



Начало выполнения задания: 21 марта.

Срок сдачи: **5 апреля (воскресенье), 23:59.**

Среда для выполнения задания: MATLAB. Неэффективная реализация кода может негативно отразиться на оценке.

Содержание

1	Распределение по вариантам	1
2	Марковское случайное поле	2
3	MRF для склеивания изображений	2
4	Задание	2
5	Спецификация реализуемых функций	3
6	Рекомендации по выполнению задания	4
7	Данные для выполнения задания	4
8	Оформление задания	4

1 Распределение по вариантам

Задание состоит из двух вариантов. Распределение вариантов задания по студентам:

Вариант 1: α -расширение	Вариант 2: $\alpha\beta$ -замена
Апишев Мурат	Готман Мария
Афанасьев Кирилл	Колмаков Евгений
Дойков Никита	Корольков Михаил
Козлов Владимир	Лисяной Александр
Лукашкина Юлия	Сендерович Никита
Ожерельев Илья	Тюрин Александр
Родоманов Антон	Хальман Михаил
Славнов Константин	Чистяков Александр
Хомутов Никита	Шапулин Андрей

2 Марковское случайное поле

Марковское случайное поле (MRF) — графическая модель, энергия которой записывается в виде:

$$E(X) = \sum_{i \in \mathcal{V}} \theta_i(x_i) + \sum_{(i,j) \in \mathcal{E}} \theta_{ij}(x_i, x_j), \quad x_i \in \mathcal{P},$$

где \mathcal{V} — множество индексов переменных, \mathcal{E} — система соседства, $\theta_i : \mathcal{P} \rightarrow \mathbb{R}$ — унарные потенциалы, $\theta_{ij} : \mathcal{P} \times \mathcal{P} \rightarrow \mathbb{R}$ — парные потенциалы. Обратите внимание, что в сумме по рёбрам $(i, j) \in \mathcal{E}$ каждое ребро графа учитывается только один раз.

Для уменьшения числа параметров, парные потенциалы можно параметризовать следующим образом:

$$\theta_{ij}(x_i, x_j) = c_{ij} \cdot d(x_i, x_j),$$

где c_{ij} — коэффициенты, соответствующие рёбрам графа, а $d(x_i, x_j)$ — расстояние между метками соседних переменных. В предлагаемой ниже спецификации функций парные потенциалы задаются именно таким образом.

Рассмотрим модель со следующими ограничениями:

- переменные x_p дискретны и принимают значения из множества $\mathcal{P} = \{1, \dots, K\}$, $K \geq 2$,
- система соседства \mathcal{E} — прямоугольная решетка.

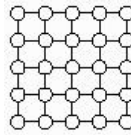


Рис. 1: Система соседства — прямоугольная решетка

В рамках данного задания каждый студент должен реализовать один алгоритм минимизации энергии: вариант 1 реализует алгоритм α -расширение, вариант 2 — $\alpha\beta$ -замена. Далее в задании алгоритм соответствующего варианта обозначается как *алгоритм*.

3 MRF для склеивания изображений

Задача склеивания изображений состоит в построении одного составного изображения на основе набора исходных изображений. В рамках данного задания предполагается, что все исходные изображения выровнены друг относительно друга. В этих условиях задачу можно решать при помощи минимизации энергии, переменные которой соответствуют номеру изображения, из которого взят конкретный пиксель.

Для задачи склеивания энергия строится следующим образом:

- Переменные x_p соответствуют пикселям финального изображения.
- Значение каждой переменной соответствует номеру изображения исходного набора, из которого взят цвет соответствующего пикселя.
- Используется стандартная 4-х связная система соседства.
- Унарные потенциалы должны показывать, из каких изображений должны быть взяты некоторые пиксели (так называемые семена).
- Парные потенциалы должны поощрять 1) короткие разрезы и 2) расположение разреза там, где изображения хорошо соответствуют друг другу.

В рамках данного задания предполагается, что студентами будет проведено исследование по подбору потенциалов, обеспечивающих визуально хорошее качество склеивания изображений.

4 Задание

1. Вывести все формулы, использующиеся в вашей реализации *алгоритма* (сведение шага алгоритма к разрезу графа).
2. Реализовать *алгоритм*, используя выданный код разрезов графов.

3. Протестировать *алгоритм* на модельных данных.
4. Реализовать процедуру решения задачи склеивания двух изображений. Построить не менее 1 *хорошей* композиции, состоящей из двух частей.
5. Реализовать процедуру склеивания произвольного числа изображений. Построить не менее 1 *хорошей* композиции, состоящей из не менее чем 4-х частей.
6. На задаче склеивания Вашего набора изображений сравнить работу алгоритмов α -расширение и $\alpha\beta$ -замена. Реализацию недостающего алгоритма можно взять у товарища, выполняющего другой вариант (в отчете обязательно указывать, чей код вы используете). Требуется провести сравнение по энергии получаемого решения, по времени работы, по визуальному качеству решения. Все выводы должны быть подтверждены числами, графиками, картинками.
7. Написать отчет в формате PDF с описанием всех проведенных исследований.

Обратите внимание, что для выполнения пунктов 4 и 5 каждый студент должен использовать уникальные (отличающиеся от изображений других студентов) изображения. Допускается как использование фотографий, сделанных собственноручно, так и использование картинок из интернета. Создание необычных (на усмотрение преподавателей) коллажей будет поощряться.

Композиция является *хорошей*, если границы на ней не более заметны, чем на композиции, приведённой в начале этого задания. Для достижения *хорошего* качества рекомендуется использовать редакторы изображений для выравнивания геометрии и цветов исходных изображений.

5 Спецификация реализуемых функций

Алгоритм

```
[labels, energy, time] = alphaExpansionGridPotts(unary, vertC, horC, metric)
[labels, energy, time] = alphaExpansionGridPotts(unary, vertC, horC, metric, options)
[labels, energy, time] = alphaBetaSwapGridPotts(unary, vertC, horC, metric)
[labels, energy, time] = alphaBetaSwapGridPotts(unary, vertC, horC, metric, options)
```

Вход:

- **unary** — унарные потенциалы, массив типа double размера $N \times M \times K$, где N — высота решетки, M — ширина решетки, K — количество меток;
- **vertC** — коэффициенты c_{pq} , соответствующие вертикальным ребрам, массив типа double размера $(N - 1) \times M$;
- **horC** — коэффициенты c_{pq} , соответствующие горизонтальным ребрам, массив типа double размера $N \times (M - 1)$;
- **metric** — расстояние между метками соседних переменных, массив типа double размера $K \times K$;
- **options** — (необязательный аргумент) набор дополнительных параметров, структура со следующими полями:
 - **maxIter** — максимально допустимое число итераций алгоритма (по умолчанию = 500);
 - **display** — параметр типа logical: если true, то при каждом запуске алгоритма разреза графа нужно выводить на экран номер итерации, номера обрабатываемых меток, текущее значение энергии;
 - **numStart** — количество запусков из разных начальных приближений;
 - **randOrder** — параметр типа logical: если true, то при каждом запуске использовать случайный порядок меток α и β ;

Выход:

- **labels** — разметка, обладающая наименьшей энергией, массив типа double размера $N \times M$;
- **energy** — значения энергии на каждой итерации, вектор типа double длины, равной количеству итераций алгоритма;
- **time** — время, пройденное с начала работы алгоритма до каждой итерации, вектор типа double длины, равной количеству итераций алгоритма.

Обратите внимание: в процедурах `alphaExpansionGridPotts` и `alphaBetaSwapGridPotts` параметры `N`, `M`, и `K` определяются неявно по размеру соответствующих элементов.

Склеивание

```
[resultImage, resultMask] = stichImages(images, seeds)
```

Вход:

- `images` — набор исходных изображений, cell array размера `K x 1`, где `K` — кол-во изображений. Все изображения должны быть одинакового разрешения и содержать ровно 3 цветовых канала;
- `seeds` — маски, заданные пользователем, cell array размера `K x 1`. Каждый элемент - логический массив размера, равного разрешению изображения, в котором значение `true` означает, что соответствующий пиксель должен быть взят из соответствующего изображения.

Выход:

- `resultImage` — построенное изображение;
- `resultMask` — маска ответа, массив типа `double`, по размеру равный разрешению изображения. Элемент `results(i,j)` равен `k`, если в построенном изображении пиксель `(i, j)` взят из исходного изображения номер `k`.

6 Рекомендации по выполнению задания

1. Обратите внимание на область применимости *алгоритма*.
2. При тестировании *алгоритма* необходимо следить за следующим:
 - после каждого применения разреза графа общая энергия не возрастает;
 - значение энергии, выдаваемое функцией `graphCutMex`, совпадает со значением энергии, подсчитанным независимой процедурой.
3. Обратите внимание, что для достижения хорошего качества решения задачи склеивания, возможно придется изменить интерфейс, выданный в задании. В том числе допускается отказ от параметризации парных потенциалов в виде $c_{ij}d(x_i, x_j)$. При использовании измененного прототипа **ОБЯЗАТЕЛЬНО** нужно прислать и функцию, согласованную с выданным прототипом.

7 Данные для выполнения задания

`graphCut` — MATLAB интерфейс к разрезам графов.

8 Оформление задания

Выполненный вариант задания необходимо прислать письмом по адресу bayesml@gmail.com с темой

«[ГМ14] Задание 2, вариант X, фамилия».

Убедительная просьба присылать выполненное задание **только один раз** с окончательным вариантом. Новые версии будут рассматриваться только в самом крайнем случае. Также убедительная просьба строго придерживаться заданной выше спецификации реализуемых функций. Очень трудно проверять большое количество заданий, если у каждого будет свой формат реализации.

Письмо должно содержать:

- PDF-файл с описанием проведенных исследований (отчет должен включать в себя описание выполнения каждого пункта задания с приведением соответствующих графиков, изображений, чисел).
- Все исходные файлы, реализованные в рамках настоящего задания. Убедитесь, что Ваши файлы работают и соответствуют прототипам!
- Не менее двух склеенных изображений и соответствующих им наборов исходных.