

Задание 2. Алгоритмы минимизации энергии для задачи склеивания панорам

Курс: Графические модели, весна 2016



Начало выполнения задания: 18 марта.
 Срок сдачи: **1 апреля (пятница), 23:59.**
 Среда для выполнения задания: Python.

Contents

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

1 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.

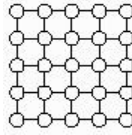


Figure 1: Система соседства — прямоугольная решетка

В рамках данного задания каждый студент должен реализовать алгоритм минимизации энергии α -расширение.

2 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.

3 Задание

1. Вывести все формулы α -расширения (сведение шага алгоритма к разрезу графа).
2. Реализовать α -расширение, используя выданный код разрезов графов.
3. Протестировать α -расширение 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. 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.

4 Спецификация реализуемых функций

Алгоритм

```
(labels, energy, time) = alphaExpansionGridPotts(unary, vertC, horC, metric, maxIter=500, display=False, numStart=1, randOrder=False)
```

Вход:

- `unary` — унарные потенциалы, `numpy.array` размера $N \times M \times K$, где N — высота решетки, M — ширина решетки, K — количество меток;
- `vertC` — коэффициенты c_{pq} , соответствующие вертикальным ребрам, `numpy.array` размера $(N - 1) \times M$;
- `horC` — коэффициенты c_{pq} , соответствующие горизонтальным ребрам, `numpy.array` размера $N \times (M - 1)$;
- `metric` — расстояние между метками соседних переменных, `numpy.array` размера $K \times K$;
- `maxIter` — максимально допустимое число итераций α -расширения;
- `display` — 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` — если `True`, то при каждом запуске использовать случайный порядок меток α ;

Выход:

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

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

Склеивание

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

Вход:

- `images` — набор исходных изображений, Python список длины K , где K — кол-во изображений. Все изображения являются `numpy.array`, должны быть одинакового разрешения и содержать ровно 3 цветовых канала;
- `seeds` — маски, заданные пользователем, Python список длины K . Каждый элемент - `numpy.array`, 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, `numpy.array`, по размеру равной разрешению изображения. Элемент `resultMask[i, j]` равен k , если в построенном изображении пиксель (i, j) взят из исходного изображения номер k .

5 Рекомендации по выполнению задания

1. Обратите внимание на область применимости алгоритма α -расширения.
2. При тестировании α -расширения необходимо следить за следующим:
 - после каждого применения разреза графа общая энергия не возрастает;
 - значение энергии, выдаваемое функцией `graphCutMex / graph_cut`, 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.

6 Данные для выполнения задания

[graphCut](#) — python интерфейс к разрезам графов.

7 Оформление задания

Выполненное задание необходимо прислать письмом по адресу bayesml@gmail.com с темой

«[ГМ16] Задание 2 Фамилия Имя».

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 все Ваши скрипты и функции работают и соответствуют прототипам!
- Не менее двух склеенных изображений и соответствующих им наборов исходных.