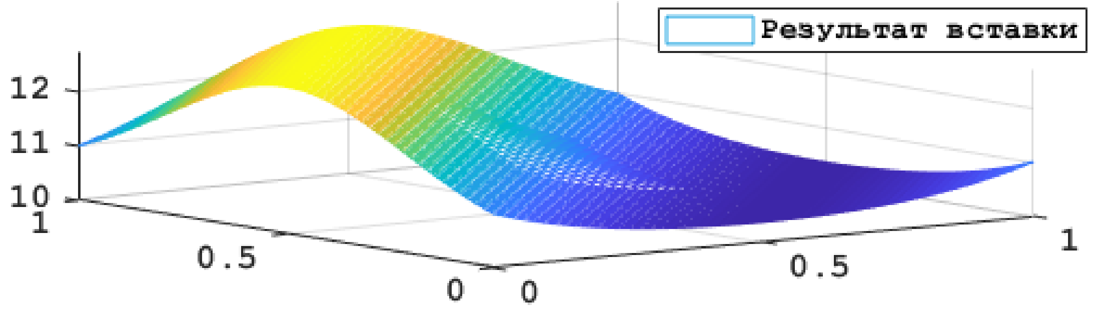
$$(x,y) \in \left[\frac{1}{3};\frac{2}{3}\right]^2, \qquad \varphi\left(\frac{1}{3},y\right) = f\left(\frac{1}{3},y\right), \qquad \varphi\left(\frac{2}{3},y\right) = f\left(\frac{2}{3},y\right),$$

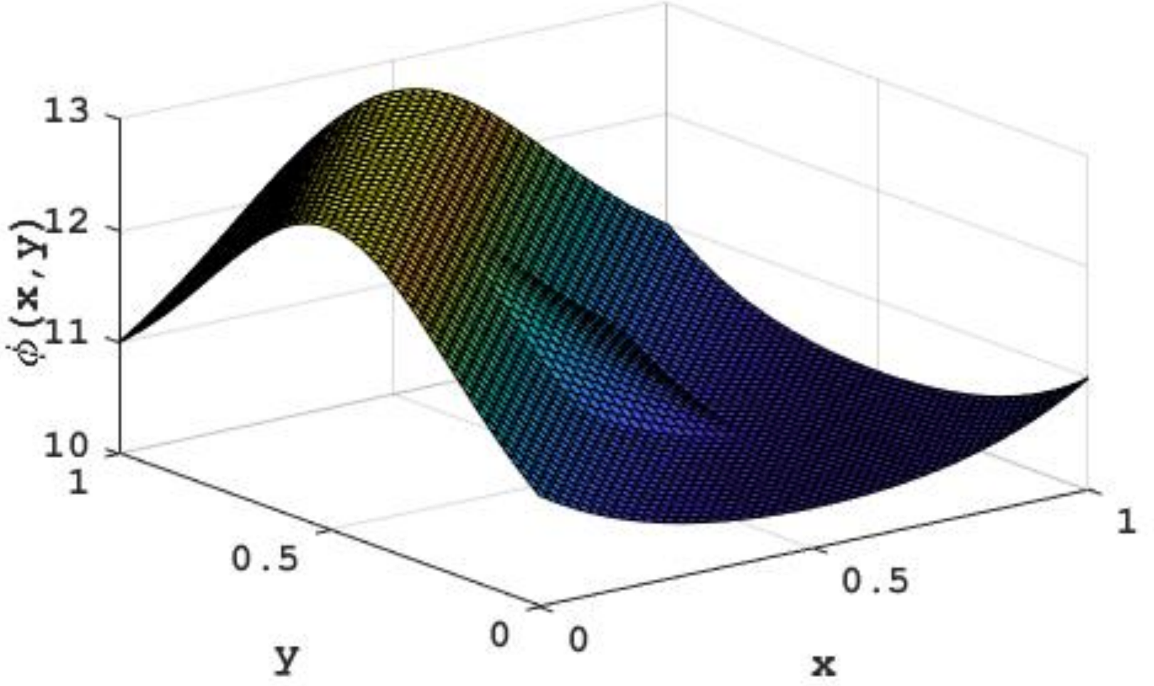
$$\varphi\left(x,\frac{1}{3}\right) = f\left(x,\frac{1}{3}\right), \qquad \varphi\left(x,\frac{2}{3}\right) = f\left(x,\frac{2}{3}\right).$$

Бесшовная вставка. Математический пример 2









Использованные технологии

- OpenCV
- python™
- SciPy
- NumPy
- matplotlib

Основные фрагменты кода

```
def CreateSparseMatrix(nx,ny):
     # задаем шаблон коэффициентов из единиц
     Ones = np.ones((nx*ny,))
     # формируем ленты матрицы на основании ф-ра Лапласа
     data = np.array([Ones, Ones, -4*Ones, Ones, Ones])
     # фомируем местоположения лент в матрице
     diags = np.array([-ny, -1, 0, 1, ny])
     # сформировали матрицу, сделали доступ к ней
     A = spdiags(data, diags, nx*ny, nx*ny);
     A = sp.csr_matrix(A)
     # удаляем лишние коэффициенты в матрице для 2-х лент
     for i in range(ny,N,ny):  A[i,i-1]=A[i-1,i]=0
```

Основные фрагменты кода


```
# Вторая основная функция для решения поставленной задачи
def FormRightPart(imgInsert,imgContainer):
    imgInsert = np.double(imgInsert) # переходим к double для попиксельной обработки
    imgContainer = np.double(imgContainer)
    # Правая часть ур.Пуассона
    tmp = imgInsert[:-2,1:-1,:] - 2*imgInsert[1:-1,1:-1,:] + imgInsert[2:,1:-1,:]
    tmp += imgInsert[1:-1,:-2,:] - 2*imgInsert[1:-1,1:-1,:] + imgInsert[1:-1,2:,:]

    # Учет граничных условий (тут подтягиваем значения с рамки
    # в контейнере для вставки изображения)
    tmp[:,0,:] -= imgContainer[1:-1,0,:]    # низ
    tmp[:,-1,:] -= imgContainer[1:-1,-1,:]  # верх
    tmp[0,:,:] -= imgContainer[0,1:-1,:]    # лево
    tmp[-1,:,:] -= imgContainer[-1,1:-1,:]  # право

    return tmp # возвращаем правую часть пока что в матричном виде, а не в векторном
```

Основные фрагменты кода


```
# функция преобразования правой части в векторную (переход от матрицы) и решение
СЛАУ
def GetSolution(A,B,nx,ny):
    Sol = np.zeros((nx,ny,channel)) # выделяем память под матрицу решения
    for i in range(channel): # решаем уравнение Пуассона для всех каналов
        b = np.reshape(B[:,:,i],(N,1)) # взяли канали и преобразовали его в вектор
        start = time.time() # стартовая точка отсчета времени решения СЛАУ
        X = spsolve(A,b) # Решили СЛАУ, используя scipy
        finish = time.time() # конец по времени для решения СЛАУ
        Sol[:,:,i] = np.reshape(X,(nx,ny)) # вектор-решение обратно в матрицу
        Sol[Sol>255] = 255; Sol[Sol<0] = 0 # обрабатываем выход за пределы для RGB
    Sol = np.uint8(Sol) # преобразование в uint8
    return Sol
```

Результаты работы кода

Пример входных данных

Изображение-контейнер

Изображение-шаблон (для вставки)

Результаты работы кода

<u>Результат – входные данные</u> (***результат решения задачи***)

Общие выводы по результатам

- В проекте реализована основная задача – задача восстановления изображения по полю его градиентов.
- На основании полученных результатов решения базовой задачи (восстановления изображения по полю градиентов) решена чуть сложнее задача – задача бесшовной вставки одного изображения в другое на основании вариационного принципа, который сводится к решению краевой задачи для уравнения Пуассона.
- Разработанный программный код протестирован на разных полноцветных изображениях. Программа работает корректно.
- Код работает быстро, учитывая развитую библиотеку SCIPY, которая была использована для решения СЛАУ с разреженной матрицей ленточной структуры.

Рекомендации по улучшению

Для получения более эффективного и эффектного результата можно применить:

- метод смешивания градиентов;
- делать вставку изображения без фона или с минимальным фоном в изображении;
- распараллелить (для получение более быстрого результата для больших изображений для блендинга;
- комбинировать (в зависимости от задачи и потребности).

Дякую за увагу! ☺