# Задание 1. Фотоколлажи

**Введение**

Задача автоматического построения панорам в последнее время успешно решается методами компьютерного зрения. Появляются специализированные сервисы и приложения, например [**photosynth**](http://photosynth.net/) или [**autostitch**](http://cs.bath.ac.uk/brown/autostitch/autostitch.html). В первых двух заданиях вам будет предложено самим реализовать алгоритм автоматической сшивки панорам и создания коллажей.

В первом задании предполагается, что изображения панорамы уже были выровнены друг относительно друга и нужно составить из этих изображений-лоскутков одну большую панораму. Задача решается в два этапа:

* Сначала в соответствии с каким-либо функционалом качества выбираются части изображения, которые попадут в итоговую панораму (сшивка)
* После составления панорамы происходит попытка выровнять разные её части, чтобы панорама визуально смотрелась целостной. Для этого используются методы блендинга, основанные на пирамидах изображений и смешивании изображений на разных частотах

**Обязательная часть**

Реализовать алгоритм сшивки и блендинга набора выровненных изображений одной панорамы.

**Описание алгоритма:**

* По набору изображений сформировать функционал, описанный в разделе "реализация". Узлами графа будут выступать пиксели панорамы. Нашей целью является присвоение метки каждому узлу графа таким образом, чтобы значение функционала было минимизировано
* После получения оптимальной разметки на финальном изображении нужно применить алгоритм блендинга, основанного на пирамидах лапласиан. Его лучше всего применять для каждого изображения-лоскутка панорамы в отдельности, смешивая его с его окружением на полученной панораме

**Реализация**

Алгоритм должен быть реализован в виде функции на Python со следующей сигнатурой *stitch\_images(in\_dir, mode)*, где *“in\_dir”* – путь к директории с выровненными изображениями в формате *".png"*, *“mode”* – базовый (0) или бонусный вариант (1), возвращать функция должна сшитую панораму. Все изображения имеют один и тот же размер. Точки панорамы, которые данное изображение не покрывает, обозначаются черным цветом (0,0,0). Изображения частично перекрываются. Пример входных данных в файле *pano.zip*.

Для минимизации энергии при сшивке панорам необходимо воспользоваться модифицированной реализацией MATLAB-обёртки (файл *gcmex\_modified.zip*) над алгоритмом разрезов графов Бойкова-Векслер. В ней для задания энергии нужно сделать следующее:

* Прочитать описание обёртки в файле *GCMex\_readme.txt*
* Задать унарные потенциалы с помощью переменной UNARY и задать связи между узлами с помощью переменной *PAIRWISE* (пример использования в файле *GCMex\_test.m*)
* В файле *GCMex.cpp* заполнить функцию *smooth\_cost*, вычисляющую парные потенциалы. На вход этой функции приходит структура, содержащая все склеиваемые изображения, развернутые в линейный массив. Также приходит четыре параметра – *SiteID s1*, *SiteID s2*, *LabelID l1*, *LabelID l2*, где *s1* и *s2* – индекс текущего пикселя на панораме, *l1* и *l2* – метки пикселей *s1* и *s2* соответственно, соответствующие изображениям-лоскуткам. Функция должна возвратить штраф для конфигурации *(s1, s2, l1, l2)*. Чтобы получить интенсивность пикселя *s1* в изображении *l1*, можно воспользоваться конструкцией *imgs->images[l1][s1]*
* Скомпилировать *GCMex* с помощью скрипта *GCMex\_compile.m*

Чтобы найти оптимальную разметку, соответствующую заданному функционалу энергии из MATLAB-кода нужно вызвать функцию *GCMex(CLASS, UNARY, PAIRWISE, LABELCOST, EXPANSION, N\_IMAGES, GRAY\_IMAGES)*, где описание *CLASS, UNARY, PAIRWISE, LABELCOST, EXPANSION* можно посмотреть в *GCMex\_readme.txt*, *N\_IMAGES* – количество изображений-лоскутков, *GRAY\_IMAGES* – cell-массив grayscale изображений-лоскутков, развернутых в стоки (обратите внимание, что нашем случае параметр *PAIRWISE* задаёт только связи между пикселями изображений, а параметр *LABELCOST* ни на что не влияет, так как используется более гибкая схема задания парных потенциалов).

Для сшивки панорам можно воспользоваться функционалом из статьи [2]:

где – значение энергии для разметки , – унарный потенциал, – парный потенциал, – граничащие пиксели, – метки граничащих пикселей. Для правильной сшивки панорам нужно задать унарный штраф таким образом, чтобы черные области не попадали в итоговую панораму, если на их месте можно поместить значимую часть изображения-лоскутка. Парные потенцилы задаются в функции *smooth\_cost* файла *GCMex.cpp* и имеют следующий вид:

где – интенсивность изображения-лоскутка номер в пикселе . То есть данный парный потенциал предпочитает проводить разрезы там, где по крайней мере два изображения согласованы по интенсивности, что и обеспечивает плавные переходы.

Блендинг изображений-лоскутков на итоговой панораме нужно написать самостоятельно.

**Советы по реализации**

При заполнении массива связанности PAIRWISE, который представлен разреженной матрицей, воспользуйтесь функцией *pairwise = sparse(rows, cols, vals)*, которая создаёт разреженную матрицу с ненулевыми элементами в строках *rows* и столбцах *cols*. Такой способ создания гораздо быстрее, чем заполнение разреженной матрицы поэлементно.

**Бонусная часть**

Добавить возможность задания с помощью мазков частей изображений, которые точно должны попадать в финальную панораму и применить эту технику к набору *family portrait* из [2], создав сшитое изображение семейного портрета. Мазки для каждого изображения должны задаваться с помощью двухцветных файлов, где белым цветом (255,255,255) задаётся мазок, а черным – всё остальное. Имена файлов с мазками должны иметь следующий вид – *'имя\_файла\_соответствующего\_изображения\_без\_расширения\_strokes.png'* и должны быть помещены в папку с изображениями-лоскутками.

**Содержание архива с выполненным заданием**

Архив должен содержать код и файл *readme*, со следующей информацией:

**Общая информация:**

* ФИО: *<Фамилия Имя Отчество>*
* Задание: *<название задания>*
* Система: *<система программирования>* ОС: *<операционная система>*
* Комментарии: *<комментарии по реализации / пожелания / впечатления и т.д.>*

**Базовая часть:**

* Склеенная панорама для набора *panoramic stitching* из [2]
* Сколько времени у вас заняло задание?
* С чем было сложнее всего разобраться?

**Бонусная часть:**

* Склеенная панорама для набора *family portrait* из [2] и файлы с мазками

**Литература**

1. [**Image Mosaic**](http://pages.cs.wisc.edu/~csverma/CS766_09/ImageMosaic/imagemosaic.html)
2. [**Interactive Digital Photomontage**](http://grail.cs.washington.edu/projects/photomontage/)
3. [**Burt P. J., Adelson E. H. A multiresolution spline with application to image mosaics //ACM Transactions on Graphics (TOG). – 1983. – Т. 2. – №. 4. – С. 217-236.**](http://www.cs.princeton.edu/courses/archive/fall05/cos429/papers/burt_adelson.pdf)

**Данные**

* Набор для сшивки панорамы – **pano.zip**
* Набор для создания семейного портрета – **family.zip**

**Библиотеки**

Модифицированный вариант *GCMex* – **gcmex.zip**